

Ўзбекистон Республикаси
Олий ва Ўрта махсус таълим вазирлиги

ТЕРМИЗ ДАВЛАТ УНИВЕРСИТЕТИ

ФИЗИКА-МАТЕМАТИКА ФАКУЛЬТЕТИ

АМАЛИЙ МАТЕМАТИКА ВА ИНФОРМАТИКА

КАФЕДРАСИ

DELPHI

дастурлаш тили

Термиз-2011 й

АННОТАЦИЯ

Сўнги йилларда дасурлашга бўлган қизиқиш тобора ортиб бормоқда. Бу компьютер технологиясининг кун сайин ривожланиб бориши билан боғлиқдир. Айниқса визуал дастурлаш технологияларидан фойдаланиб дастурлар яратиш компьютер технологиясининг ривожланишига катта таъсир этмоқда.

Ушбу ўқув қўлланмада талабаларга Delphi дастурий воситасида визуал дастурлаш технологиялари ҳақида тўлиқ тасаввурга эга бўлиши учун керакли маълумотлар берилган.

Мазкур услубий қўлланма олий ўқув юртларининг талабалари учун “Дастурлаш асослари”, “Delphi дастурлаш тили”, “Объектга йўналтирилган дастурлаш” фанларидан амалий ва тажриба машғулоти ўтказишга мўлжалланган.

Услубий қўлланма Delphi да программалашни ўрганиш учун зарур бўлган асосий тушунча ва кўрсатмаларни ўз ичига олган.

Тузувчилар:

к.ўқ. Л.Ю. Тураева,
ўқ. Ш.А.Менглиев,
ўқ. Б.Хуррамов

Тақризчи:

ф.м.ф.д.доц Ч.Б.Нармурадов

Кириш

Ҳозирги вақтга келиб компьютер оламида кўплаб дастурлаш тиллари мавжуд. Улар Бейсик, Паскал, Си, C++, Delphi ва бошқа дастурлаш тилларидир. Паскал тили 1969 йил Н. Вирг томонидан яратилган бўлиб, кейинчалик американинг Borland фирмаси томонидан қайта ишланиб Turbo Pascal деб номланди. Turbo Pascal ни қайта ишлаш натижасида объектли дастурлаш йўлга қўйилди ва у Object Pascal деб атала бошланди. Ҳисоблаш техникаси ва технологиясининг ривожланиши натижасида Borland фирмаси томонидан янги Delphi дастурлаш тили яратилди.

Delphi дастурлаш тили Windows учун мўлжалланган бўлиб, унинг биринчи версияси Windows 3.1 операцион система қобиғида ишлаган. Windows 95 операцион система яратилганидан сўнг, 16-разрядли Delphi 2, кейинроқ 32-разрядли Delphi 3 версияси яратилди. Windows 98 операцион системаси учун Delphi нинг тўртинчи версияси ва ҳозирги кунда Delphi 5, Delphi 6, Delphi 7 ва Delphi 8 лар пайдо бўлди.

Delphi дастурлаш тили – бу дастурларни қайта ишлаш мухити бўлиб, Windows операцион системасида ишлайди. Унда объектли дастурлаш тили бўлган Object Pascal мужассамлашган.

Delphi визуал проектлар, турли ҳолат процедураларини қайта ишлаш ва дастурларни қайта ишлашда вақтдан ютиш ва бошқаларни ўз ичига олади.

DELPHI дастурий востаси-бу Windows учун мўлжалланган дастурлаш мухити бўлиб, 1995 йилда BORLAND компанияси гуруҳи дастур тузувчилари Чак (Chuck) ва Денни (Danny) томонидан яратилган.

Бу тил ўзининг кенг қамровли имкониятларига эгаллиги билан бирга, бошқа дастурлаш тилларидан ўзининг баъзи бир хусусиятлари билан ажралиб туради.

Borland Delphi нинг пайдо бўлиши дастурлашни ривожлантириш тарихида ёрқин кўриниш бўлди. Delphi нинг дунёга келишига қуйидагилар сабаб бўлди;

- Windows учун дастулаш ва компоненталар технологияси.
- Масалаларни ечиш учун объектга йўналтирилган усул.
- Компоненталар технологиясига асосланган иловаларни тез яратишнинг визуал мухитлари.
- Интерпритациядан эмас, компиляциядан фойдаланиш. Бу шундан иборатки, интерпретатор билан ишлашга қараганда компилятор билан ишлаш тезлиги ўн мартага устунликка эга бўлади.
- Универсал усуллар ёрдамида маълумотлар базаси билан ишлаш имкониятларининг мавжудлиги.

Borland Delphi юкорида баён этилган тенденцияларни жорий этиш мақсадида яратилган. Аммо, унинг энг асосий элементи Паскаль тили бўлиб ҳисобланади. Ҳозирги кунда Delphi тилининг бир неча хил вариантлари мавжуд (Delphi 2, Delphi 3... , Delphi 10 ва ҳ.к.).

Delphi дастурлаш тили учун Паскаль дастурлаш тили асос қилиб олинган. Бу икки тилнинг қандай ўхшашлик ва фарқли томонлари бор?

Биламизки, Паскалда бирор катгароқ дастур, масалан амалий дастурлар мажмуини тузмокчи бўлсак, албатта биз модулли дастурлашдан фойдаланамиз. Яъни қўйилган масалани кичикроқ бўлакларга бўлиб оламиз ва шулар билан ишлаймиз. Натижани олиш учун эса, бу бўлакларни бирлаштирувчи битта асосий дастур қилинади. Айнан шу нарса Delphi дастурлаш тилининг асоси ҳисобланади. Агар биз Delphi да бирор форма ҳосил қилсак, у ҳолда Delphi бизга ўзи автоматик тарзда шу формага мос бўлган модулларни ва бу модулларни бошқарувчи дастурни тузиб беради. Бошқарувчи дастурни Delphi да, агар унга фойдаланувчи ном бермаса, Project1.dpr деб номлайди. Бунда .dpr файл кенгайтмаси. Автоматик тарзда ташкил қилинган модулга фойдаланувчи ном бермаса, у ҳолда Delphi уни Unit1.pas деб номлайди. Фойдаланувчи эса .pas кенгайтмасини сақлаган ҳолда хоҳлаган номини бериши мумкин.

Delphi ni ishga tuшириш

Delphi ni ikki usulda ishga tuшириш mumkin:

1. Пуск (Start) кнопкаси босилади, Программы сатри танланади ва Borland Delphi 4 сатридан Delphi 4 ойнасига кириб устига сичқончани чап тугмасини босиш билан ;
2. Ишчи столга ўрнатилган ёрликни устига сичқонча кўрсаткичини ўрнатиб, чап тугмасини икки марта босиш билан (Ёрликни фойдалувчини ўзи яратиб олиши керак).

Delphi da бошлангич амаллар ва проектлар

Delphi ni ishga tuширгандан кейин уни экран кўриниши ҳосил бўлиб, у унчалик оддий эмас. Экранда бешта ойна ҳосил бўлади:

- бош ойна-Delphi Project 1;
- форма ойнаси -Form1,
- объект хоссаларини тахрирлаш ойнаси- Object Inspector;
- объектлар руйхатини кўриш ойнаси – Object tree View;
- дастур кодларини тахрирлаш ойнаси– Unit1.pas

Бош ойна экраннинг юқори қисмида жойлашган былиб, унинг биринчи қаторида сарлавха, яъни проектнинг номи жойлашган. Иккинчи қаторда буйруқлар менюси горизонтал кўринишда жойлашган. Кейинги қаторнинг чап тарафида ускуналар панели ва ўнг тарафида компоненталар палитраси жойлашган.

Object Inspector ойнаси ёрдамида объектлар хусусиятларини ўзгартириш mumkin: формалар, буйруқ тугмалари, киритиш майдонлари ва ҳоказоларни. Масалан, Width ва Height хоссалари форма ўлчамини, Top ва Left эса форманинг экрандаги холати, Caption-сарлавха матнини аниқлайди. Delphi da дастурлаш иккита ўзаро таъсир этувчи бир бири билан боғлиқ жараён асосида ташкил қилинади:

-дастурни визуал лойихалаш жараёни;

-дастур кодларини киритиш жараёни.

Delphi da дастурлаш форма ойнасини ташкил этишдан бошланади. Оддий дастур иловасини яратиш кетма-кет File-New-Application буйруғини бериш билан бошланади.

Тузилган дастурни ЭХМ хотирасида сақлаш учун асосий менюнинг File бандидан фойдаланилади. Дастурни компиляция қилиш учун асосий менюнинг RUN бандидан фойдаланилади. Агар дастурда хатоликлар бўлса, компиляция қилиш жараёнида бу хатоликлар кўрсатиб борилади. Агар хатолик бўлмаса, дастур ишга тушади.

Delphi дастурлаш тилини ишга туширилганда экранда қуйидаги мухит ҳосил бўлади:

можете увидеть на экране картинку, подобную представленной на рис. 1.1.

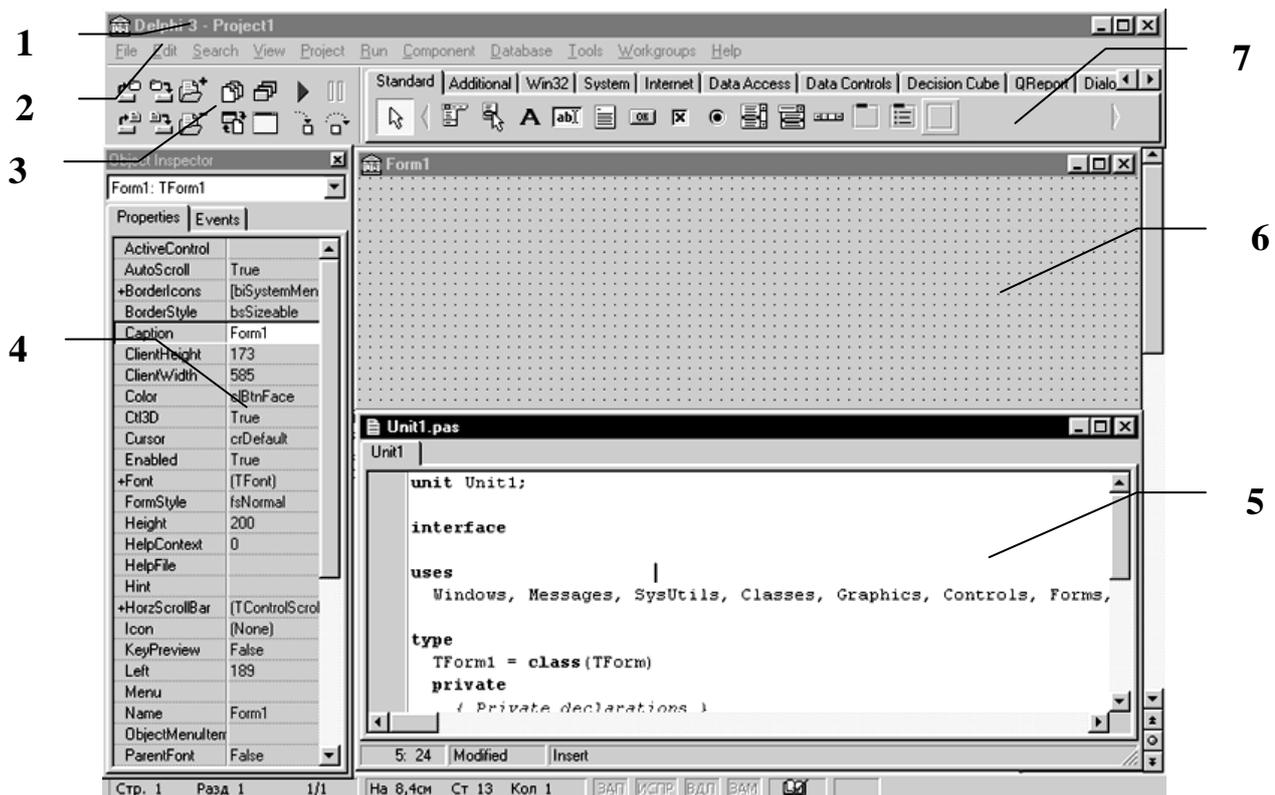


Рис.1.1.

1- главное окно; 2 – основное меню; 3 – пиктограммы основного меню;

4 - окно инспектора объектов; 5 – окно текста программы;

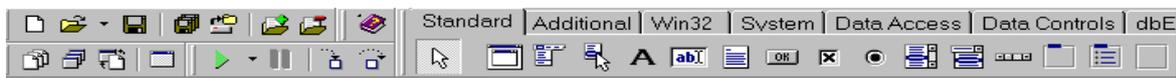
Главное окно всегда присутствует на экране и предназначено

Меню сатрининг кўриниши



Delphi асосий меню бандлари вазифалари Microsoft office стандарт дастурларининг меню бандлари вазифаларидан деярли фарқланмайди ва улардан фойдаланиш тартиби ҳам худди шундай сакланади.

Асбоблар панели ва компоненталар рўйхати



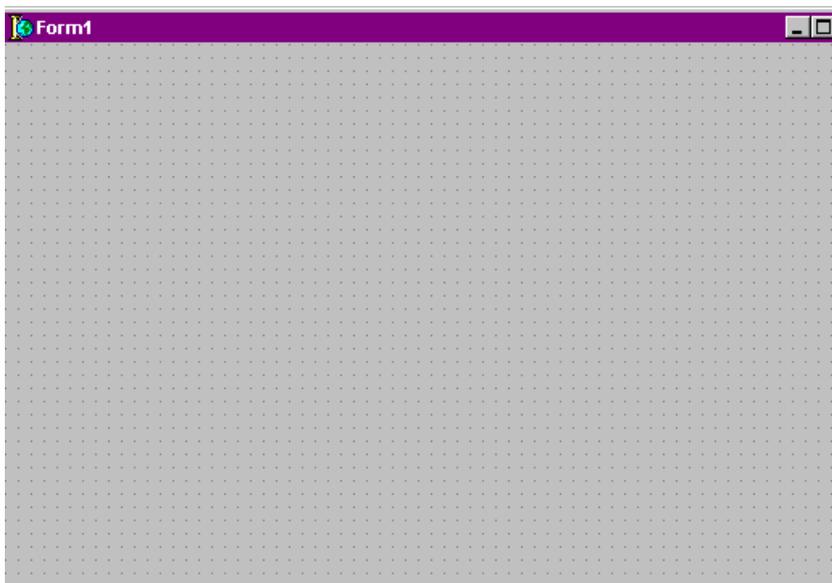
Объектлар дарахти



Объектлар инспектори



Дастур формаси



Объектлар инспекторидан фойдаланиш

Windows учун тузилган программанинг ички тузилиши консоль программаларнинг ички тузилишидан фаркланади. Дастур бажарилишида калит сўзлар бўлган `begin` ва `end` орасига олинган операторларни `windows` бошқача шаклда бошқаради, яни уларни ҳодиса деб қарайди. Ҳар бир ҳодисанинг ўзига мос хоссалари мавжуд. Дастурлашда бу хоссалардан керагини танлаб ишлатилади. Танлаш объектлар инспекторида бажарилади. Объектлар инспектори икки қисмдан иборат `Properties` яни ходисалар ва `Events` хоссалар. Delphi нинг характерли томони шундан иборатки, агар биз бирор компонентадан фойдалансак, объектлар инспектори унга мос ходисаларни ажратиб кўрсатиб

туради. Масалан биз бирор янги форма ҳосил қилган бўлсак, объектлар инспекторида сартionда Form1 ёзуви туради. Агар биз хоҳласак Form1нинг ўрнига янги ўзимизнинг программага мос номни беришимиз мумкин. Ҳар бир компонентга объектлар инспекторида унга тегишли ходиса ва шу ходисага мос хоссалар бўлади.

Delphi дастурлаш тилининг яна бир характерли томони шундан иборатки, агар биз дастурда бирор тугмага қандайдир вазифани юкламоқчи бўлсак, шу тугманинг устида сичқончани икки марта чертилса, шу тугмага мос келувчи процедуранинг ичига автоматик равишда киритиб қўяди. Бундан ташқари дастур тузиш жараёнида формадан модулнинг ичига ва модулдан формага ўтишга эҳтиёж бўлади, бунинг учун асбоблар панелидан Toggle Form-Unit тугмаси танланади ёки F12 тугмасини босиш билан амалга оширилади.

Формалар билан ишлаш

Delphiда тузиладиган дастурлар бирор форма асосида тузилади. Delphi да ҳар бир янги ҳосил қилинган формага унга мос бўлган модул автоматик ташкил қилиниб турилади. Бу эса дастурчи учун жуда қулай имконият, яни унинг ишини тезлаштиради.

Бу форманинг шаклини танлаш, унда компоненталарни жойлаштириш фойдаланувчи ихтиёрида бўлади. Формага бирор компонентани қўймоқчи бўлсак, шу компонентанинг устида сичқончанинг чап тугмаси икки марта босилади. Бу компонента форманинг ўртасига жойлашади. Биз уни хоҳлаган жойга суриб кўчиришимиз мумкин.

DELPHI да кодлар муҳити

Delphi кодлар муҳити автоматик тарзда Object Pascal дастурлаш тилидаги калит сўзларни (begin, end, procedure, const, var) қалин ҳарфлар билан белгилайди.

Маълумот ёзилган сатр (дастур изохи) ни белгилаш учун фигурали кавслардан фойдаланилади. Кавс очилса ундан кейин турган кодлар кўриниши ўзгаради. Керакли жойда кавс беркитилса кўриниши ўзгарган кодлар фақат кавс оралиғидагина қолади ва дастур ишлаш жараёнида шу оралик ишлатилмайди.

Delphi кодлар муҳитининг имкониятларидан яна бири шуки, у ерга бирор функцияни масалан: «StrToFloat» ни ёзиб, кавс очсак сатр остида кичик ойна ҳосил бўлади. Бу ойнада кавс ичидаги ўзгарувчи типи кўрсатилган бўлади, ёки бирор операторни масалан: Label1 ни ёзиб нуқта қўйилса сатр остида нуқтадан кейинги ёзиш мумкин бўлган операторлар рўйхати чиқади ва улاردан кераклигини танлаб қўйишимиз мумкин.

Кодлар ойнасида бирор оператор устига курсорни олиб бориб Ctrl+F1 тугмалари тенг босилса шу оператор ҳақидаги ёрдам ойнаси ҳосил бўлади. У ердан керакли ахборотни олиш мумкин. Агар курсорни бўш жойга олиб келиб F1 босилса умумий ёрдам файллари чиқади.

Кодлар ойнасида тахрирлаш оддий матн муҳаррирлари каби амалга оширилади. Яъни белгиланган (блокка олинган) код нусхасини олиш, қирқиб олиш ва керакли жойга қўйиш мумкин. Ундан ташқари кодлар ичидан керакли белгини излаб топиш ва алмаштириш, Delete тугмаси ёрдамида курсордан кейин турган белгини, Backspace ёрдамида эса курсордан олдин турган белги ёки белгиларни ўчириш мумкин. Ctrl+→, Ctrl+← клавишлари ёрдамида бир сўз кейинга ва олдинга, PgDn, PgUp клавишлари ёрдамида эса бир экран пастга ва юқорига ўтилади.

Проект

Delphi проекти – бу компилятор томонидан, дастур яратгандан сўнг, яратилган дастурга тегишли бўлган файллар тўпламидир. Проект, бир ёки бир нечта проект файлларини ва модулларни ўз ичига олади (Unit модули).

Проект файли *.dpr кенгайтмасига эга бўлиб, проектнинг умумий ҳолатини ўзида сақлайди. Проект модули файли эса *.pas кенгайтмали бўлиб, ишчи файлини яратишда компиляторга керак бўлувчи процедура, функция матнлари, типларни тавсифи ва бошқа маълумотларни ўзида сақлайди.

Модуль имеет следующую структуру:

```
unit Unit1;  
interface
```

```

// Раздел объявлений
implementation
// Раздел реализации
begin
// Раздел инициализации
end.

```

Delphi мухити.

Delphi дастурлаш тилини ишга туширганимизда унинг ишчи экран кўринишини кўрамиз. Дастур мухитидан буйруқлар менюси, буйруқ тугмачалари, компонентлар палитраси, объект инспектор, форма ва уни ортида тахрирлагич ойнаклари жой олган бўлиб деярли экранни тўлдириб туради.

Буйруқ менюси.

Delphi нинг меню сатридан қуйидагилар жой олган.

File, Edit, Search, View, Project, Run, Component, Database, Tools, Help.

Буларнинг барчасида ост менюлар мавжуддир. File нинг ост менюсида бир неча буйруқлар бўлиб улар ёрдамида янги проект очиш, янги форма очиш ва уларни сақлаш мумкин шу билан биргаликда очилган проектни ёпиш, Delphi дан чиқиш ва шуларга ўхшаш файллар билан ишлаш имкониятлари бор.

Edit менюси ост менюларидан фойдаланиб кодларни тахрирлаш, умуман кодлар устида турли хил амалларни бажариш мумкин.

View ёрдамида эса Delphi ишчи мухити кўринишининг ўзгариши мумкин.

Run менюси ёрдамида дастурни ишга туширишни турли йўллари амалга оширилади.

Database менюсида маълумот баъзасини ташкил қилиш мумкин.

Help менюси эса Delphi ва унда дастурлаш ҳақидаги барча маълумотларни олиш имкониятини яратади.

Буйруқлар тугмачаси

Буйруқлар тугмачаси ёрдамида янги формалар яратиш, мавжуд файлни очиш, Дастурни сақлаш, янги форма яратиш ва шунга ўхшаш амаллар тез бажарилади.

Компонентлар палитраси

Бу ерда стандарт ёки дастурчилар томонидан яратилган компонентлар мавжуд бўлиб, улардан тез ва сифатли дастурлар яратишда фойдаланилади.

Object Inspector ойнаси

Object Inspector ойнаси қуйидаги объектларнинг ҳолатини ўзгартиради: формалар, буйруқлар тугмачаси, кодлар майдони ва бошқалар.

Дастур формаси

Дастур тузишда ишлатиладиган барча компонентлар дастур формасига жойланади ва ана шу ердан уларга ўзгартириш киритилиши мумкин. Дастур ишга туширилгандан сўнг, барча амаллар дастур формаси ёрдамида бажарилади.

Дастур коди

Дастур коди форма орқасига яширинган бўлиб, у ерга дастур матнлари киритилади. У ойнага F12 ёки Ctrl+F12 тугмалари ёрдамида ўтиш мумкин. Delphi дастурлаш мухитида ишлаш жараёнида қуйидаги кенгайтмали файллар ишлатилади:

-лойиха файли, кенгайтмаси .dpr;

-паскал модули файли, кенгайтмаси .Pas;

-компонентлар жойлашган файл, кенгайтмаси .dcu;

-формалар жойлашган файл, кенгайтмаси .dfm;

-маълумотлар базаси файли, кенгайтмаси.dbf .

Компиляция этапида тайёрланган дастур матни Object Pascal тилига ўтказилади. Компановка босқичида эса керакли қўшимча ёрдамчи дастурлар ва остдастурлар унга бирлаштирилади. F9

тугмасини босиш билан Save Unit As диалог ойнаси пайдо бўлади ва у фойдаланувчидан Unit.pas модули учун файл номини ва у жойлашадиган папка номини кўрсатишингизни сўрайди. Агар жой кўрсатилмаса Delphi тизими автоматик равишда дастурни Bin папкасига жойлаштиради. Дастур компиляция қилиниш пайтида Delphi системаси .Pas; .dfm; .dcu кенгайтмали модуллар тузади. .Pas кенгайтмали файл кодларни ёзиш ойнасига киритилган дастур матнини, .dfm форма ойнаси ташкил этувчиларини, .dcu кенгайтмали файл эса .Pas ва .dfm кенгайтмали файлларнинг биргаликдаги машина кодига ўтказилган вариантини сақлайди. .dcu кенгайтмали файл компилятор томонидан ташкил қилинади ва ягона ишчи (бажарилувчи) .exe кенгайтмали файл ташкил қилишга база яратади.

Delphi компоненталари.

Биз куйида Delphi дастурлаш тилининг компоненталар библиотеказида мавжуд бўлган ҳамда кўп фойдаланиладиган компоненталар билан танишиб чиқамиз. Ушбу компоненталар Delphi тилини ўрганиш вақтида гурухларга ажратиб ўргатилган. Маълум бир компоненталар тўпламини компоненталар палитраси (палитра компонентов) деб номлаймиз.

Standard бўлими



MainMenu – асосий менюни ҳосил қилиш.



PopupMenu – асосий менюга аналог ҳолдаги меню.



Label – Текстни экранга чиқариш.



Edit – универсал киритиш элементи.



Memo – Матнларни купрок киритиш мумкин. 32 кб (10-20 варок) маълумотни узида сақлайди.



Button – кнопка урнатиш.



CheckBox – танлаш.



RadioButton – фақат биттасини танлаш.



ListBox – бир неча катор сузларни курсатиш. Улардан бирортасини танлаш мумкин.



ComboBox – Бунда ListBox дан фаркли уларок, яширинган ҳолатдаги сузлардан бирортасини танлаш мумкин.



ScrollBar – прокрутка (югурдак).



GroupBox – Гурухлаб олиш учун махсус жой.



RadioGroup – GroupBox га аналог холда RadioButton объектларини гурухлайди.



Panel – Бошқарувчи элемент. Янгисини урнатиб унинг устига яна янги элементларни куйиш мумкин.

Additional бўлими



BitBtn – Кнопка ўрнатиш. Button га ўхшаш.



SpeedBotton – Сичконча ёрдамида босиладиган кичкина кнопкача.



MaskEdit – Ниқобли киритиш элементи. Edit га аналог. Хусусияти аниқланади (масалан сана, валюта ва хоказо).



StringGrid – Текстли маълумотларни жадвал кўринишида чиқариш.



DrawGrid – Ихтиёрий турдаги маълумотларни жадвал кўринишида чиқариш.



Image - *.jpg, *.jpeg, *.bmp, *.ico, *.emf, *.wmf типли график тасвирларни чиқариш.



Shape – оддий график объектларни чиқариш (айлана, квадрат ва х.к.).



Bevel – Хошияларни безовчи элемент.



ScrollBar – янги прокруткали (югурдакли) майдон ҳосил қилиш.



CheckListBox – бир нечасини танлаш имконияти. CheckBox нинг бир нечтаси битта варақда бўлади.



Splitter – Формаларни бўлувчи ёки ажратувчи элемент.



StaticText – маълумотни махсус формада чиқариш.



ControlBar - toolbar компонентларини бошқариш.



Chart – махсус диаграмма графикларини ўрнатиш.



TabControl – қисмларга бўлинган ёзув дафттарини яратиш.



PageControl – кўп варақли диалог панели.



ImageList – тасвирларни сақловчи махсус жой.



RichEdit – матнларни киритувчи ва тахрирлочи ойна. Методдан имконияти кенг.



TrackBar – махсус бошқариш элементи.



ProgressBar – жараёни кўзатиш элементи.



UpDown – юқорига ва қуйига бошқариш элементи.

-  HotKey – актив (босилган) тугмачани номларини кўрсатувчи элемент.
-  Animate – анимация (харакат) қилувчи объектлар.
-  DateTimePicker – Йил, ой ва кун саналари билан ишлаш.
-  MonthCalendar – кунлар билан ишловчи алоҳида ой календари.
-  TreeView – маълумотларни иерархик, яъни дарахт кўринишида кўрсатувчи ойна.
-  TListView - маълумотларни тўлиқ қаторда кўрсатувчи ойна.
-  HeaderControl – объектлар учун бошқарувчи контейнер
-  StatusBar - ҳолат сатри (қуйи ҳошия) ни ўрнатиш.
-  ToolBar – Тугмачаларни бошқариш панели (қуроллар панели). У орқали тугмачаларларни ўлчами ва кўринишини ўзгартириш мумкин.
-  CoolBar - CoolBand объектовлари учун контейнер.
-  PageScroller – кичкина бошқариш панели. Toolbar га ўхшаш.

System бўлими

-  Timer – вақт билан ишлаш.
-  PaintBox – тасвирларни чизиш соҳаси.
-  MediaPlayer – овоз ва видео файлларини бошқариш.
-  OleContainer - OLE объектларини қўйиш ва боғлаш.

Data Access бўлими

-  DataSource – маълумотларни ёки компоненталарни бир-бири билан боғлаш.
-  Table – маълумотлар омборини (файлни) боғлаш.
-  Query – маълумотлар омборидаги ёзувларни бошқариш.
-  StoredProc – сервердан маълумотлар омборини юклаш.
-  Database – ягона маълумотлар омборини боғлаш.

Data Controls бўлими



DBGrid – маълумотлар омборидаги ёзувларни жадвал кўринишида чиқариш.



DBNavigator – маълумотлар омборидаги ёзувларни таҳрирловчи компонент. Унда янги сўз кўшиш, ўзгартириш, ўчириш ва х.к. ишларни қилиш мумкин.



DBText – маълумотлар омборидаги матнли майдон маълумотларини чиқариш.



DBEdit – маълумотлар омборидаги бирор майдонни таҳрирлаш.



DBMemo – маълумотлар омборидаги мемо типдаги маълумотларни таҳрирлаш.



DBImage – маълумотлар омборида жойланган тасвирларни кўрсатиш.



DBListBox – маълумотлар омборидаги маълумотларни листга чиқариш.



DBComboBox маълумотларни комбинацияли танлаш.



DBRichEdit – маълумотларни таҳрирлашнинг мемо га нисбатан кенгрок имконияти.

Dialogs бўлими



OpenDialog – файлларни очиш.



SaveDialog – файлларни сақлаш.



OpenPictureDialog – расмли файлларни очиш .



SavePictureDialog - расмли файлларни сақлаш.



FontDialog – шрифтларни танлаш.



ColorDialog – рангларни танлаш.



PrintDialog – принтерга чоп этиш.



PrinterSetupDialog – принтер хусусиятларини ўзгартириш.



FindDialog – файлдан маълумотларни излаш.

Win 3.1 бўлими



FileListBox – ажратилган каталогдаги файллар рўйхатини чиқариш.



DirectoryListBox – актив дискдаги каталоглар рўйхати.



DriveComboBox – дискларни активлаштириш ва танлаш элементи.



FilterComboBox – файлларни тури(кенгайтмаси) бўйича танлаш.

Samples бўлими



ColorGrid – ранг турини танлаш ёки активлаштириш .



SpinButton – юқорига ёки пастага бошқариш.



SpinEdit – сонларни автоматик ўзгартириш.



DirectoryOutline – каталоглар рўйхати.



Calendar – календар.

ActiveX бўлими



F1Book – формулалар ёзиш учун жадвал. Excel га ўхшаш.



VtChart – диаграмма.



Graph – график диаграмма ва чизмалар.

Label, Edit, Memo матн компонентлари ва Button тугмачаси.

Label белгиси. Бу белги тушунтиришлар, номлар, мавзулар ва бошқа матнли маълумотларни экранга жойлаштириш учун ишлатилади. Белги учун Caption асосий хоссалардан бири бўлиб, унда экранга чиқариладиган матн жойлашади. Матнни экранга жойлаш учун Delphi нинг Standart палитрасидан «A» пиктограммаси белгиланиб форма устига келинади ва сичқонча тугмасини босган ҳолда матнга жойлаштирилиши лозим бўлган жой ажратилади. Натижада label1 матн майдони хосил қилинади ва Caption хоссасига кирилиб матн терилади. Матнга ишлов бериш учун (масалан, катталаштириш ёки кичиклаштириш; курсив ёки қалин қилиш ва бошқалар) керакли хосса танланиб улар ўзгартирилади. Кирилган матнни катталаштириш ёки кичиклаштириш учун олдин матн майдони ажратилиб, кейин Font хоссасига кирилиди ва мулоқот дарчасидан shrift, унинг ўлчами ва ранги танланиб ok тугмаси босилади.

Label компонентаси нафақат маълумотларни экранга жойлаштириш учун хизмат қилади, балки дастур натижаларини чиқаришда ҳам ишлатиш мумкин. Бунинг учун дастурда Label 5.caption:='дастур натижаси'; буйруғи берилиши керак.
Мисол: Label 5.caption:='ечим'+s; бу ерда s:string ўзгарувчиси.

Edit киритиш қатори.

Edit киритиш қатори матнни бир қатордан киритиш ва уни тахрирлаш учун ишлатилади. Матн киритиш қаторини экранга жойлаш учун Delphi нинг Standart палитрасидан «ab» пиктограммаси белгиланиб, форма устига келинади ва сичқонча тугмасини босган ҳолда матн киритилиши лозим бўлган жой ажратилади. Натижада Edit 1 матн киритиш майдони ҳосил қилинади. Матнни киритиш дастур ишчи ҳолатга ўтганда бажарилади. Матн қаторига киритилган маълумот фақат матн, яъни String (қатор) бўлиб ҳисобланади. Edit киритиш қаторига киритилган маълумот дастурда ўқиб ва уни рақамга ўтказиш учун кўп ҳолларда val функциясидан фойдаланилади. Бу функция Turbo Pascal да қуйидагича ёзилади.

val(edit 1.text,a,cod)- бу ерда a:real;- ўзгарувчиси бўлиб, edit 1.text майдонидаги маълумотни рақам қилиб ўзлаштиради. cod: integer; деб эълон қилинади.

Мемо матн чиқариш қатори.

Мемо матнларни бир неча қатор килиб чиқариш учун ишлатилади. Мемо матн чиқариш қаторини экранга жойлаш учун Delphi нинг Standart палитрасидан «ab» пиктограммаси ёнидаги Мемо тугмаси белгиланиб форма устига келинади ва сичқонча тугмасини босган ҳолда матн чиқарилиши лозим бўлган жой ажратилади. Натижада Мемо 1 матн чиқариш майдони ҳосил қилинади. Бу чиқариш майдони дастурда натижаларни чиқаришда қўл келади. Натижани чиқаришда у дастур ичида қуйидагича ишлатилади.

```
Memo 1.lines.add('ечим=' + s);
```

Мемо майдонини тозалаш эса натижани чиқаришдан олдин модулда Memo 1.clear; буйруғини бериш билан амалга оширилади.



Мемо компонентаси содда матн муҳарири бўлиб, у кўп сатрли маълумотларни киритиш ва чиқариш учун ишлатилади.

Бу компоненталар мантлар устида амаллар бажариш учун ишлатилади.

Password Char хоссаси. Ушбу хосса матнни киритишдан химоя учун фойдаланилади. Парол киритиш керак пайтида фойдаланилади. Ушбу хоссадан фойдаланувчи махфийликнинг қўшимча даражасини таъминлаш учун ва фойдаланувчи исмини яшириш учун фойдаланиш мумкин.

Read Only хоссаси. Ушбу хосса объектнинг қийматини тахрирлаш имкониятини бошқаради. Фойдаланувчи унинг қийматини **False** ёки **True** қилиб қўйиши мумкин. Агарда хоссанинг қиймати **True** бўлса фойдаланувчи ушбу объектдан фойдаланишга, яъни ихтиёрий матнни белгилаш ва нусхасини буферга олиш учун бирмунча рухсат беради.

Max Length хоссаси. Ушбу хоссадан учала объектга киритиладиган символларни чеклаш учун фойдаланилади.

Mask Edit компонентасининг умумий хоссаларидан ташқари қўшимча хоссаларга ҳам эгадир. Ушбу хоссадан бу объект орқали клавиатурадан киритиладиган символларни филтерлаш учун фойдаланилади. Маска қандай символ қабул қилиниши ёки қилинмаслигини аниқлайди.



-List Box компонентаси.

Ушбу компонента фойдаланувчига танлаш учун элементлар рўйхатини кўрсатади. Айрим пайти фойдаланувчи ўз дастурида жуда кўп элементларнинг рўйхатини кўрсатиш лозим бўлиб қолади. Бундай ҳолатлар учун рўйхатлар дарчасида стандарт кўринишдаги сургич пайдо бўлади ва ёрдам беради. Рўйхатлар панелига элементларни қўшиш ёки олиб ташлаш мумкин. Ушбу компонентанинг ички кўринишини ўзгартириш учун мўлжалланган турли хил хоссалари мавжуд.

Рўйхатлар дарчасини формада жойлаш учун кўп жой зарур бўлади. Комбинациялашган панел компонентаси **Combo Box** ни формада жойни тежаш учун фойдаланиш қулайроқдир.

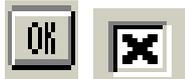


-Combo Box компонентаси.

Combo Box компонентаси **Combination Boxes** сўзидан қисқартирилган бўлиб **List Box** компонентасига ўхшашдир. Лекин комбинациялашган дарча кам жойни талаб қилади ва бир вақтнинг ўзида битта элементни кўрсатади.

Демак **List Box** компонентаси билан **Combo Box** компонентасининг фарқи бир вақтнинг ўзида қанча элемент кўрсата олишда экан. Жойдан тежаш мақсадида **Combo Box** дан фойдаланиш қулайроқ бўлса, кўпроқ элементни кўрсатиш учун **List Box** дан фойдаланиш қулайдир.

Button ва Check Box компоненталари.



Button (тугма) ва **Check Box** компоненталарининг кўпгина Windows муҳитида учратиш мумкин.

Button (тугма) компонентаси.

Button тугмачаси босилиши натижасида киритилиши лозим бўлган жараёнлар бажарилишга туширилади. Button тугмачасини экранга жойлаш учун Delphi нинг Standart палитрасидан «ok» пиктограммаси белгиланиб форма устига келинади ва сичқонча тугмачасини босган ҳолда тугмача қўйилиши лозим бўлган жой ажратилади.

Натижада Button 1 тугмачаси ҳосил қилинади. Тугмача номини ўзгартириш Caption хоссасига кирилиб ўзгартирилади. Дастурдаги ҳисоблаш жараёнлари, киритиш ва чиқариш операциялари ҳосил қилинган тугмачани икки марта тез тез босиш билан “события” ни қайта ишлаш дарчасига ўтилиб, у ердан модул ичига керакли операторларни ёзиш билан амалга оширилади.

Бу компонента одатда диалог дарчаларида фойдаланилади. Кўпгина Windows дастурларида таниш былган **Ok** ва **Cancel** жуфтлигини қыргансиз. Аммо сизга тугма компонентасидан фойдаланиш учун диалог дарчасини яратиш шарт эмас. Тугма компонентасини сиз дастурингизнинг исталган жойида жойлаштиришингиз мумкин.



-BitBtn компонентаси.

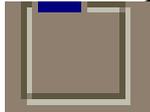
График тугма компонентаси оддий тугма компонентасига жуда ҳам ўхшашдир. **Delphi** тугмаларнинг стандарт мажмуасини таклиф қилади. Буларни дастурда яратиш учун танлаб олиш мумкин.

Check Box (белгилаш тугмаси) компонентаси ушбу тугма одатда дастурда «**Ха**» ёки «**Йўқ**» жавобини олиш керак бўлган вазиятларда фойдаланилади.

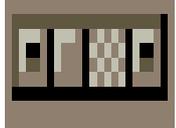
Гурухлашган компоненталар

Delphi дастурлаш тилида компоненталарни гурухлаштириб фойдаланиш мумкин.

Одатда бир-бирига алоқадор компоненталар битта жойда гурухлаштирилади.



-Croup Boxes гурухлашган рамка компонентаси. Гурухлашган рамка фақатгина бир тоифадаги компоненталарни ташкил топган бўлиши шарт эмас.



Scroll Bors компонентаси. Ушбу компонента турли ойналарни экранда кўринмай турган қисмларини кўриш ва бошқа мақсадларда ишлатилади.

Delphi: дастурлаш тилида **Scroll Bors** жуда яхши ташкил қилинган ва тули хил хоссалари мавжуд.

Position жойлашиш хассаси оний моментда қаерда жойлашишини белгилаб беради. Бу хоссанинг қиймати фақат бутун сонларни қабул қилади.

График компоненталар

График компоненталар формада фигура ва объектларни кўрсатиш ҳамда маълумотларни акслантириш учун фойдаланилади. Бу компоненталарни баъзилари жуда ҳам оддий баъзилари жуда ҳам мураккаб. Уларни хар бири аниқ вазифа учун мўлжалланган.



Image компонентаси

Бу компонентани формада расмларни жойлаштириш учун фойдаланилади. Расмни номи эса **Picture** хоссасида келтирилади. Бу расмларнинг кенгайтмаси **.BMP, ICO, WMF** бўлиши мумкин. Агар фойдаланувчи **Image** компонентасининг ўлчамини ўзгартиришни хоҳласа **Auto Size** хоссасига **True** қиймати бериши лозим. **Stretch** хоссасига **True** берилган бўлса, белгиланган соҳага жойлаштирилади.



Paint Box компонентаси. (Чизиш дарчаси)

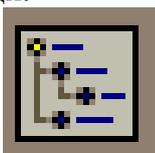
Ушбу компонента форманинг чекланган қисмида чизиш имконини беради. Чизиш дарчаси ўзининг чегараларини кўрсата олмайди. Ушбу қийинчиликни йўқотиш учун унинг ичига гурухланган панел **Group Boxes** қўйиш ва уни ўлчамини **Paint Box** компонента ўлчами билан бир хил қилиш лозим. Ана шундагина фойдаланувчига чизиш учун рамка пайдо бўлади.



Shapes (фигуралар) компонентаси

Фигуралар компонентаси ўз табиати билан жуда ҳам оддий. Фойдаланувчи содда геометрик фигуралардан дастурни бойитишда фойдаланишлари мумкин.

Фигураларнинг ва контурларнинг ранги мос равишда **Pen** ва **Brush** хоссалари ёрдамида ўрнатилади.

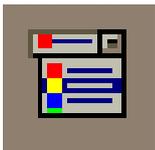


Outlines (схемалар) компонентаси

Схемалар иехархик тарзда ташкил қилинган маълумотларни акслантириш учун фойдаланилади. Мисол учун каталоглар дарахти.

Схема **Lines** ва **Items** хоссалари орқали мурожат қилиш мумкин бўлган қаторлардан ташкил топади.

Outlines Tile (схемали) хоссаси исталган конфигурациясини олиш учун фойдаланилади.



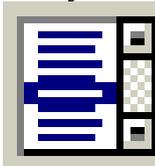
Color Grids компонентаси (ранглар тури)

Бу компонента чизиш ва фон ранглари бошқариш учун интерфейсни ифодалайди. Бу компонента ёрдамида чизиш ва фон рангини танлашда фойдаланилади.

Foreground (чизиш рангли) ва **Background** (фон ранги) хоссаларда белгиланган рангдан ташкил топган бўлади.

Файл ва каталогга мурожат қилиш компоненталари

Ушбу компоненталар файл ва каталоглар билан турли амалларни бажаришда фойдаланилади.



File List Box (файллар рўйхати панели) компонент файллар рўйхати панели компонентаси каталогга мурожат интерфейсини ифодалайди.

Mask хоссаси филтер ёки маскани ўрнатиш учун фойдаланилади.

Фойдаланувчи томонидан белгиланган файл номи **File Name** хоссаси орқали мурожат қилиши мумкин.

Delphi ning консол иловасини яратиш

*Delphi*да консолли иловаларни турли усулларда яратиш мумкин. Улардан энг осон усули куйидагича.

1. Delphi мухити ишга туширилади.

Пуск ⇒ Программы ⇒ Borland Delphi

2. Бош менюдан File пункти очилиб, у ердан New, кейин эса Other буйруқлари берилади.
File ⇒ New ⇒ Other

3. Форма ва лойихаларни сақлаш учун очилган махсус ойнадан “Console Application” пиктограммаси танланади ва ок тугмаси босилади.

4. Натижада экранда begin ва end ичига олинган лойиха ойнаси (.dpr кенгайтмали ном билан) очилади.

{TODO –oUser-cConsole Main:Insert code here} изохи ўрнига файлинг дастур матни киритилади. Тузилган дастурни ишга туширишдан олдин уни сақлаш керак бўлади. Уни сақлаш учун file ⇒ Save All буйругини бериш лозим. Хар бир лойиха алоҳида янги папкада сақланиши тавсия этилади. Лойиха файлини сақлашда алоҳида кўрсатилмаган ҳолатида ProjectN.dpr номли файл номини тавсия этади. Лойихани сақлаб бўлгандан кейин уни бажаришга ўтилади.

Дастур матни 1 да киритилган килограммни неча фунт эканлигини кўрсатувчи дастур кўрсатилган. Унда бирор буюмнинг оғирлиги фойдаланувчи томонидан килограммда киритилади. Натижа эса компьютер томонидан фунтда экранга чоп этилади.

```
{$APPTYPE CONSOLE}
```

```
Program Project2;
```

```
Var
```

```
k,f:Real;
```

```
Begin
```

```
Writeln('Buyum ogirlugini kilogrammda kiriting');
```

```
Writeln('va <Enter> tugmasini bosing');
```

```
Write('→');
```

```
Readln(k);
```

```
f:=k*0.4095;
```

```
Writeln(k:10:4,' kilogramm=', f:10:4,' funt');
```

```
Readln;
```

```
End.
```

Юқоридаги дастурда *{\$APPTYPE CONSOLE}* қатори мавжуд бўлиб, у изох кўринишида ёзилган. Лекин у, дастурнинг консолли илова эканлигини билдиради. Бундай дастурни тузишда албатта *{\$APPTYPE CONSOLE}* қатори ёзилиши шарт. Дастурни ишга тушириш учун Run менюсидан Run буйруғи танланади ёки F9 клавиши босилади.

Танлаш тугмаларини ўрнатиш

RadioGroup гуруҳи танлаш тугмалари иловалар яратишда бир неча вариантлардан бирини танлаш учун ишлатилади. Бу компонента Standart компонентлар палитрасида жойлашган. Унинг асосий хоссаси items бўлиб, у тугмалар номлари руйхатини ўзида сақлайди. Тугмалар номлари руйхатини киритишдан олдин RadioGroup тугмаси учун формадан жой ажратилади ва кейин items хоссаси кўрсатилиб, ундан уч нухтали тугмача босилади, натижада stringlist Editor ойнаси очилади. Бу ойнадан танлаш тугмалари номларининг хар қайсиси янги қатордан киритилади ва кейин “ok” тугмаси босилади. Формага RadioGroup гуруҳли танлаш тугмаси жойлаштирилганда и RadioGroup 1 ном билан ёзилади. Бу номни бошқа мос номга алмаштириш Caption хоссасига кириб амалга оширилади.

CheckBox компонентаси руйхатдан бир нечтасини танлаш имконини беради. CheckBox компонентаси Additional палитрасида жойлашган. RadioGroup боғлиқ

переключателларни, `CheckBox` эса боғлиқ бўлмаган переключателларни бирлаштиради. Бунда ёқувчи уч хил ҳолатда бўлиши мумкин:

- Ёқилган –тўғри белгиси;
- Ўчирилган–бўш белгиси;
- Нейтрал ҳолат –кўкиш рангда тўғри белгиси;

`CheckBox` нинг асосий хоссалари:

`AllowOryer` учинчи нейтрал ҳолат вариантыни ишлатишни тақиқлайди.

`Items` танлаш тугмалари номлари рўйхатини сақлайди.

Listbox ва Combobox компонентлари

`Listbox` компоненти рўйхат ва массив кўринишидаги маълумотларни экранда акс эттиришда ишлатилади. Маълумотларни киритишда эса `Edit` компонентасидан фойдаланилади. `Listbox` компонентаси `Standart` компоненталар палитрасида жойлашган. Бу тугмачани босиб формадан рўйхат учун жой ажратилади ва хоссалари аниқланади. Унинг айрим хоссалари:

`Items` – рўйхат элементини беради.

`Sorter`- рўйхат элементларини алфавит бўйича автоматик равишда тартиблайди.

`Clear` –барча рўйхат элементларини ўчиради.

`Combobox` компоненти рўйхат ва массив кўринишидаги маълумотларни экранга киритиш учун ишлатилади. У `Listbox` ва `Edit` компоненталарининг биргаликдаги ишини бир ўзи бажаради. Ташқи кўринишидан бу компонент оддий `Edit` киритиш қаторини эслатади. Унинг ўнг қисмида пастга белгиси бўлиб, киритилаётган маълумотларни кўриб бориш мумкин. Бу компонента `Standart` компоненталар палитрасида жойлашган. Бу тугмачани босиб формадан рўйхат учун жой ажратилади ва кейин хоссалари аниқланади. Унинг айрим асосий хоссалари:

`DropDownCount` – рўйхатдаги экранга чиқадиган маълумотлар сонини аниқлайди. Бу хоссанинг бошланғич қиймати 8 га тенг бўлади. Агар экранга чиқариладиган маълумотлар сонини 10 та бўлсин десак, унда унинг қийматини 10 га ўзгартириш керак бўлади. Агар киритилган маълумотлар ундан ортиқ бўлса, у ҳолда пастга ва юқорига силжитиш тугмачаси автоматик равишда пайдо бўлади;

`Style` – рўйхатдаги маълумотнинг кўринишини тасвирлайди;

`Text`- рўйхатдаги киритилган маълумот матн эканини билдиради.

StringGrid жадвал компонентаси

`StringGrid` жадвал компонентаси икки ўлчовли маълумотларни, масалан, матрица элементларининг қийматини экранда жадвал кринишида тасвирлаш, улар қийматини киритиш ва таҳрирлаш учун ишлатилади. Жадвалнинг қатор ва устун номерлари нолдан бошланади. Жадвалнинг устун ва қаторлар сонини кераклича ўзгартириш мумкин. Бу унинг хоссаси ёрдамида аниқланади. Жадвалнинг шар бир кесишган устун ва сатри ячейка дейилиб, унга киритилган маълумот символ қатори қилиб аниқланади. Масалан, (3,5) ячейка тўртинчи устун ва олтинчи қаторда жойлашган.

`StringGrid` жадвал компонентасининг асосий хоссалари:

`ColCount` - жадвалдаги устунлар сонини аниқлайди;

`RowCount` - жадвалдаги сатрлар сонини аниқлайди;

`FixedCols` - фиксирланган устунлар сонини аниқлайди;

`FixedRows` - фиксирланган сатрлар сонини аниқлайди;

`Options` -жадвал ҳолатини аниқлайди (аниқлаш унинг параметрларига асосан бажарилади, масалан `Goediting` параметр `true` қийматига эга бўлса ячейкани таҳрирлаш мумкин, акс ҳолда мумкин эмас. Бу параметрларни аниқлаш учун `Options` хоссасига ўтиб, у икки марта тез-тез босилади);

`ColWidths` - жадвалдаги ҳар бир устун кенлигини аниқлайди;

`DefaultColWidth` - жадвалнинг бошланғич устунлар кенлигини аниқлайди;

`DefaultRowHeight` - жадвал сатрининг бошланғич баландлигини аниқлайди;

`FixedColor` - фиксирланган ячейка рангини аниқлайди;

`RowHeights` - жадвал сатри баландлигини аниқлайди;

`Cells` – символ қаторли икки ўлчамли массивни аниқлайди.

Символлар ва қаторлар

Белгили тип *Char* хизматчи сўзи билан эълон қилиниб, бу типнинг қийматлари 1 байт жой эгаллайди. Тилнинг барча белгилари бу типнинг қийматлар соҳасига тегишлидир. Белгили қийматни уни қўштирноқ ичига олиб ёки # белгисидан кейин керакли белгининг ASCII кодини ёзиб аниқлаш мумкин. Мисол: "А" , ёки # 60.

Қатор - бу белгиларнинг оддий кетма-кетлигидир: 'Ab21#9!cd', 'dasturchi Saidkarim Gulomov'. Қатор бўш ёки битта белгили бўлиши ҳам мумкин. Қаторли ўзгарувчи узунлиги 255 гача бўлган белгили қийматларни қабул қилиши мумкин. Умуман олганда, ҳар бир қаторли ўзгарувчига хотирадан 256 байт жой ажратилади. Хотирани тежаш учун қаторнинг типини куйидагича кўрсатиш мақсадга мувофиқдир: *String*[N], N - қатордаги белгилар сони. Бу ҳолда белгили ўзгарувчи учун N байт жой ажратилади. Белгилар ва қаторлар устида бир қанча амаллар бажариш мумкин, яъни қатордан керакли бўлакни кесиб олиш, қаторларни бир-бирига қўшиш ва натижада янги қаторлар ҳосил қилиш. Сатрли ва символли белгилар устида турли амаллар бажариш мумкин.

Ёзилиши	Вазифаси
Function Length(S):Integer	S сатрли ўзгарувчидаги белгилар сонини аниқлайди
Function Copy(S; Index, Count: Integer): string;	S сатрли ўзгарувчидаги Index - белги Count та белгидан нусха олиш
Function Concat(s1 [, s2,..., sn]: string): string;	S1 дан sn тагача бўлган сатрли ўзгарувчиларни битта сатрли ўзгарувчига бирлаштириш
function Pos(Substr: string; S: string): Integer;	Substr сатри S сатридан изланади. Агарда изланган сатр топилмаса натижа нолга тенг бўлади
procedure Delete(var S: string; Index, Count:Integer);	S сатрдаги Index – белгидан Count та белгини ўчириб ташлайди
procedure Insert(Source: string; var S: string; Index: Integer);	S сатрига Index – белгидан бошлаб Source сатрини жойлаштиради

Массивлар. Массивлар устида амаллар.

Программалашда энг кўп қўлланиладиган программа объектларининг бири бўлган массивлар билан танишиб чиқамиз. Массив - бу бир хил типли, чекли қийматларнинг тартибланган тўпламидир. Массивларга мисол сифатида математика курсидан маълум бўлган векторлар, матрицалар ва тензорларни кўрсатиш мумкин. Программада ишлатилувчи барча массивларга ўзига хос исм бериш керак. Массивнинг ҳар бир ҳадига мурожаат эса унинг номи ва ўрта қавс ичига олиб ёзилган тартиб ҳади орқали амалга оширилади. Массивнинг зарур ҳадига мурожаат куйидагича амалга оширилади: <массив номи>[<индекс>]

бу ерда <индекс> массив ҳадининг жойлашган жойини аниқлаш учун тартиб қиймати.

Умуман олганда, <индекс> ўрнида ифода қатнашиши ҳам мумкин. Индексни ифодаловчи ифоданинг типини индекс типини деб аталади. Индекс типининг қийматлар тўплами албатта номерланган тўплам бўлиши, шу билан бир қаторда массив ҳадлари сонини аниқлаш ва уларнинг тартибини белгилаши керак. Массивларни эълон қилишда индекс типини билан бир қаторда массив ҳадларининг типини ҳам кўрсатилиши керак. Бир ўлчамли массивни эълон қилиш куйидагича амалга оширилади: array [<индекс типини>] of <ҳад типини>;

Кўпинча <индекс типини> сифатида чекланма типлардан фойдаланилади, чунки бу типга тегишли тўплам тартибланган ва қатъий номерлангандир. Мисол учун 100 та ҳақиқий сонли ҳадлардан иборат массив куйидагича эълон қилинади: array [1..100] of real;

Массивларни эълон қилиш ҳақида тўлиқроқ маълумот бериш учун турли типдаги индексларга оид мисолларни эътиборингизга ҳавола қиламиз:

```
array [1000..5000] of integer;
array [-754..-1] of byte;
array [0..100] of real;
array [0..10] of boolean;
array [10..25] of char;
```

Кўп ўлчамли массивлар

Бир ўлчамли массивларнинг ҳадлари скаляр микдорлар бўлган эди. Умумий ҳолда эса массив ҳадлари ўз навбатида яна массивлар бўлиши мумкин, агар бу массивлар скаляр микдорлар бўлса натижада икки ўлчамли массивлар ҳосил бўлади. Икки ўлчамли массивларга мисол сифатида математика курсидаги матрицаларни келтириш мумкин. Агар бир ўлчамли массивнинг ҳадлари ўз навбатида матрицалар бўлса натижада уч ўлчовли массивлар ҳосил қилинади ва ҳ.к. **Икки ўлчамли массив типини кўрсатиш қуйидагича бажарилади:**

```
array [<индекс типи>] of array [<индекс типи>] of <скаляр тип>;
```

Икки ўлчамли массив индексларининг типлари турли хил ҳам бўлиши мумкин. Бу ҳолни қуйидаги мисол устида кўриб чиқайлик:

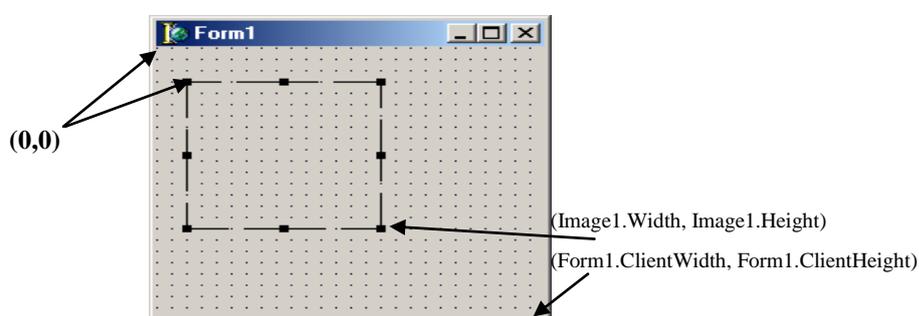
```
const n = 24;  
type hafkun = (dush, sesh, chor, pay, jum, shan, yaksh);  
Ishkun = dush..jum;  
detson = array [1..n] of char;  
var A: array [boolean] of array [1..n] of char;  
B: detson;  
C: array [1..365] of detson;
```

Delphi нинг график имкониятлари

Delphi дастурчига турли хилдаги схемалар, чизмалар ва иллюстрациялар билан ишлаш имкониятларини беради. Дастур графикани объект (форма ёки компонент Image) сиртида ҳосил қилади. Объект сирти Canvas хусусиятига мос келади. График элемент (тўғри чизик, айлана, тўғри тўртбурчак ва ҳ.к.) ларни объект юзасида ҳосил қилиш учун Canvas дан фойдаланилади. Масалан, *Form1.Canvas.Rectangle(10,10,50,50)* инструкцияси дастур ойнасида тўғри тўртбурчак ҳосил қилади. *Чизма ҳосил бўлувчи сирт.*

Юқорида айтиб ўтилганидек, графикани ҳосил қилувчи сирт (юза) Canvas хусусиятига тўғри келади. Ўз навбатида Canvas хусусияти TCanvas типидagi объектдир. Бу тип услублари график примитивларни (нуқта, чизик, айлана ва ҳ.к.) ҳосил бўлишини таъминлайди, хусусияти эса ҳосил бўлувчи графикани характеристикаларини: ранги, чизик қалинлиги ва тури; бўялувчи худудни ранги ва кўринишини; харфни характеристикаларини беради. Canvas «сирт», «чизиш учун юза» сифатида таржима қилинади. Чизиш юзаси алоҳида нуқта – пикселлардан ташкил топади. Пикселни жойлашуви горизонтал (X) ва вертикал (Y) координаталар билан характерланади.

Чап юқоридаги нуқта координатаси (0,0). Координаталар юқоридан пастга ва чапдан ўнгга қараб ўсиб боради.



1-расм. Чизиш юзаси нуқта координаталари.

Чизиш юзаси ўлчамларини иллюстрация (Image) худуди учун Height ва Width, форма учун эса ClientHeight ва ClientWidth лар аниқлаш мумкин.

Қалам ва мўйқалам.

Одатда рассом сурат чизиш учун қалам ва мўйқаламдан фойдаланади. Delphi нинг график имкониятлари ҳам қалам ва мўйқаламдан фойдаланиш имкониятларини яратади. Қаламдан чизиқ ва контур чизишда, мўйқаламдан эса контур билан чегараланган юзани бўяш учун фойдаланилади.

Қалам ва мўйқалам графикани чизиш юзасида ҳосил қилишда мос равишда Pen(қалам) ва Brush(мўйқалам) хусусиятларига ҳосдир. Шу билан бирга улар TPen ва TBrush типларига тегишлидир.

Қалам.

Қаламдан нукта, чизиқ, геометрик шакллар: тўғри тўртбурчак, айлана, эллипс ва ҳ.к. ларни чизишда қурол сифатида фойдаланилади. TPen объект хусусияти 1-жадвалда келтирилган.

1-жадвал

Хусусият	Вазифаси
Color	Чизиқ (контур) ранги
Width	Чизиқ қалинлиги
Style	Чизиқ кўриниши
Mode	Тасвирлаш режими

Color хусусияти чизувчи қалам рангини белгилайди. Қуйидаги 2-жадвалда PenColor хусусиятлари келтирилган:

2-жадвал

Константа	Ранг	Константа	Ранг
<i>clBlack</i>	қора	<i>clSilver</i>	кумушранг
<i>clMaroon</i>	каштанранг	<i>clRed</i>	қизил
<i>clGreen</i>	яшил	<i>clLime</i>	салатранг
<i>clOlive</i>	оливковий	<i>clBlue</i>	кўк
<i>clNavy</i>	тўқ кўк	<i>clFuchsia</i>	Fuchsia
<i>clPurple</i>	атиргулранг	<i>clAqua</i>	ёруј кўк
<i>clTeal</i>	Teal	<i>clWhite</i>	оқ
<i>clGray</i>	кулранг		

Width хусусияти чизувчи қалам қалинлигини (пикселда) белгилайди.

Масалан, *Canvas.Pen.Width:=2* чизиқ қалинлиги 2 пикселга тенг бўлади.

Style хусусияти чизилувчи чизиқнинг турини белгилайди. Style компонентлари 3-жадвалда келтирилган.

Константа	Чизиқ кўриниши
<i>psSolid</i>	Тўғри чизиқ
<i>psDash</i>	Узун штрихли пунктир чизиқ
<i>psDot</i>	Қисқа штрихли пунктир чизиқ
<i>psDashDot</i>	Узун-қисқа штрихли пунктир чизиқ
<i>PsDashDotDot</i>	Бир узун ва икки қисқа штрихли пунктир чизиқ
<i>PsClear</i>	Кўринмас чизиқ

Мўйқалам

Мўйқалам (Canvas.Brush)дан ёпиқ соҳаларни тўлдириш учун фойдаланилади, масалан, геометрик шаклларни бўяш ва ҳ.к. Мўйқалам объект сифатида қуйидаги икки хусусиятни ўз ичига олади:

Color – бўялувчи соҳа ранги

Style – тўлдирувчи соҳа типи

Масалан, контурнинг ички соҳаси бўялиши ёки штрихланиши мумкин.

Color хусусияти сифатида Tcolor нинг барча ўзгармасларидан фойдаланиш мумкин. Style хусусиятлари 4-жадвалда келтирилган.

4-жадвал

Константа	Бўялувчи соҳа типи
bsSolid	тўлиқ
bsClear	Бўялмайди
bsHorizontal	горизонтал штрихлаш
bsVertical	вертикал штрихлаш
bsFDDiagonal	олдинга эгилган диагонал штрихлаш
bsBDDiagonal	орқага эгилган диагонал штрихлаш
bsCross	горизонтал-вертикал сеткали штрихлаш
bsDiagCross	диагонал сеткали штрихлаш

Матн ҳосил қилиш

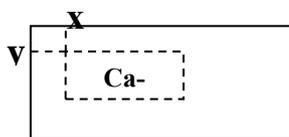
График объект сиртида матнни ҳосил қилиш учун TextOut дан фойдаланилади. TextOut нинг ёзилиш формати куйидагича:

Объект.Canvas.TextOut(x,y,Text);

Бу ерда Объект – матн ҳосил бўлувчи объект номи;

x,y – матн бошланувчи координата

Text – ҳосил бўлувчи белги катталикидаги матн ёки сатрли ўзгарувчи.



3-расм. Матн ҳосил бўлувчи соҳа координатаси

ҳосил бўлувчи матн белгилари Canvas объектига мувофиқ келувчи Font хусусияти орқали ифодаланади. Font хусусияти TFont объектига тегишли бўлиб, 5-жадвалда белги характеристикалари ва қўлланилувчи услублари келтирилган.

5-жадвал

Хусусият	Аниқланиши
Name	Фойдаланилувчи шрифт. Қиймат сифатида шрифт номи ёзилади, масалан, Arial Cyr
Size	пунктларда ифодаланилувчи шрифт ўлчами. Пункт-полиграфияда қўлланилувчи ўлчов бирлиги бўлиб, у тахминан 1/72 дюйм ¹ га тенг
Style	белгини ёзиш усули, куйидагича бўлиши мумкин: оддий, қалин, курсив, остига чизилган, устига чизилган. Булар куйидаги константалар ёрдамида амалга оширилади: <i>fsBold</i> (қалин), <i>fsItalic</i> (курсив), <i>fsUnderline</i> (остига чизилган), <i>fsStrikeOut</i> (устига чизилган). style бир нечта усулларни комбинация қилиши мумкин. Масалан, қалин курсив ҳолатини ифодалаш: <i>Объект.Canvas.Font:=[fsBold, fsItalic]</i>
Color	Белги ранги. Қиймат сифатида TColor константаларидан фойдаланиш мумкин.

Қуйидаги дастур қисми TextOut функциясини қўллаш учун мисол бўла олади:

with Form1.Canvas do

begin

Brush.Color:=Form1.Color;

Font.Size:=14;

Font.Style:=[fsItalic, fsBold];

```
TextOut(10,10,'Salom, Delphi!');
```

End;

Матн эканда ҳосил бўлгандан сўнг кўрсаткич унинг ўнг юқори бурчагига силжийди.

Баъзида матндан сўнг бирор маълумотни чиқариш керак бўлиб қолади. Агар матн узунлиги номаълум бўлса кўрсаткич турган координатани аниқлаш мушкул. Масалан «сўм» сўзини рақамдан кейин ҳосил қилиш керак бўлсин. Бундай ҳолатларда кўрсаткич турган координатадан бошлаб давом этиш учун PenPos дан фойдаланишга тўғри келади:

```
TextOut(10,10,SumPr); // SumPr – String типли катталик  
TextOut(PenPos.X, PenPos.Y, ' сум');
```

Тўғри чизиқ.

Delphi да тўғри чизиқ ҳосил қилиш учун LineTo дан фойдаланилади. Унинг ёзилиш формати куйидагича:

Компонент.Canvas.LineTo(x,y)

LineTo тўғри чизиқни қалам (кўрсаткич) турган координатадан бошлаб x,y – нуқтагача чизади. Шунинг учун чизиқнинг бошланғич нуқтасини керакли жойга ўрнатиб олиш лозим бўлади. Бунда биз MoveTo га мурожаат қиламиз:

Компонент.Canvas.MoveTo(X0,Y0)

Чизиқнинг кўриниши (ранги, калинлиги ва тури) Pen объекти билан ифодаланади.

Айлана ва эллипс.

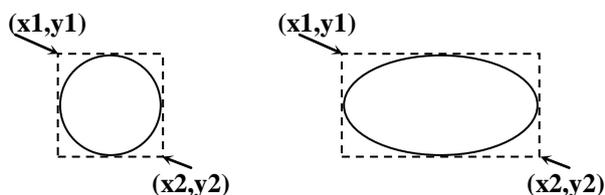
Ellipse услуби эллипс ва айлана чизиш учун қўланилади. Ellipse нинг ёзилиш формати куйидагича:

Объект.Canvas.Ellipse(x1,y1,x2,y2)

бу ерда, объект – чизма ҳосил бўлувчи объект номи;

$x1,y1,x2,y2$ – ҳосил бўлувчи айлана ёки эллипсга ташқи чизилган тўғри тўртбурчакнинг мос равишда юқори чап ва куйи ўнг нуқталарини координаталари (4-расм).

Чизиқнинг кўриниши (ранги, калинлиги ва тури) Pen объекти билан ифодаланади.



4-расм.

Ёй.

Ёй ҳосил қилиш учун Arc услубидан фойдаланилади. Унинг ёзилиш формати куйидагича:

Объект.Canvas.Arc(x1,y1,x2,y2,x3,y3,x4,y4)

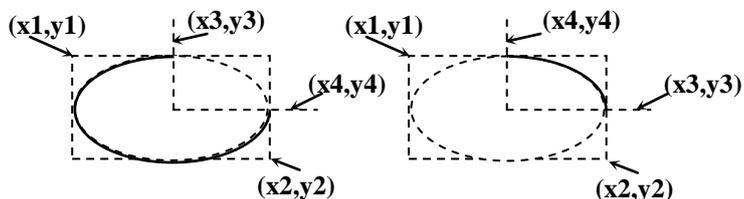
бу ерда, объект – ёй чизилувчи объект номи;

$x1,y1,x2,y2$ – ҳосил бўлувчи ёйни давом эттириб ҳосил қилинувчи эллипс (айлана)га ташқи чизилган тўғри тўртбурчакнинг мос координаталари;

$x3,y3$ – ёйнинг бошланғич нуқтаси;

$x4,y4$ – ёйнинг тугаш нуқтаси.

Шуни айтиб ўтиш лозимки, ёй соат стрелкаси йўналишига қарама-қарши йўналишда чизилади (5-расм).



5-расм.

Чизиқнинг кўриниши (ранги, қалинлиги ва тури) Pen объекти билан ифодаланади.

Тўғри тўртбурчак.

Тўғри тўртбурчак ҳосил қилишда Rectangle услубидан фойдаланилади. Унинг ёзилиш формати қуйидагича:

Объект. Canvas.Rectangle(x1,y1,x2,y2)

Бу ерда

объект – тасвир ҳосил бўлувчи объект номи;

x1,y2,x2,y2 – тўғри тўртбурчакнинг мос равишда юқори чап ва қуйи ўнг бурчак координаталари.

RoundRec услуби ҳам тўғри тўртбурчак чизади, фақат Rectangle дан фарқи шундаки, унинг бурчаклари юмалоқ (силлик) шаклда бўлади. Ёзилиш формати:

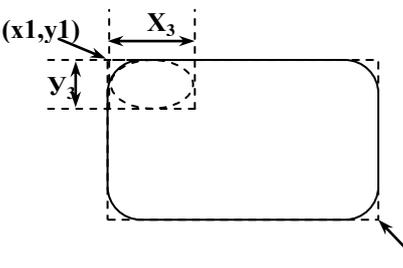
Объект. Canvas.RoundRec(x1,y1,x2,y2)

Бу ерда

объект – тасвир ҳосил бўлувчи объект номи;

x1,y2,x2,y2 – тўғри тўртбурчакнинг мос равишда юқори чап ва қуйи ўнг бурчак координаталари;

x3,y3 – юмалоқ ҳосил қилишда қўлланилувчи эллипс ўлчамлари (6-расм).



Кўпбурчак.

(x1,y1) Polygon дан фойдаланиб кўпбурчак чизиш мумкин. Polygon

TPoint типли массивни параметр сифатида қабул қилади. ҳар бир массив элементи ўзида кўпбурчакнинг битта бурчаги координатаси(x,y) ни сақлайди. Polygon эса шу нукталарни кетма-кет тўғри чизиқлар билан туташтириб чиқади. Чизиқнинг кўриниши (ранги, қалинлиги ва тури) Pen объекти билан ифодаланади.

Қуйида учбурчак чизиш учун дастур қисми келтирилган:

```
procedure TForm1.Button1Click(Sender:TObject);
```

```
var
```

```
  pol: array[1..3] of TPoint; //учбурчак нукталари координатаси
```

```
{ TPoint=record
```

```
  X:longint;
```

```
  Y:longint
```

```
End;}
```

```
Begin
```

```
  Pol[1].x:=10;
```

```
  Pol[1].y:=50;
```

```
  Pol[1].x:=40;
```

```
  Pol[1].y:=10;
```

```
  Pol[1].x:=70;
```

```
  Pol[1].y:=50;
```

```
  Form1.Canvas.Polygon(pol);
```

```
End;
```

Сектор.

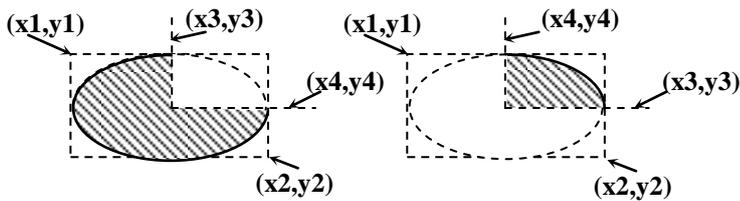
Эллипс ёки айлана секторини ҳосил қилишда Pie услубидан фойдаланилади. Pie нинг умумий ёзилиш формати:

Объект. Canvas.Pie(x1,y1,x2,y2,x3,y3,x4,y4)

бу ерда

объект – ёй чизилувчи объект номи;

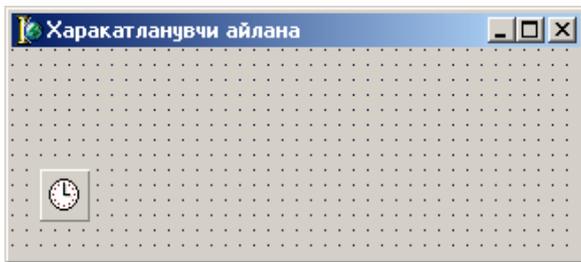
x_1, y_1, x_2, y_2 – ҳосил бўлувчи секторни давом эттириб ҳосил қилинувчи эллипс (айлана)га ташқи чизилган тўғри тўртбурчакнинг мос координатлари;
 x_3, y_3 – секторнинг бошланғич нуқтаси;
 x_4, y_4 – секторнинг тугаш нуқтаси.



7-расм.

Мультипликация.

Мультипликация дейилганда одатда ҳаракатланувчи ёки ўзгарувчи расмни тушунилади. Оддий ҳолатларда расм ҳаракатланиши ёки ўзгариши мумкин. 2-осил қилинган расм (чизик, айлана, ёй ва х.к.)ларни силжитиш жуда оддий: аввал расм ҳосил қилинади, бир оздан сўнг уни тозаланади ва яна янгитдан аввалги жойидан бошқа ерда ҳосил қилинади. Бундай алмаштириш бир маромда давом эттирилса, натижада тасвир экран бўйлаб ҳаракатланаётган ўхшайди. Қуйидаги кичик дастур ёрдамида айланани дастур ойнасининг чап чегарасидан ўнг чегарасига қараб ҳаракатлантиришимиз мумкин. 8-расмда форма кўриниши келтирилган.



8-расм.

Дастур матни

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls;
type
  TForm1 = class(TForm)
    Timer1: TTimer;
    procedure Timer1Timer(Sender: TObject);
    procedure FormActivate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
  x,y,dx:byte;
implementation
{$R *.dfm}
procedure Ris;
begin
  { айланани куринмас килиш }
  form1.canvas.Pen.Color:=form1.Color;
  form1.Canvas.Ellipse(x,y,x+10,y+10);
  x:=x+dx;
```

```

{айланани янги жойда ҳосил қилиш}
form1.Canvas.Pen.Color:=clBlack;
form1.Canvas.Ellipse(x,y,x+10,y+10);
end;
procedure TForm1.Timer1Timer(Sender: TObject);
begin
ris;
end;
procedure TForm1.FormActivate(Sender: TObject);
begin
x:=0;
y:=10;
dx:=5;
timer1.Interval:=50;
form1.Canvas.Brush.Color:=form1.Color;
end;
end.

```

Асосий ишни айланани ўчириб янги жойда ҳосил қилувчи Ris процедураси бажаради. Айланани ўчиришни унинг рангини 9-расм.



Timer

форма рангига ўзгартириш йўли билан амалга оширилади.

Форма ёки дастурда эътибор берган

бўлсангиз визуал бўлмаган компонент Timer (таймер)дан фойдаландик. У ёрдамида ҳаракатни вақт бўйича амалга оширилиши таъминланган. Timer компоненти компонентлар палитрасининг System бўлимида жойлашган (9-расм). Timer хусусиятлари 10-жадвалда келтирилган.

10-жадвал

Константа	Бўялувчи соҳа типи
Name	Компонент номи
Interval	Миллисекундларда берилувчи OnTimer генерацияси
Enabled	Ишга рухсат бериш. Қиймат true бўлса рухсат берилади, false бўлса берилмайди.

Timer компоненти OnTimer ходисасини ишга туширади. OnTimer вақтли ходисаси миллисекундларда ўзгаради ва Interval хусусиятларига мос келади. Enabled хусусияти эса дастурда таймерни «ишга тушириш» ёки «тўхтатиш» имкониятини яратади. Агар Enabled True бўлса OnTimer ишламайди.

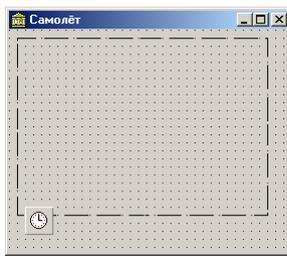
Битли тасвирлардан фойдаланиш.

Юқоридаги мисолда тасвирни ўзимиз ҳосил қилиб олдик. Энди эса қандай қилиб бир мураккаб тасвирни бошқасини фонида ҳаракатланишини кўриб ўтамиз. Масалан, шаҳар тасвири фонида самолётни юргизишни олайлик. Суратни силжитиш эффекти суратни бир неча жойда вақти-вақти билан қайтадан чизиш усули билан ташкил қилиниши мумкин. Бундан олдин, тасвирни янги нуқтада чиқаришдан аввал уни аввалгиси ўчирилади. Суратни ўчириш тўлиқ фонни бошқатдан ёки фақат ўша қисмини чизиш йўли билан амалга оширилиши мумкин. Биз кўриб ўтадиган дастурда иккинчи йўлдан фойдаланамиз. Тасвир Image компонентининг Canvas хусусиятида Draw услуби билан чиқарилади, тозалаш эса фоннинг керакли қисмини нусхасини олиш йўли (CopyRect услуби) билан амалга оширилади.

Дастур форма кўриниши 10-расмда келтирилган.

Image компоненти фонни чиқариш учун фойдалана-

риш учун, Timer компоненти эса ҳаракатланади. 10-расм



Дастур матни

```
unit anim_;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;
type
  TForm1 = class(TForm)
    Image1: TImage;
    Timer1: TTimer;
    Procedure FormActivate(Sender:TObject);
    Procedure Timer1Timer(Sender:TObject);
    Procedure FormClose(Sender:Tobject; var Action:TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
  Back, bitmap, Buf:Tbitmap; //фон, тасвир, буфер
  BackRct, bitmapRct, BufRct:Trect; //фон, тасвир, буфер сохаси
  x,y:integer; // тасвирнинг юкори чап бурчак координатаси
  w,h:integer; // тасвир улчами
implementation
{$R *.DFM}
Procedure TForm1Activate(Sender:TObject);
Begin
  Back:=Tbitmap.Create; //фон
  Bitmap:=Tbitmap.Create; //тасвир
  Buf:=Tbitmap.Create; //буфер
  // юклаш ва фонни хосил қилиш
  Back.LoadFromFile('factory.bmp');
  Form1.Image1.Canvas.Draw(0,0,Back);
  // харакатланувчи тасвирни юклаш
  bitmap.LoadFromFile('aplane.bmp');
  bitmap.Transparent:=true;
  bitmap.TransparentColor:=bitmap.Canvas.pixels[1,1];
  // фон сохаси нухасини саклаш учун буфер ташкил этиш
  w:=bitmap.Width;
  h:=bitmap.Height;
  Buf.Width:=w;
  Buf.Height:=h;
  Buf.Palette:=Back.Palette;
  Buf.Canvas.CopyMode:=cmSrcCopy;
  BufRct:=Bounds(0,0,W,H);
  X:=-40;
  Y:=20; //сакланувчи фон сохасини аниқлаймиз
  BackRct:=Bounds(x,y,w,h); // ва уни саклаймиз
  Buf.Canvas.CopyRect(BufRct,Back.Canvas,backRct);
End;
```

```

Procedure TForm1.Timer1Timer(Sender:TObject);
Begin // фонни тиклаймиз (буфердан) расмни йук киламиз
  Form1.Image1.canvas.Draw(x,y,Buf);
  X:=x+2;
  If x>form1.Image1.width then x:=-40; //сакланувчи фон сохасини аниклаймиз
  BackRct:=Bounds(x,y,w,h); // унинг нухасини саклаймиз
  Buf.Canvas.CopyRect(BufRct,Back.Canvas,BackRct); //расмни чиқарамиз
  Form1.Image1.canvas.Draw(x,y,bitmap);
End;
Procedure TForm1.FormClose(Sender:Tobject; var Action: TcloseAction);
Begin
  Back.Free;
  Bitmap.Free;
  Buf.Free;
End;
end.

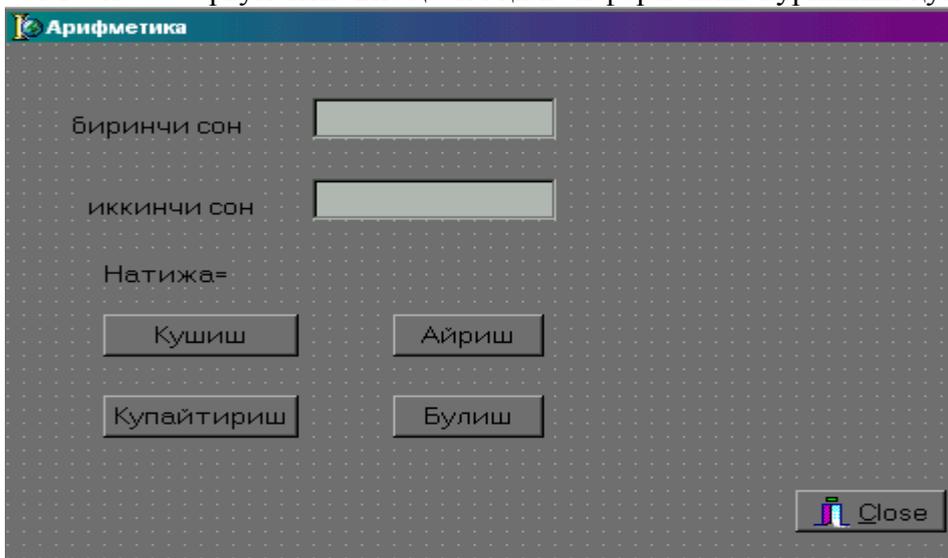
```

Илова

1. Чизикли алгоритмларга доир дастурлар тузиш.

1.1-масала. Delphi да чизикли дастурга мисол сифатида арифметик амалларни ҳисоблаш дастурини кўриб чиқамиз.

Бу дастурни тузиш учун янги формада киритиладиган иккита сон учун алоҳида «ойна» очамиз. Бунинг учун компоненталар рўйхатидан Edit танланади. Edit да киритиладиган маълумотлар матнли маълумот бўлади, шунинг учун биз киритган матн шаклидаги сонларни ҳақиқий сонга айлантириш керак. Бунинг учун Delphi да strtofloat(edit1.text) функцияси ишлатилади. Бунда strtofloat функция номи edit1.text эса аргументи. Биз ҳосил қилган форманинг кўриниши қуйидагича бўлади:



Бу формадаги биринчи сон, иккинчи сон, Натижа ёзувлари компоненталар рўйхатидан Label ни танлаш билан ҳосил қилинган. Одатда формада Label ишлатилса, унинг ишлатилиш сонига қараб Label1, Label2 ва ҳаказо шаклда чиқади, буни ўзгартириш учун Caption да Label1, Label2 ва ҳаказоларни ўчириб ўрнига дастурга мос матн киритилади.

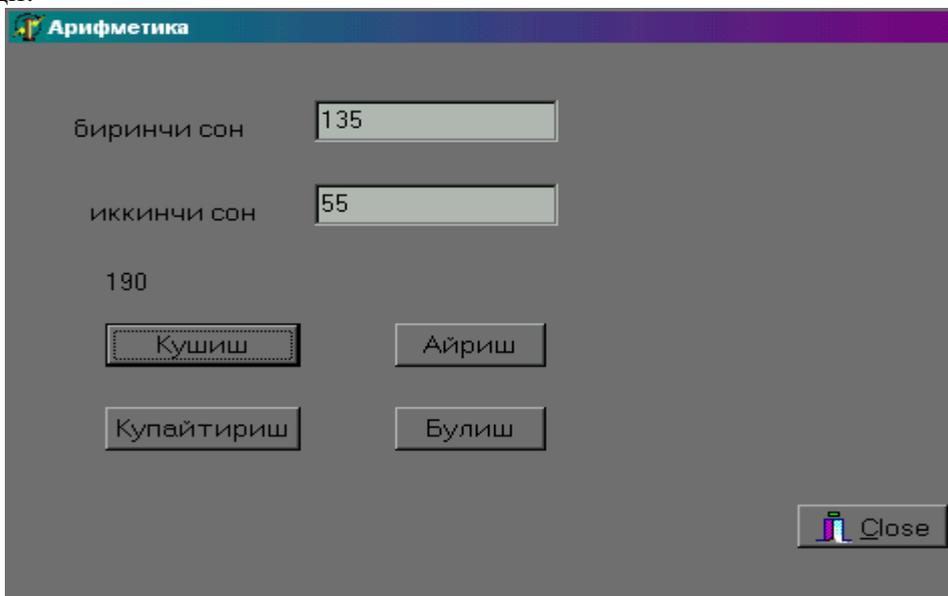
Формада қўшиш, айриш, қўпайтириш, бўлиш амалларига стандарт тугмалардан Button компонентаси танланган. Бунда ҳам Caption да Button лар ўрнига мос равишда қўшиш, айриш, қўпайтириш, бўлиш деб ўзгартирилган.

Delphi да тузилган дастурдан нормал ҳолда чиқиб кетиш учун стандарт х тугмасидан ташқари яна бошқа тугма иштлилади. Бунинг учун компоненталар меню сатридан Additional банди танланади ва унинг ичидан BitBtn компонентаси танланади. Агар Caption да &Close деб ўзгартирила, Kind да Vkclose деб ёзиб қўйсақ, у ҳолда ёпиш тугмасида тагига чизилган Close ёзуви ҳосил бўлади.

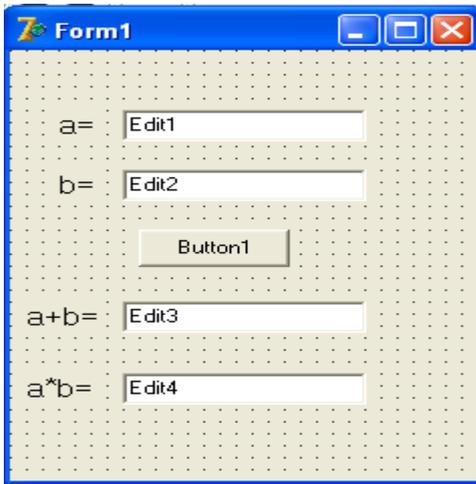
Амалларни бажариб, натижани олиш учун мос тугмаларнинг устида сичқончанинг чап тугмасини икки марта босилса, у ҳолда шу тугмага мос бўлган процедуранинг ичига автоматик кирилади. Бунда ҳар бир тугмага мос бўлган процедураларнинг кўриниши қуйидагича бўлади:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
label3.Caption:=floattostr(strtfloat(edit1.Text)+ strtfloat(edit2.Text));
end;
procedure TForm1.Label3Click(Sender: TObject);
begin
label3.Caption:=floattostr(strtfloat(edit1.Text)- strtfloat(edit2.Text));
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
label3.Caption:=floattostr(strtfloat(edit1.Text)* strtfloat(edit2.Text));
end;
procedure TForm1.Button4Click(Sender: TObject);
begin
label3.Caption:=floattostr(strtfloat(edit1.Text)/strtfloat(edit2.Text));
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
label3.Caption:=floattostr(strtfloat(edit1.Text)- strtfloat(edit2.Text));
end;
```

Бу дастур бажарилгандан кейинги экранда ҳосил бўладиган натижанинг кўриниши қуйидагича бўлади:



1.2-масала. А ва Б иккита ҳақиқӣ сонлар берилган. Уларнинг йиғиндиси ва кўпайтмасини ҳисоблаш дастурини тузинг.



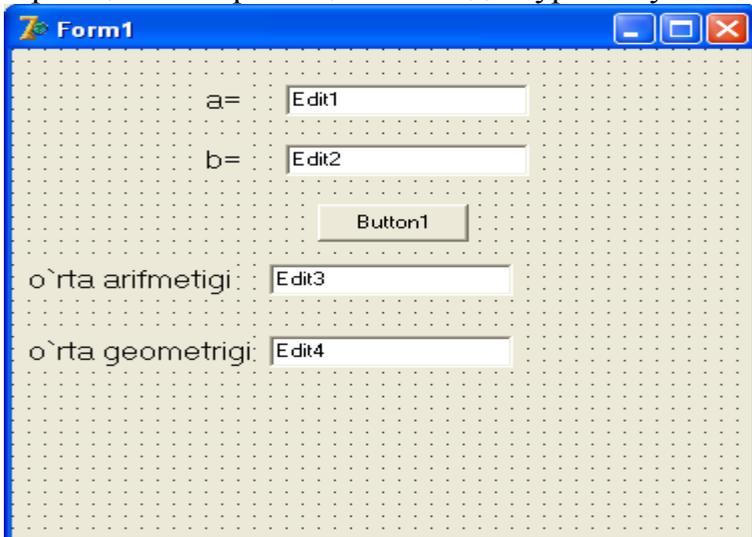
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b,s,p:real;
begin
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
s:=a+b;
p:=a*b;
Edit3.Text:=FloatToStr(s);
Edit4.Text:=FloatToStr(p);
end;
end.

```

1.3-масала. Иккита мусбат сон берилган, бу сонларнинг ўрта арифметик ва ўрта геометрик қийматларини ҳисоблаш дастурини тузинг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b,s,p:real;
begin
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
s:=(a+b)/2;
p:=sqrt(a*b);

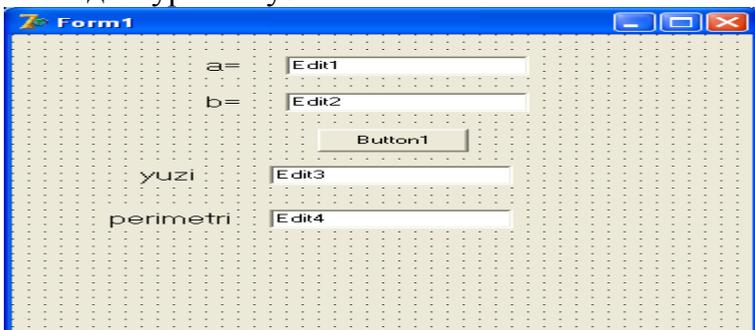
```

```

Edit3.Text:=FloatToStr(s);
Edit4.Text:=FloatToStr(p);
end;
end.

```

1.4-масала. Томонлари a ва b га тенг бўлган тўғри тўртбурчакнинг юзи ва периметрини ҳисоблаш дастурини тузинг.



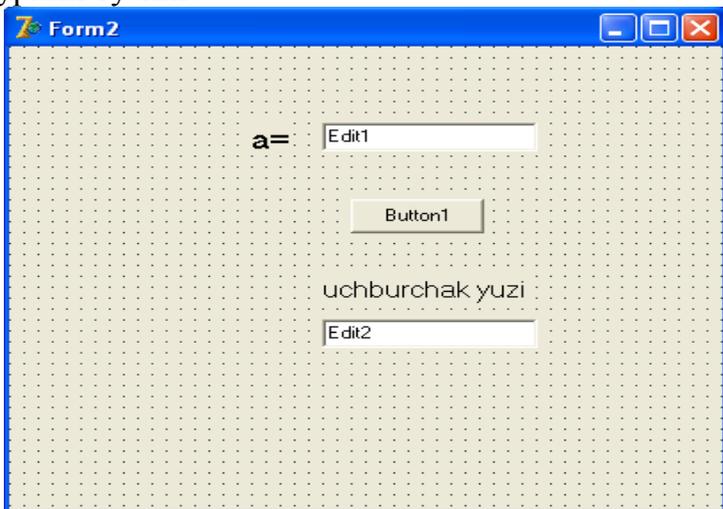
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var
a,b,s,p: real;
begin
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
s:=a*b; p:=2*(a+b);
Edit3.Text:=FloatToStr(s);
Edit4.Text:=FloatToStr(p);
End; end.

```

1.5-масала. Тенг томонли учбурчакнинг томони A га тенг. Учбурчакнинг юзини топиш дастурини тузинг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,s:real;
begin
a:=StrToFloat(Edit1.Text);
S:=(sqrt(3))*a*a/4;
Edit2.Text:=FloatToStr(S);
end;

```

end.

1.6-masala. Координаталари x_1, y_1 ва x_2, y_2 га тенг бўлган нуқталар орасидаги масофани ҳисоблаш дастурини тузинг.

Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var x1,x2,y1,y2,d:real;
begin
x1:=StrToFloat(Edit1.Text);
x2:=StrToFloat(Edit2.Text);
y1:=StrToFloat(Edit3.Text);
y2:=StrToFloat(Edit4.Text);
d:=sqrt(sqr(x2-x1)+sqr(y2-y1));
Edit5.Text:=FloatToStr(d);
end;
end.
```

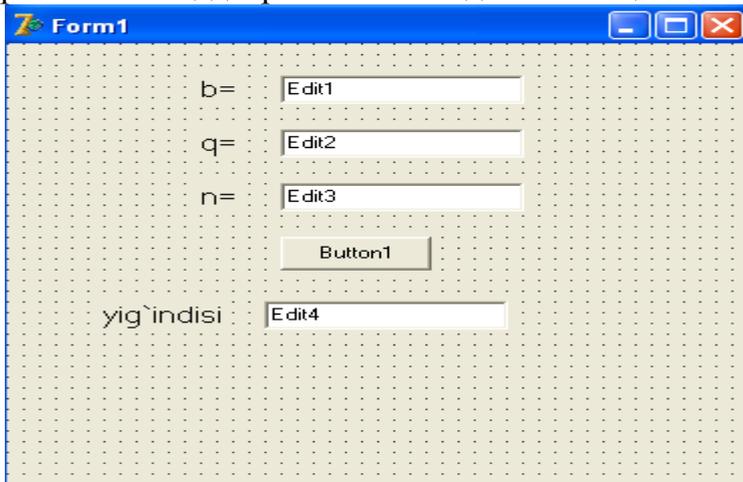
1.7-masala. Биринчи ҳади a_1 , айирмаси d , ҳадлари сони n га тенг арифметик прогрессиянинг ҳадларининг йиғиндисини ҳисоблаш дастурини тузинг.

Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var a1,d,n,S: real;
begin
a1:=StrToFloat(Edit1.Text);
d:=StrToFloat(Edit2.Text);
n:=StrToFloat(Edit3.Text);
S:=(2*a1+d*(n-1))/2*n;
Edit4.Text:=FloatToStr(S);
end;
```

end.

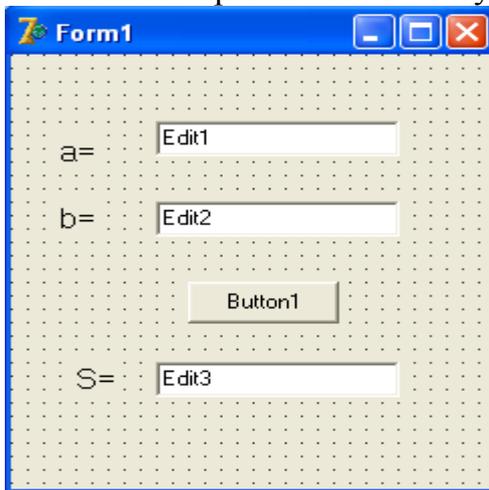
1.8-масала. Биринчи ҳади b , махражи q ва ҳадлари сони n га тенг геометрик прогрессиянинг ҳадларининг йиғиндисини S ни ҳисоблаш дастурини тузинг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var B,N,Q,S:real;
begin
B:=StrToFloat(Edit1.Text);
Q:=StrToFloat(Edit2.Text);
N:=StrToFloat(Edit3.Text);
S:=(B*(1-exp(N*ln(Q))))/(1-Q);
Edit4.Text:=FloatToStr(S);
end;
end.
```

1.9-масала. Берилган соннинг бутун қисмини аниқланг.

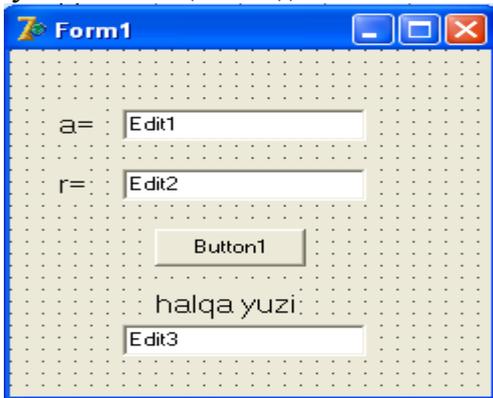


Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var a,b,s:real;
begin
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
s:=a div b;
Edit3.Text:=FloatToStr(s);
end;
```

end.

1.10-масала. Радиуслари A ва R га тенг ($A < R$) ҳалқа юзи ҳисоблансин. (Ҳисоблаш формуласи: $S = \pi(R^2 - A^2)$).



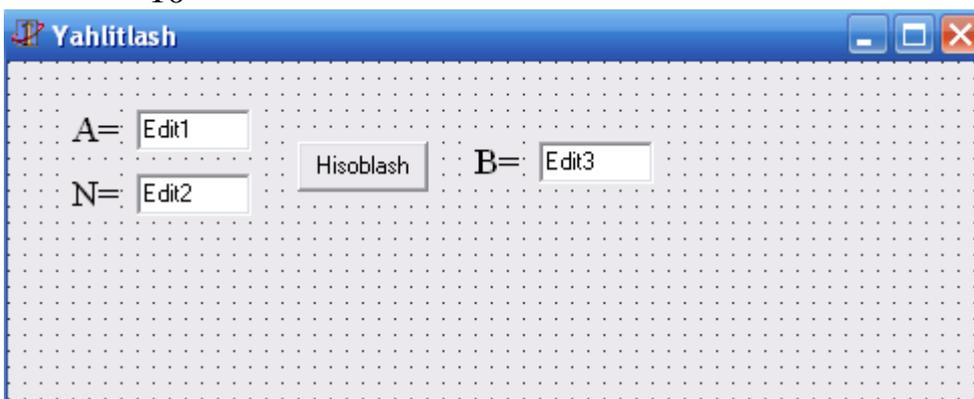
Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var a,r,s:real;
begin
a:=StrToFloat(Edit1.Text);
r:=StrToFloat(Edit2.Text);
if a<r then s:=pi*(r*r-a*a) else s:=pi*(a*a-r*r);
Edit3.Text:=FloatToStr(S);
end;
```

1.11-масала. Сонни берилган анқликда яхлитланг.
Ечиш.

А сонни N та ўнли хоналар аниқлигида яхлитлаш учун

$$B = \frac{INT(A \cdot 10^N + 0.5)}{10^N}$$
; формуладан фойдаланамиз.



```
procedure TForm1.Button1Click(Sender: TObject);
var A,N,B:real;
begin
A:=StrToFloat(Edit1.Text);
N:=StrToFloat(Edit2.Text);
B:=INT(A*EXP(N*LN(10))+0.5)/EXP(N*LN(10));
Edit3.Text:=FloatToStr(B);
```

end;

Тармоқланувчи алгоритмларга доир дастурлар тузиш.

2.1-масала. Тармоқланувчи дастурларга доир Delphi да квадрат тенгламанинг ҳақиқий илдизларини топиш дастури тузилган. Бу дастурга мос процедураларнинг кўриниши қуйидагича бўлади:

```
procedure TForm1.Label4Click(Sender: TObject);
var d,a,b,c,x1,x2:real;
begin
a:=strtofloat(edit1.Text);
b:=strtofloat(edit2.Text);
c:=strtofloat(edit3.Text);
d:=b*b-4*a*c;
if d<0 then Label4.Caption:='хақиқий ечими йук'
else begin x1:=(-b-sqrt(d))/(2*a);
x2:=(-b+sqrt(d))/(2*a);
Label4.Caption:='x1= '+floattostr(x1)+' x2= '+floattostr(x2);
end;
end;
procedure TForm1.Button1Click(Sender: TObject);
var d,a,b,c,x1,x2:real;
begin
a:=strtofloat(edit1.Text);
b:=strtofloat(edit2.Text);
c:=strtofloat(edit3.Text);
d:=b*b-4*a*c;
if d<0 then Label4.Caption:='хақиқий ечими йук'
else begin x1:=(-b-sqrt(d))/(2*a);
x2:=(-b+sqrt(d))/(2*a);
Label4.Caption:='x1= '+floattostr(x1)+' x2= '+floattostr(x2);
end;
end;
```

Бу дастурга мос келувчи форманинг кўриниши қуйидагича:

Дастур бажарилгандан кейинги натижанинг кўриниши эса қуйидагича бўлади:

2.

2.2-масала. Икки бутун мусбат сон m ва n ларнинг энг катта умумий бўлувчиси (EKUB) ни аниқлаш дастурини тузинг.

Дастур матни

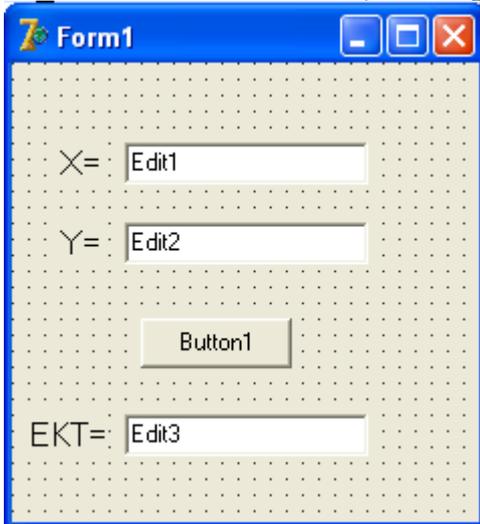
```
procedure TForm1.Button1Click(Sender: TObject);  
label 1,2;  
var m,n,x,y:ineteget;
```

```

begin
m:=StrToInt(Edit1.Text);
n:=StrToInt(Edit2.Text);
x:=m;y:=n;
1: if x=y then goto 2;
if x>y then x:=x-y else y:=y-x; goto 1;
2: Edit3.Text:=IntToStr(x);
end;

```

2.3-масала. Иккита x ва y сонларнинг каттасини танлаш (EKT) дастурини тузинг.



Дастур матни

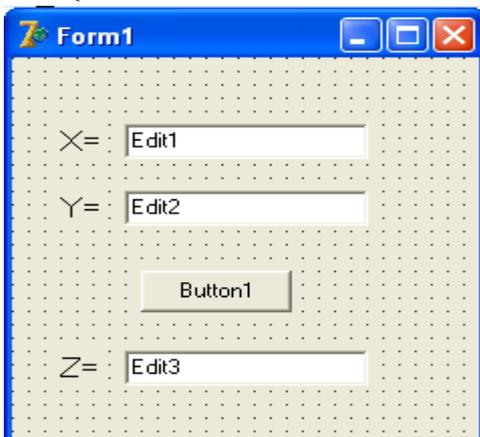
```

procedure TForm1.Button1Click(Sender: TObject);
var m,y,x:integer;
begin
x:=StrToInt(Edit1.Text);
y:=StrToInt(Edit2.Text);
if x>y then begin m:=x; Edit3.Text:=FloatToStr(m); end
else if y>x then begin m:=y; Edit3.Text:=FloatToStr(m); end
else showmessage('Bu sonlar teng!');
end;

```

2.4-масала. X ва y ҳақиқий сонлар берилган. z функцияни ҳисоблаш дастурини тузинг:

$$Z = \begin{cases} X - Y, & \text{агар } X > Y \\ X + 1, & \text{агга } X \leq Y \end{cases}$$



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var z,y,x:integer;
begin
x:=StrToFloat(Edit1.Text);
y:=StrToFloat(Edit2.Text);
if x>y then z:=x-y;
else z:=x+1; Edit3.Text:=FloatToStr(z);
end;

```

2.5-масала. Ракета v (км/соат) тезлик билан Ер экваторидаги нуқтадан Ернинг қуёш атрофидаги орбитаси бўйлаб учирилади. Ракетани учириш натижаси қандай бўлади?



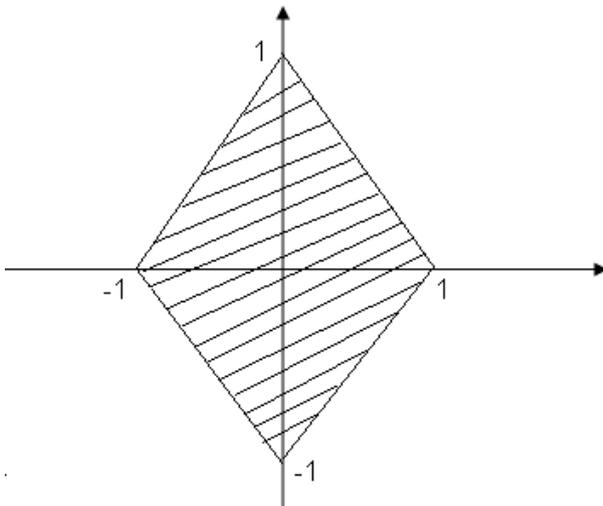
Дастур матни

```

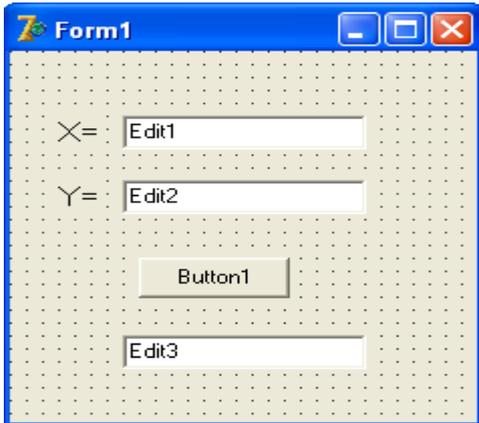
procedure TForm1.Button1Click(Sender: TObject);
var a:real;
begin
a:=strtofloat(edit1.Text);
if a<7.9 then label3.Caption:='Raketa Yerga qaytib tushadi';
if (a>=7.9) and (a<11.2) then label3.Caption:='Raketa Yerning sun`iy yo`ldoshiga aylanadi';
if (a>=11.2) and (a<16.7) then label3.Caption:='Quyoshning sun`iy yo`ldoshiga aylanadi';
if a>=16.7 then label3.Caption:='Raketa Galaktikaning sun`iy yo`ldoshiga aylanadi';
end;
end.

```

2.6-масала. Координаталари x ва y га тенг бўлган нуқта 2.1-расмда тасвирланган текисликдаги шаклга тегишлими?



2.1-rasm



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var y,x:real;
begin
x:=StrToFloat(Edit1.Text);
y:=StrToFloat(Edit2.Text);
if (x>-1) and (x<1) and (y>-1) and (y<1) then
Edit3.Text:='Tegishli!' else Edit3.text:='Tegishli emas!';
end;

```

2.7. Оптимал оғирликни аниқлаш дастурини ёзинг. Дастур формаси қуйидагича кўринишга эга.



Дастур матни

```

procedure tform1.button1click(sender: tobject);
var w,h,opt,d:real;
begin
w:=strtofloat(edit1.text);
h:=strtofloat(edit2.text);
opt:=h-100;
if w=opt then
begin
label3.caption:='optimal';
end
else
if w<opt then
begin d:=opt-w;

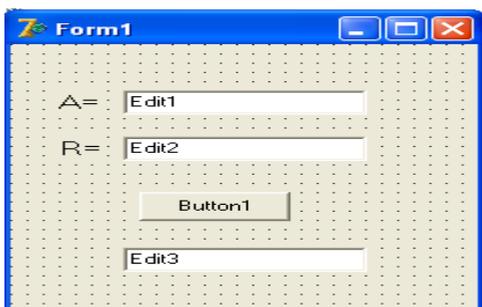
```

```

label3.caption:='siz semirishingiz kerak';
end
else
begin
d:=w-opt;
label3.caption:='siz ozishingiz kerak'+floattostr(d);
end;
end;
end.

```

2.8-масала. Агар квадратнинг томони A , доиранинг радиуси r га тенг бўлса, квадрат ва доиранинг юзларини солиштириб каттасини аниқланг.



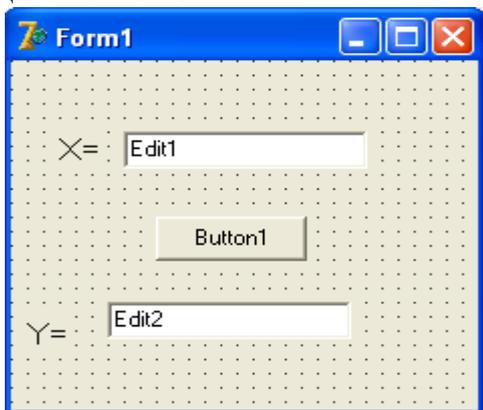
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,r,sk,sd:real;
begin
a:=StrToFloat(Edit1.Text);
r:=StrToFloat(Edit2.Text);
sk:=a*a;sd:=pi*r*r;
if sk>sd then Edit3.Text:='Kvadratning yuzi katta!' else if sk<sd then Edit3.Text:='Doiraning yuzi katta!' else Edit3.Text:='Yuzlari teng!';
end;

```

2.9-масала. Қуйидаги функция ҳисоблансин: $x > 0$ бўлганда 1 га тенг; $x = 0$ да нолга тенг; $x < 0$ да -1 га тенг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var x,y:real;
begin
x:=StrToFloat(Edit1.Text);

```

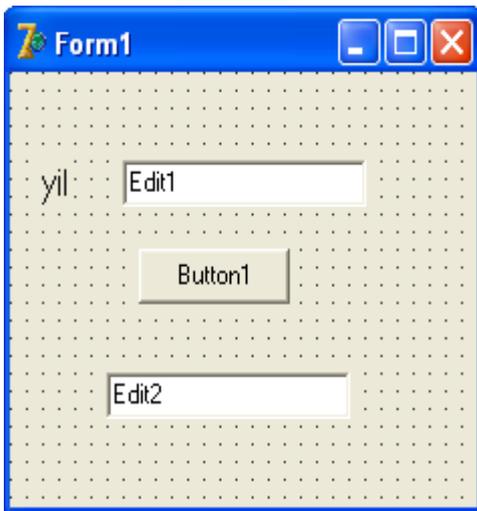
```

if x>0 then y:=1 else if x=0 then y:=0 else y:=-1;
Edit2.Text:=FloatToStr(y);
end;

```

2.10-масала. Берилган N йил кабиса йили бўлиш-бўлмаслигини аниқланг.

Изоҳ. Агар N 100 га қаррали сон бўлмаса ва унинг охири икки рақами 4 га қаррали сон бўлса, у ҳолда N -йил кабиса йилидир. Агар N сони 100 қаррали бўлса, у ҳолда N сони 400 га қаррали бўлгандагина мазкур йил кабиса йили бўлади.



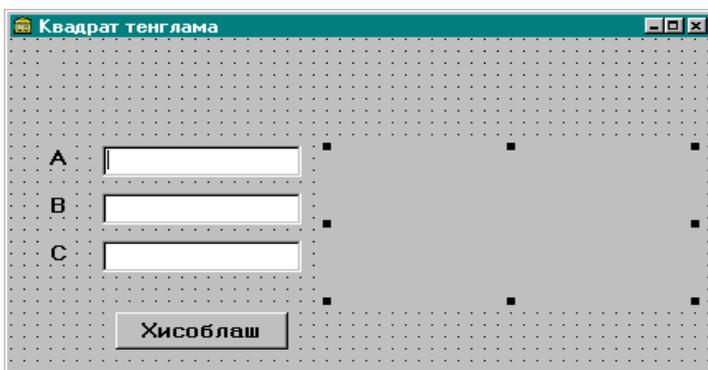
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var n:integer;
begin
n:=StrToInt(Edit1.Text);
if n mod 100=0 then begin if n mod 400 =0 then Edit2.Text:='Bu yil kabisa yili!' else Edit2.Text:='Bu yil kabisa yili emas!'; end else if n mod 4 =0 then Edit2.Text:='Bu yil kabisa yili!'else Edit2.Text:='Bu yil kabisa yili emas!';
end;

```

2.10. Квадрат тенгламининг илдизини топиш дастурини тузинг.



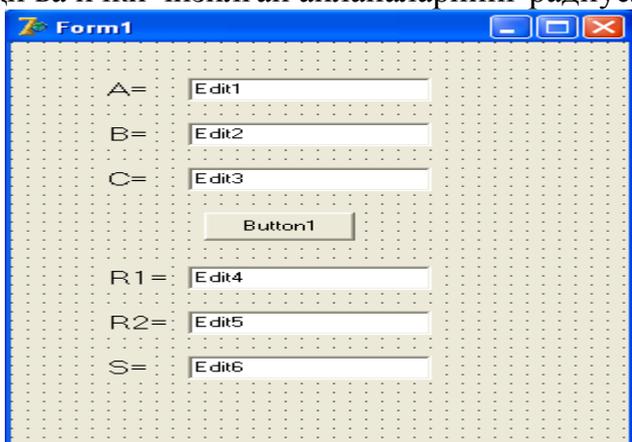
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
Var
a,b,c:Real; { Тенглама коэффициентлари }
d:Real; { Дискриминант }
x1,x2:Real; { Тенглама илдизлари }
begin
{ Керакли маълумотларни киритиш }
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
c:=StrToFloat(Edit3.Text);
d:=b*b-4*a*c; { Дискриминантни ҳисоблаш }
If d<0 Then
Begin
Label5.Caption:='Дискриминант нолдан кичик'+#13+ 'Тенгламанинг илдизи йук.'
End
Else
Begin
x1:=(-b-Sqrt(d))/(2*a); { Илдизларни ҳисоблаш }
x2:=(-b+Sqrt(d))/(2*a); { x1, x2 натижани чоп этиш }
Label5.Caption:='Тенглама илдизлари' +#13+'x1='+FloatToStr(x1)+#13+'x2='+FloatToStr(x2);
End;
end;
procedure TForm1.FormActivate(Sender: TObject);
begin Label1.Caption:='Тенглама коэффициентини киритинг' +#13+'ва Ҳисоблаш тугмасини
босинг';
end;
end.

```

2.11-масала. a , b , c сонлар мос равишда учта кесманинг узунликларини ифодалайди. Агар кесмалар учбурчакнинг томонларини ифодаласа, учбурчакнинг юзи s , учбурчакка ташқи ва ички чизилган айланаларнинг радиуслари r_1 ва r_2 ларни топинг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b,c,r1,r2,s,p:real;
begin
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
c:=StrToFloat(Edit3.Text);
if ((a+b)>c) and ((a+c)>b) and ((b+c)>a) then begin

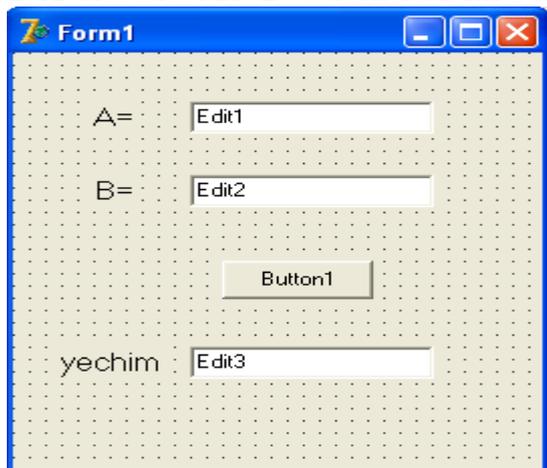
```

```

p:=(a+b+c)/2;
s:=sqrt(p*(p-a)*(p-b)*(p-c));
r2:=s/p;
r1:=a*b*c/(4*s);
Edit4.Text:=FloatToStr(r1);
Edit5.Text:=FloatToStr(r2);
Edit6.Text:=FloatToStr(s); end else showmessage('Kiritilgan sonlar uchburchak
tomonlarini ifodalamaydi!');
end;

```

2.12-масала. $Ax+B=0$ тенгламани ечинг.



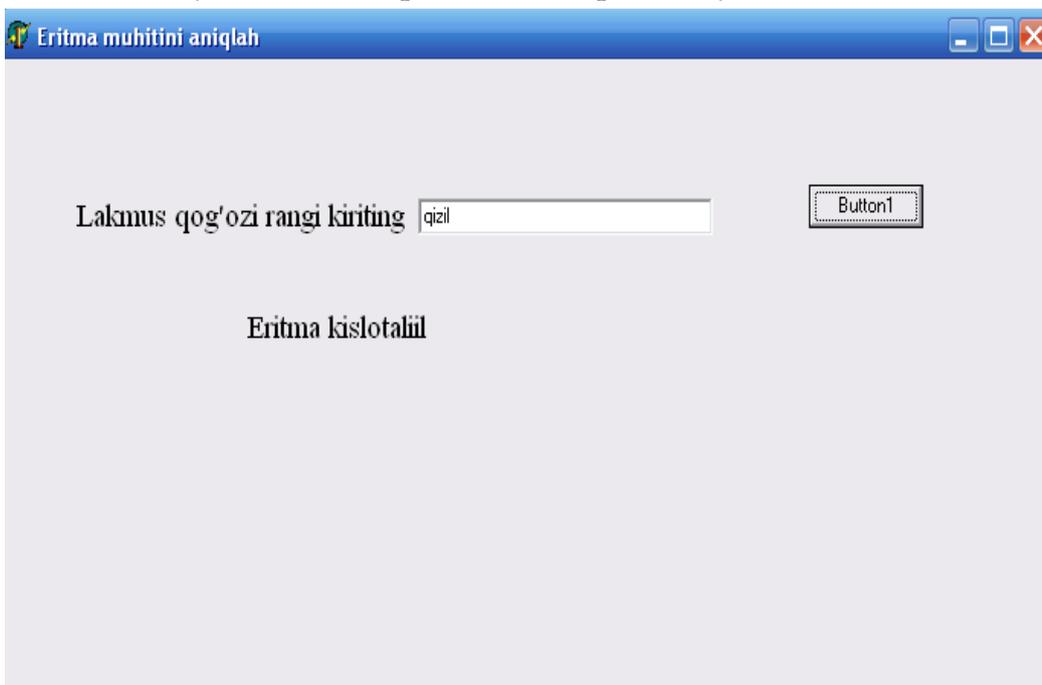
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b,x:real;
begin
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
if a=0 then if b<>0 then showmessage('Bu tenglamaning yechimi yo`q!')
else Edit3.Text:='Bu tenglamaning yechimlari cheksiz ko`p!' else begin x:=-b/a; Ed-
it3.Text:=FloatToStr(x);end;
end;

```

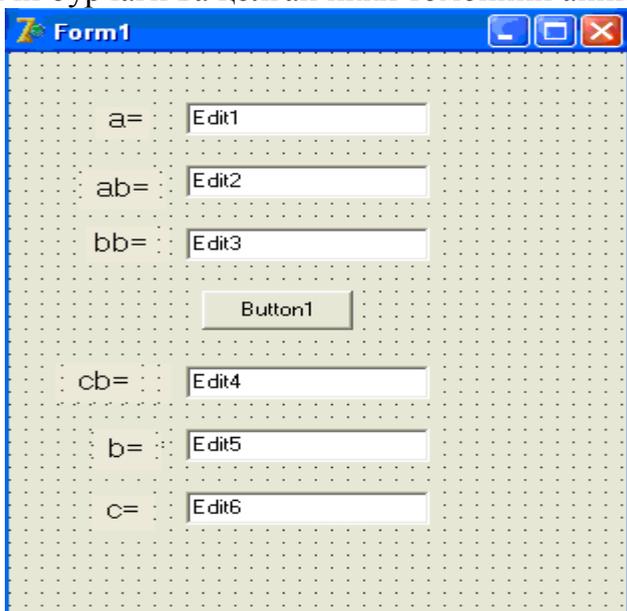
2.13-масала. Лакмус қоғозидан фойдаланиб эритма муҳитини аниқлаш дастурини тузинг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var a:string[10];
begin
a:=Edit1.text;
if a='qizil' then
Label3.Caption:='Eritma kislotaliil' else
if a='ko`k' then
Label3.Caption:='Eritma ishqorli'
else
Label3.Caption:='Eritma neytral';
end;
end.
```

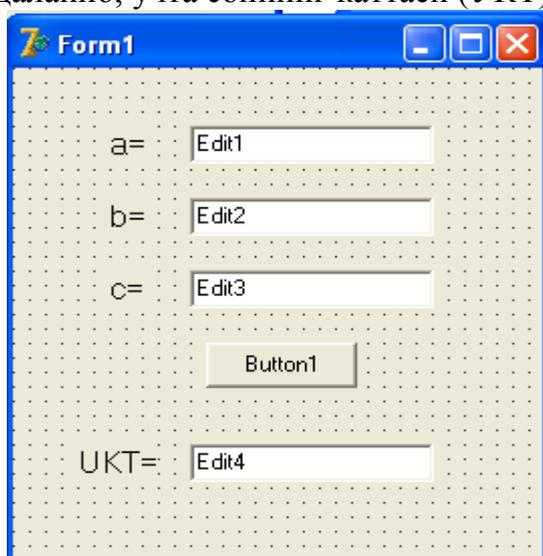
2.14-масала. Бир томони ва унга ёпишган иккита бурчаги берилган учбурчакнинг учинчи бурчаги ва қолган икки томонини аниқланг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var a,ab,bb,b,c,cb:real;
begin
a:=StrToFloat(Edit1.Text);
ab:=StrToFloat(Edit2.Text);
bb:=StrToFloat(Edit3.Text);
cb:=pi-ab*pi/180-bb*pi/180;
b:=a*sin(bb)/sin(ab);
c:=a*sin(cb)/sin(ab);
Edit4.Text:=FloatToStr(cb);
Edit5.Text:=FloatToStr(b);
Edit6.Text:=FloatToStr(c);
end;
```

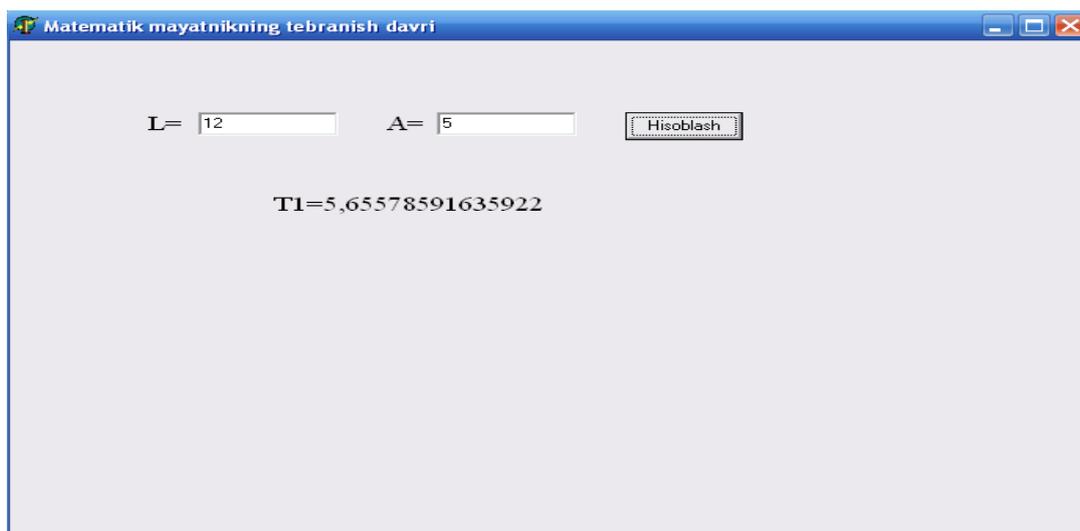
2.15-масала. Иккита соннинг каттаси (ИКТ) ни топиш ёрдамчи алгоритмидан фойдаланиб, учта соннинг каттаси (УКТ) ни топинг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c,k:real;
function ИКТ(x,y:real):real;
begin if x>y then ИКТ:=x else ИКТ:=y;
end;
begin
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
c:=StrToFloat(Edit3.Text);
if ИКТ(a,b)=a then k:=ИКТ(a,c) else k:=ИКТ(b,c);
Edit4.Text:=FloatToStr(k);
end;
```

2.16. Узунлиги l га тенг математик маятникнинг осилган нуқтаси қўзгалмас ёки юқорига ё пастга тезланиш билан ҳаракатланган ҳолларда унинг тебраниш даври аниқлансин.



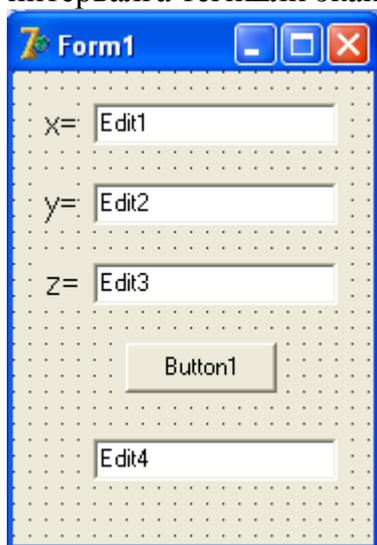
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
const G=9.81;
var L,A:real;
begin
L:=StrToFloat(Edit1.Text);
A:=StrToFloat(Edit2.Text);
if A=0 then Label3.Caption:='T='+FloatToStr(2*pi*sqrt(L/G))
else if A=G then Label3.Caption:='mayatnik vazinsiz holatda bo`ladi'
else if A<G then Label3.Caption:='T1='+FloatToStr(2*pi*sqrt(L/(G+A)))
else Label3.Caption:='T2='+FloatToStr(2*pi*sqrt(L/(A-G)));
end;
end.

```

2.17-масала. Учта x , y , z ҳақиқий сонлар берилган. Бу сонлардан қайси бири $[1,5]$ интервалга тегишли эканлигини аниқланг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var x,y,z:real;

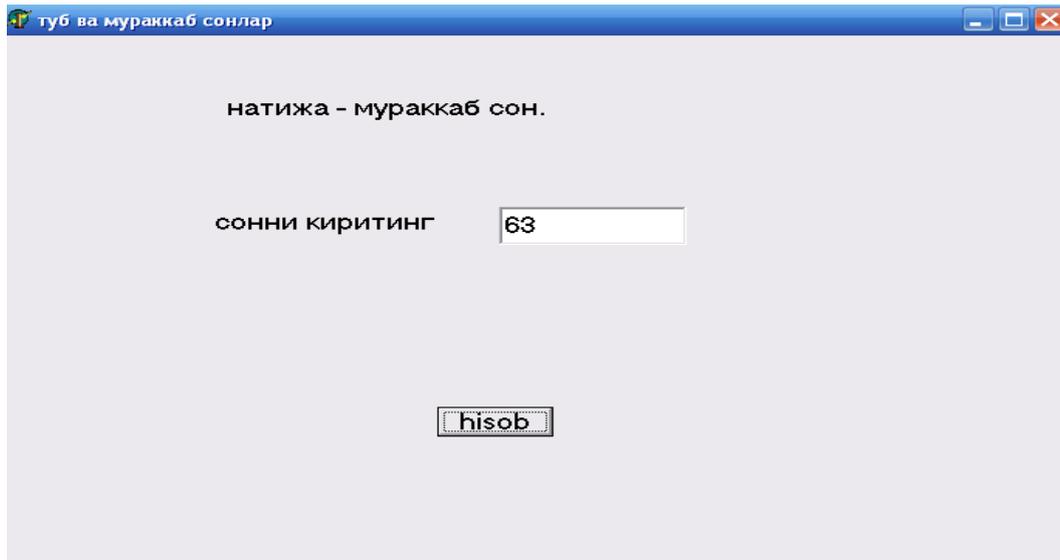
```

```

begin
x:=StrToFloat(Edit1.Text);
y:=StrToFloat(Edit2.Text);
z:=StrToFloat(Edit3.Text);
if (x>1) and (x<5) then Edit4.Text:= Edit4.Text +'x,';
if (y>1) and (y<5) then Edit4.Text:= Edit4.Text +'y,';
if (z>1) and (z<5) then Edit4.Text:= Edit4.Text +'z';
end;

```

2.18. Клавиатура орқали киритилган сон туб сон ёки мураккаб сон эканлигини аникловчи дастур тузинг.



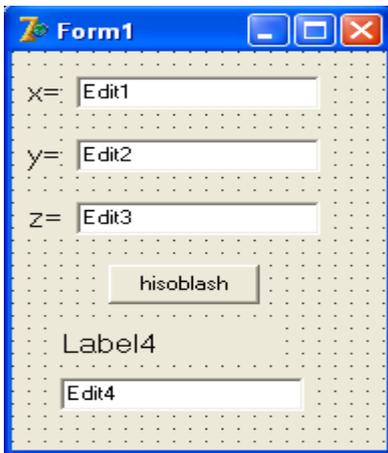
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var
n: integer; // текширилаётган сон
d: integer; // булувчи
r: integer; // n ни d га бўлишдан колган колдик
begin
n:=StrToInt(Edit1.text);
d := 2; // дастлаб иккига бўламыз
repeat
r := n mod d;
if r <> 0 // n сони d га колдиксиз булинмади
then d := d + 1;
until r = 0; // n га колдиксиз булинадиган сонни топгунча давом этиш
label1.caption:=Edit1.text;
if d = n
then label1.caption:=label1.caption + ' - туб сон.'
else label1.caption:=label1.caption + ' - мураккаб сон.';
end;

```

2.19-масала. Учта x , y , z мусбат сонлар берилган. Томонлари x, y, z га тенг учбурчак мавжудми? Агар мавжуд бўлса бу учбурчакнинг юзини топинг.



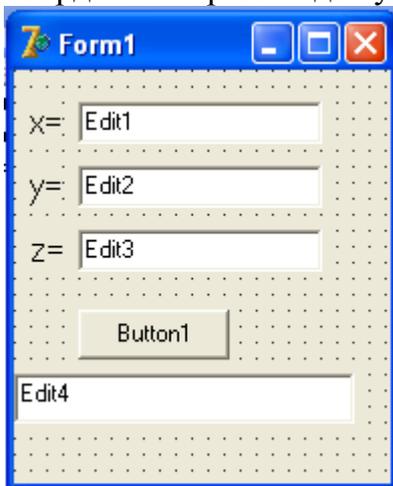
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var x,y,z,s,p:real;
begin
x:=StrToFloat(Edit1.Text);
y:=StrToFloat(Edit2.Text);
z:=StrToFloat(Edit3.Text);
if (x+y>z) and (x+z>y) and (z+y>x) then begin label4.caption:='Bunday uchburchak mavjud!';
p:=(x+y+z)/2;
s:=sqrt(p*(p-x)(p-y)(p-z));
Edit4.Text:=FloatToStr(s);end else label4.caption:='Bunday uchburchak mavjud emas!';
end;

```

2.20-масала. Координаталари берилган $M(X,Y)$ нуқтанинг радиуси R га тенг ва маркази координаталар бошида бўлган доирага тегишли бўлишини аниқланг.



Дастур матни

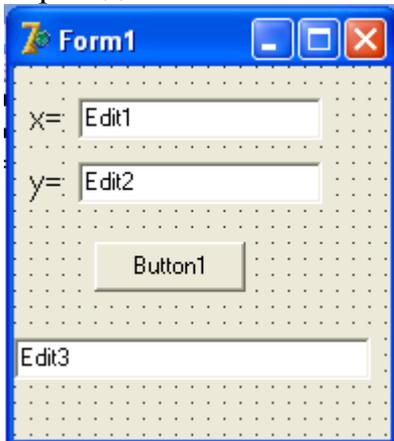
```

procedure TForm1.Button1Click(Sender: TObject);
var x,y,r:real;
begin
x:=StrToFloat(Edit1.Text);
y:=StrToFloat(Edit2.Text);
r:=StrToFloat(Edit3.Text);
if x*x+y*y>r*r then

```

```
Edit4.Text:='M('+ FloatToStr(x)+','+ FloatToStr(y)'+')'+
nuqta radiusi '+FloatToStr(r)+
'bo`lgan doiraga tegishli emas!'else Edit4.Text:='M('+ FloatToStr(x)+','+ FloatToStr(y)'+')'+
nuqta radiusi '+FloatToStr(r)+'bo`lgan doiraga tegishli !';
end;
```

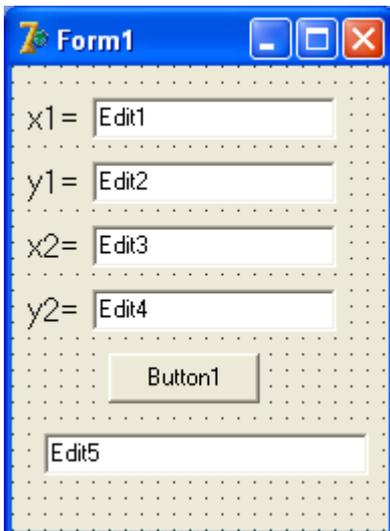
2.21-масала. Координаталари берилган $m(x,y)$ нукта координата текислигининг қайси чорагида эканлигини аниқлайдиган дастур тузинг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var x,y:real;
k:integer;
begin
x:=StrToFloat(Edit1.Text);
y:=StrToFloat(Edit2.Text);
if (x<0) and (y<0) then k:=3;
if ((x<0) or (x>0)) and (y=0) then k:=0;
if (x<0) and (y>0) then k:=2;
if (x>0) and (y<0) then k:=4;
if (x>0) and (y>0) then k:=1;
if (x=0) and ((y<0) or (y>0)) then k:=5;
if k=0 then begin Edit3.Text:='Ushbu nuqta OX o`qiga tegishli!' end
else if k=5 then begin Edit3.Text:='Ushbu nuqta OY o`qiga tegishli!' end
else Edit3.Text:='Ushbu nuqta '+FloatToStr(k)+' -chorakka tegishli!';
end;
```

2.22-масала. Координаталари берилган $m_1(x_1,y_1)$ ва $m_2(x_2,y_2)$ нукталарнинг қайси бири координата бошига яқин туради?

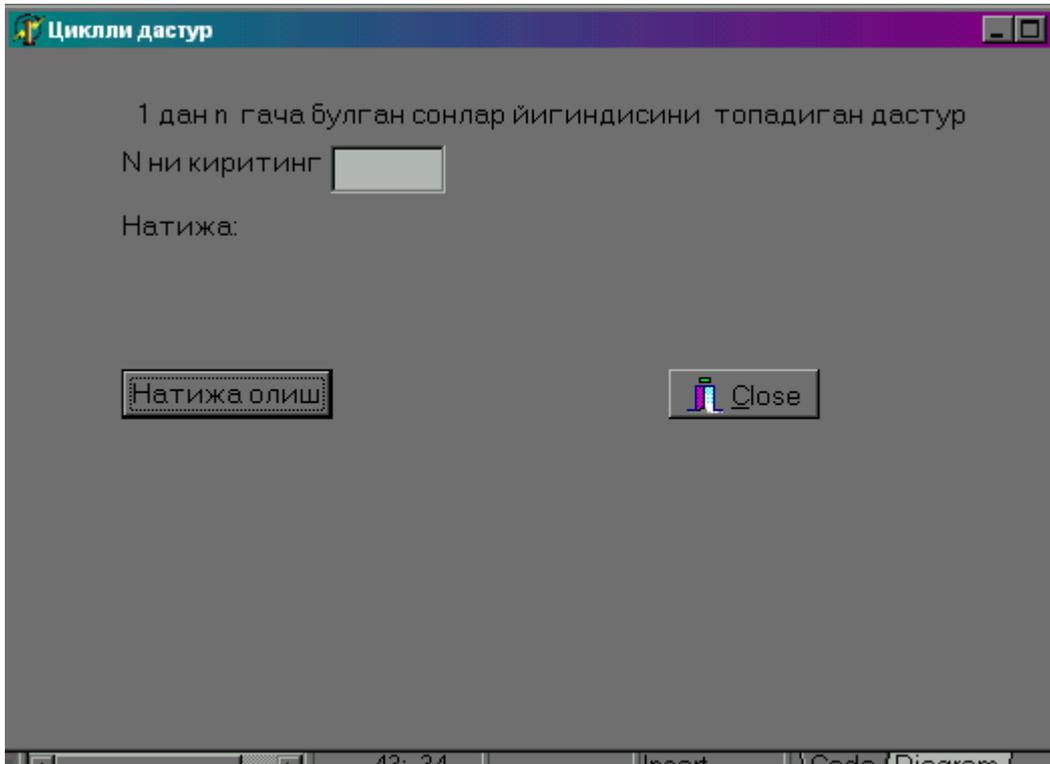


Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var x1,y1,x2,y2,r1,r2:real;
begin
x1:=StrToFloat(Edit1.Text);
y1:=StrToFloat(Edit2.Text);
x2:=StrToFloat(Edit3.Text);
y2:=StrToFloat(Edit4.Text);
r1:=sqrt(x1*x1+y1*y1);
r2:=sqrt(x2*x2+y2*y2);
if r1>r2 then Edit5.Text:='M2 nuqta yaqin turadi!' else if r1<r2 then Edit5.Text:='M1 nuqta
yaqin turadi!' else Edit5.Text:='Ikkala nuqta bir xil uzoqlikda turadi!';
end;
```

3. Такрорланувчи алгоритмларга доир дастурлар тузиш.

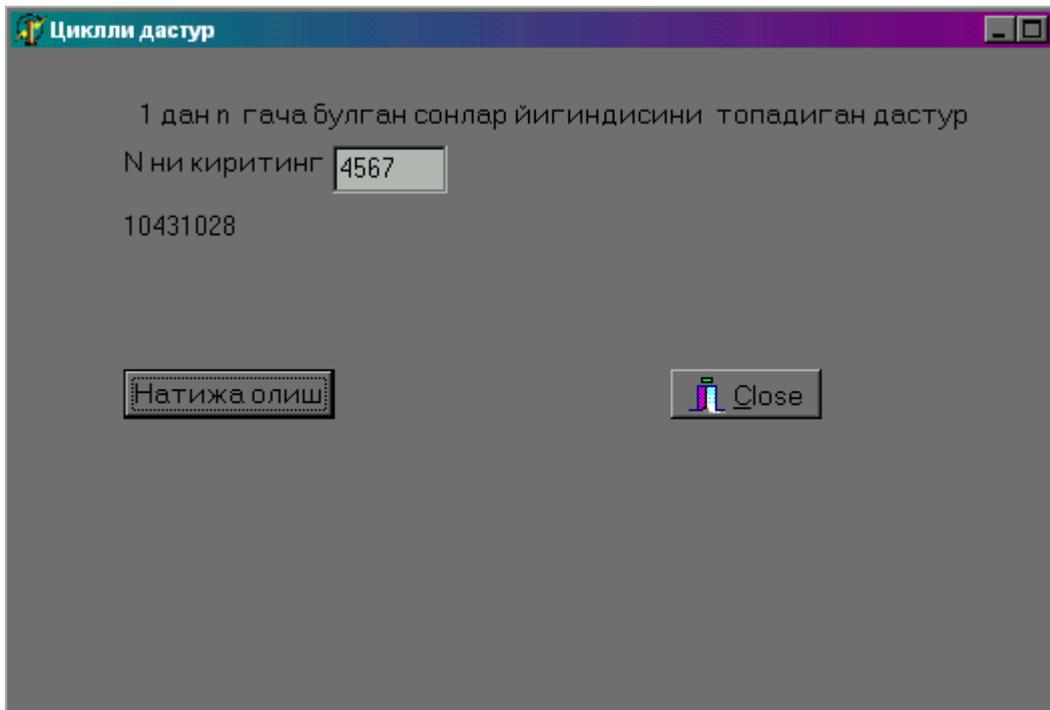
3.1-масала. Такрорланувчи дастурга тузилган форманинг кўриниши қуйидагича:



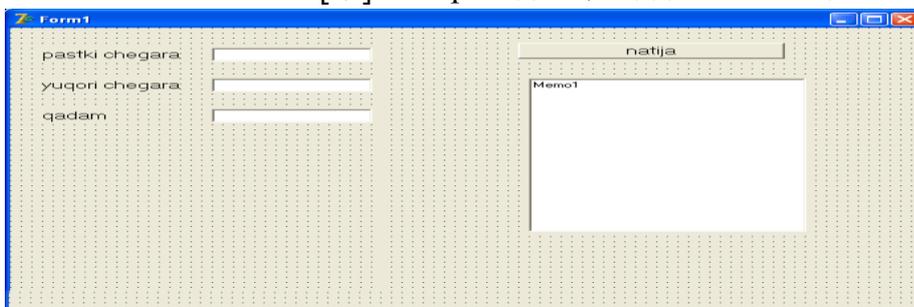
Такрорланувчи дастурга Delphiда тузилган программадаги мос процедураларнинг кўриниши куйидагича:

```
procedure TForm1.Label2Click(Sender: TObject);  
var i,s,a:integer;  
begin  a:=strToInt(edit1.text);  
s:=0;  
for i:=1 to a do s:=s+i;  
label2.Caption:=floattostr(s);  
end;  
procedure TForm1.Button1Click(Sender: TObject);  
var i,s,a:integer;  
begin  a:=strToInt(edit1.text);  
s:=0; for i:=1 to a do s:=s+i;  
label2.Caption:=floattostr(s);  
end;
```

Дастур бажарилгандан кейин экранда куйидаги шаклда натижа ҳосил бўлади:



3.1- масала. $Y=X^2$ ни $[0,1]$ интервалда 0,1 қадам билан қийматлар жадвалини аниқланг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b:byte;
n,y,x:real;s1,s2:string;
begin
a:=strtoint(edit1.Text);
b:=strtoint(edit2.Text);
n:=strtofloat(edit3.Text); x:=a;
memo1.Clear;
repeat y:=sqr(x);
str(x:2,s1);str(y:5,s2);
memo1.Lines.Add('x='+s1+' y='+s2);
x:=x+n;
until x>=b;
end; end.

```

3.2. Мисол: $S = \sum_{i=1}^n \frac{1}{i}$ йи\индининг n та ҳадини топиш дастурини тузиш.

Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);

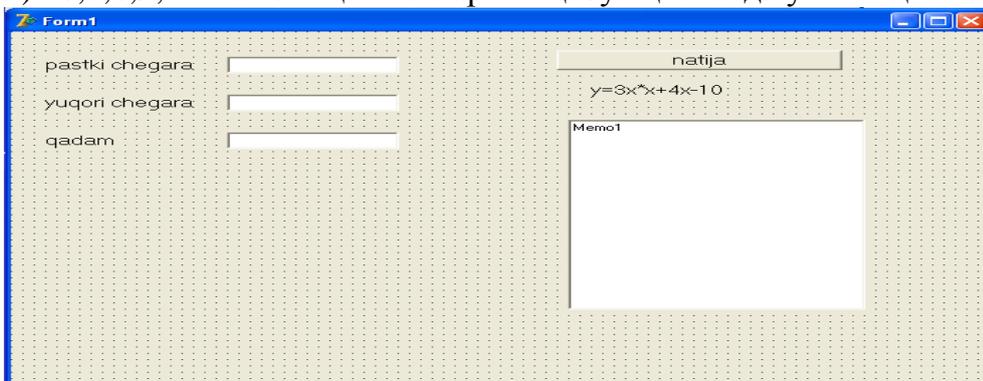
```

```

Var
S:String;
i,n:Integer;
Summ:Real;
begin
S:=InputBox('Киритиш ойнаси','N ни киритинг,');
N:=StrToInt(S);
Summ:=0;
For i:=1 to n do Summ:=Summ+(1/i);
Label2.Caption:='Сумма='+FloatToStr(Summ);
end;
end.

```

3.3-масала. $Y=3x^2 + 4x - 10$ функциянинг x ўзгарувчи 1). 0,1,2,3,4,5;
 2). 0,3,6,9,12 га тенг қийматларини қабул қилганда у нинг қийматини ҳисобланг.



Дастур матни

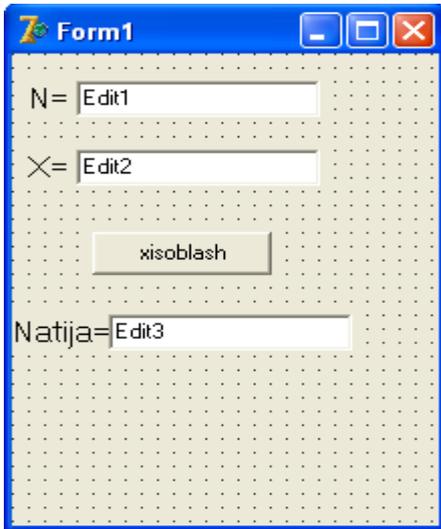
```

procedure TForm1.Button1Click(Sender: TObject);
var a,b:byte;
n,y,x:real;s1,s2:string;
begin
a:=strtoint(edit1.Text);
b:=strtoint(edit2.Text);
n:=strtofloat(edit3.Text);
x:=a; memo1.Clear;
repeat
y:=3*sqr(x)+4*x-10; str(x:1,s1);str(y:5,s2);
memo1.lines.Add('x='+s1+' y='+s2);
x:=x+n;
until x>b;
end; end.

```

3.3- масала. N натурал сон ва x ҳақиқий сонлар берилган. Қуйидаги йиғиндини ҳисобланг.

$$\sin X + \sin^2 X + \dots + \sin^N X .$$



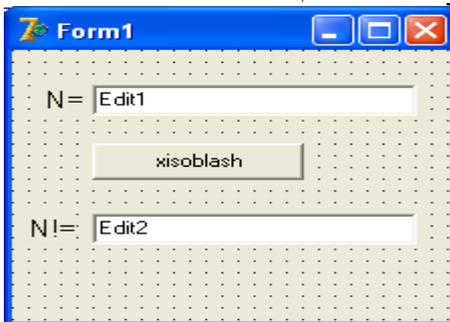
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var n,i:byte; x,s:real;
begin
n:=strtoint(edit1.text);
x:=strtofloat(edit2.text);
s:=0; x:=sin(x);
for i:=1 to n do
s:=s+exp(i*ln(x));
edit3.text:=floattostr(s);
end; end.

```

3.4- масала. $n!$ аниқлансин. Бунда n натурал сон.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer; f:int64;
begin
n:=strtoint(edit1.Text);
f:=1;
for i:=1 to n do
f:=f*i;
edit2.Text:=inttostr(f);
end; end.

```

3.5- масала. 1дан 20 гача натурал сонлар квадратлари йиғиндисини топинг.



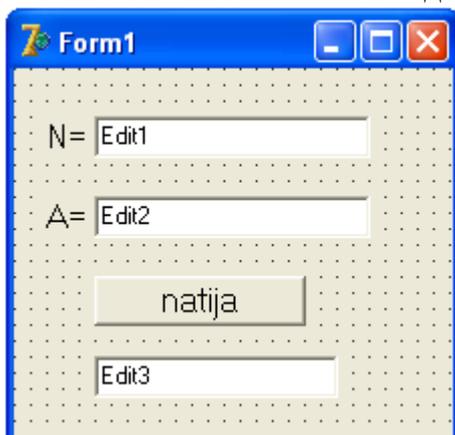
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer; f:int64;
begin
n:=strtoint(edit1.Text);
f:=0;
for i:=1 to n do
f:=f+sqr(i);
edit2.Text:=inttostr(f);
end; end.

```

3.6- масала. А сонининг N - даражасини такрорлаш буйруғи ёрдамида ҳисобланг.



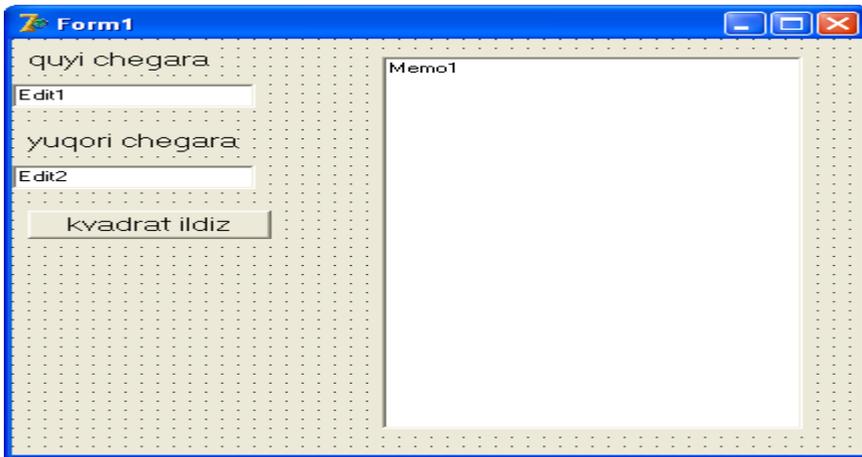
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var n,i,a,f:integer;
begin
n:=strtoint(edit1.Text);
a:=strtoint(edit2.Text);
f:=1;
for i:=1 to n do f:=f*a;
edit3.Text:=inttostr(f);
end; end.

```

3.7-масала. 1 дан 10 гача бўлган сонлардан цикл қадами 1 га тенг ҳолда квадрат илдиз чиқариш дастурини тузинг.



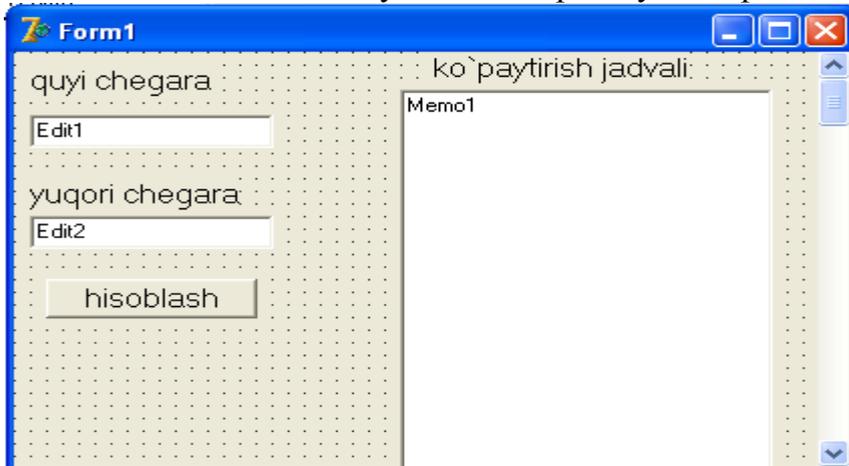
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b,i:byte;
s:real; s1,s2:string;
begin
a:=strtoint(edit1.Text);
b:=strtoint(edit2.Text);
memo1.Clear;
for i:=a to b do begin
s:=sqrt(i);
str(i,s2); str(s,s1);
memo1.Lines.Add('x='+s2+' ildiz='+s1);
end;
end; end.

```

3.8-масала. 1 дан 9 гача бўлган сонларни кўпайтириш жадвалини экранга чиқаринг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b,i,j,s:integer;s1,s2,s3:string;
begin
a:=strtoint(edit1.Text);
b:=strtoint(edit2.Text); memo1.Clear;
for i:=a to b do
for j:=1 to 10 do begin
s:=i*j;
str(s,s1);str(i,s2);str(j,s3);
memo1.Lines.Add(s2+'*'+s3+'='+s1);

```

end; end;
end.

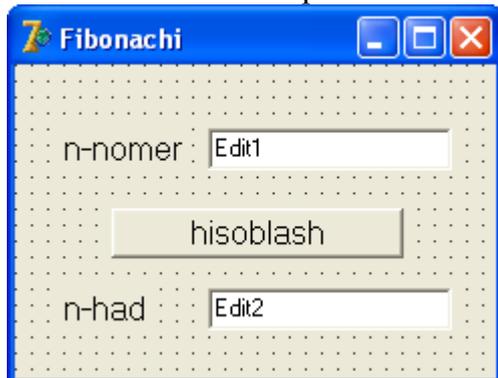
3.9-масала. Мураккаб циклга доир дастур тузиш. $s = \sum_{i=1}^{10} \prod_{j=1}^5 \frac{i+j}{\sqrt{i*j}}$;



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);  
Var I,j:integer; s,p:real;  
Begin  
s:=0;  
For i:=1 to 10 do  
Begin  
p:=1;  
For j:=1 to 5 do p:=p*(i+j)/sqrt(i*j);  
s:=s+p;  
End;  
Edit3.text:=floattostr(s);  
End;  
end.
```

3.10-масала. L номерли Фибоначчи сонини экранга чиқаринг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);  
var n,v,w,r,i:integer;  
begin  
n:=strtoint(edit1.Text);  
w:=0;v:=1;i:=1;  
while i<n do begin  
r:=v+w; w:=v; v:=r;  
i:=i+1;end;
```

```
edit2.Text:=inttostr(v);
end;
end.
```

3.11-masala. $y = e^x = 1 + x + \frac{x^2}{2!} + \dots + \frac{x^N}{N!} + \dots$

функциянинг қийматини 0,001 аниқликда ҳисобланг.

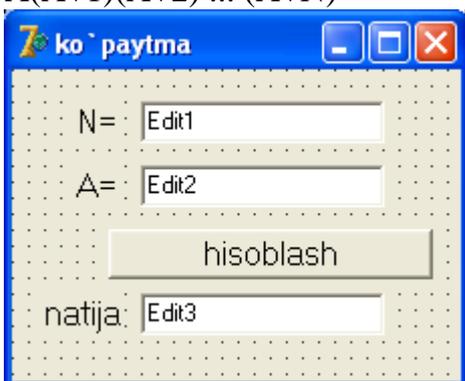


Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var x,y,p,s,eps:real; n:integer;
begin
eps:=strtofloat(edit1.Text);
x:=strtofloat(edit2.Text);
s:=1;p:=1;n:=1;
repeat
p:=p*x/n; s:=s+p;
n:=n+1;
until abs(p)<=eps;
edit3.Text:=floattostr(s);
end; end.
```

3.12-масала. N натурал сон ва A ҳақиқий сон берилган. Қуйидаги кўпайтмани ҳисобланг:

$$A(A+1)(A+2) \dots (A+N)$$



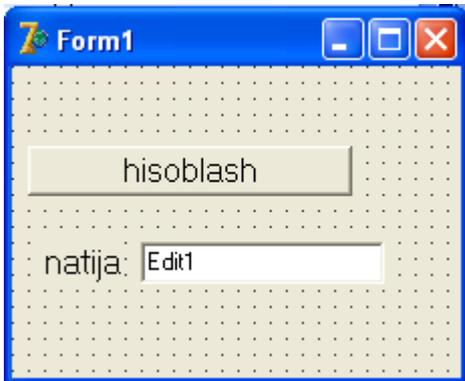
Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var a,p:real;i,n:integer;
begin
n:=strtoint(edit1.Text);
a:=strtofloat(edit2.Text);
p:=1;
for i:=0 to n do
```

```
p:=p*(a+i);
edit3.Text:=floattostr(p);
end; end.
```

3.13-масала. Даражага кўтариш амалини бажармай, ушбу йиғиндини ҳисобланг:

$$S = \sum_{n=1}^{10} (-1)^n n^2$$



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var s:real; n:integer;
begin
s:=0;
for n:=1 to 10 do
if odd(n)=true then s:=s-sqr(n) else s:=s+sqr(n);
edit1.Text:=floattostr(s);
end; end.
```

3.14-масала. Натурал сонни туб кўпайтувчиларга ажратинг.

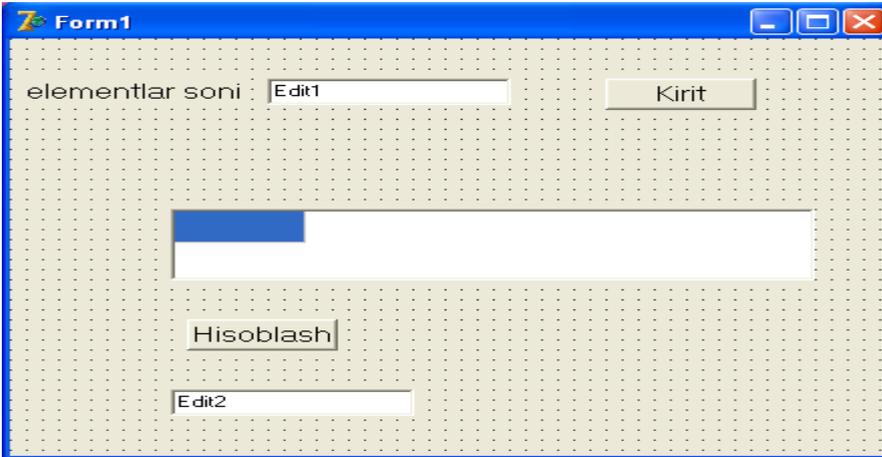


Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer; s:string;
begin
n:=strtoint(edit1.Text);
memo1.Clear;
for i:=1 to n do
if n mod i=0 then begin
str(i,s);
memo1.Lines.Add(s);
end;end;end.
```

4. Массив ва матрицаларга доир дастурлар тузиш.

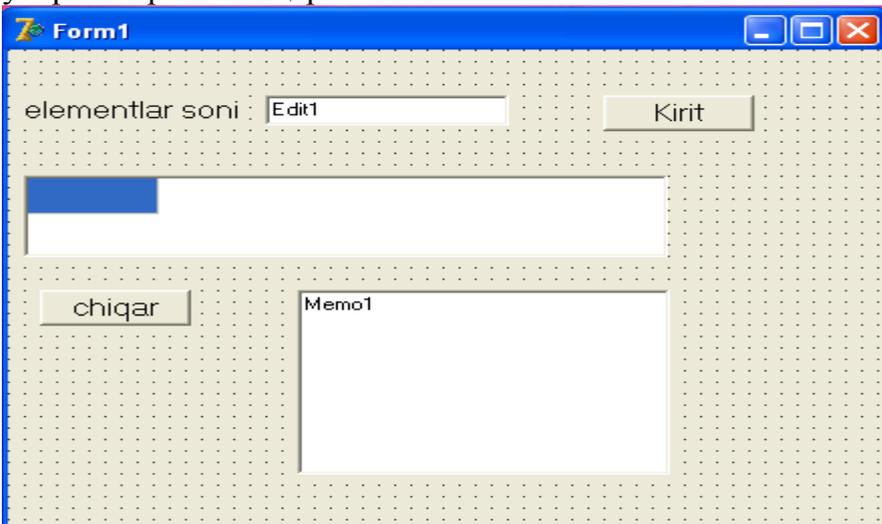
4.1-масала. n та элементдан ташкил топган массивнинг элементлари йиғиндисини топинг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
begin
N:=StrToInt(Edit1.Text);
StringGrid1.ColCount:=N;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
S:=0;
FOR I:=0 TO n-1 DO
S:=S+STRTOFLOAT(StringGrid1.Cells[I,0]);
Edit2.Text:= FloatToStr(S);
end;
end.
```

4.2-масала. n та элементдан иборат бир ўлчовли массив элементларини киритинг ва уларни экранга чиқаринг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
begin
```

```

N:=StrToInt(Edit1.Text);
StringGrid1.ColCount:=N;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
Memo1.Clear;
FOR I:=0 TO n-1 DO
memo1.Lines.Add(inttostr(i+1)+'-element '+StringGrid1.Cells[i,0]);
end;end.

```

4.3-масала. N та элементдан иборат бир ўлчовли массив элементларнинг йиғиндиси S ва манфий элементлар сони K ни аниқланг.



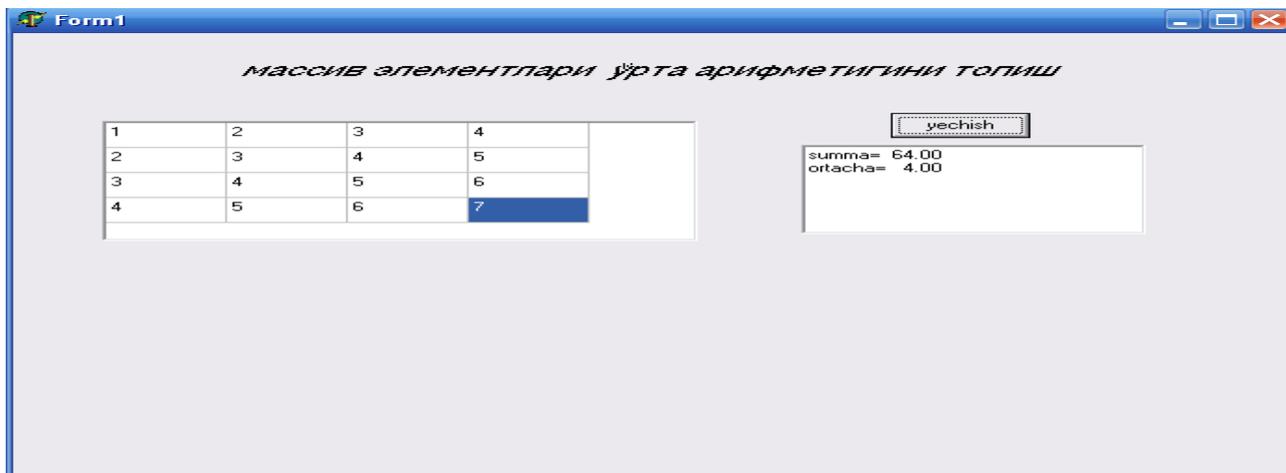
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
VAR N:Integer;
begin
N:=StrToInt(Edit1.Text);
StringGrid1.ColCount:=N;
end;
procedure TForm1.Button2Click(Sender: TObject);
var S:REAL;
I,k:INTEGER;
begin
S:=0;k:=0;
FOR I:=0 TO strtoint(Edit1.Text)-1 DO
S:=S+STRTOFLOAT(StringGrid1.Cells[I,0]);
Edit2.Text:= FloatToStr(S);
FOR I:=0 TO strtoint(Edit1.Text)-1 DO
if STRTOFLOAT(StringGrid1.Cells[I,0])<0 then k:=k+1;
Edit3.Text:=inttostr(k);
end;
end.

```

4.4. Бутун сонли массив элементлари йиғиндисини ва ўртача арифметик қийматини ҳисоблаш



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var i,j,cod:integer;
a:array[1..4,1..4] of real;
s:real; s1:string;
begin
for i:=1 to 4 do
for j:=1 to 4 do
val(stringGrid1.Cells[i-1,j-1],a[i,j],cod);
s:=0;
for i:=1 to 4 do
for j:=1 to 4 do
s:=s+a[i,j];
str(s:7:2,s1);
memo1.clear;
memo1.lines.add('summa='+s1);
s:=s/4/4;
str(s:7:2,s1);
memo1.lines.add('ortacha='+s1);
end;
end.
```

4.5-масала N та элементдан иборат массив элементларнинг ўрнини алмаштириб, уларни ўсадиган тартибда жойлаштиринг.

Дастур матни

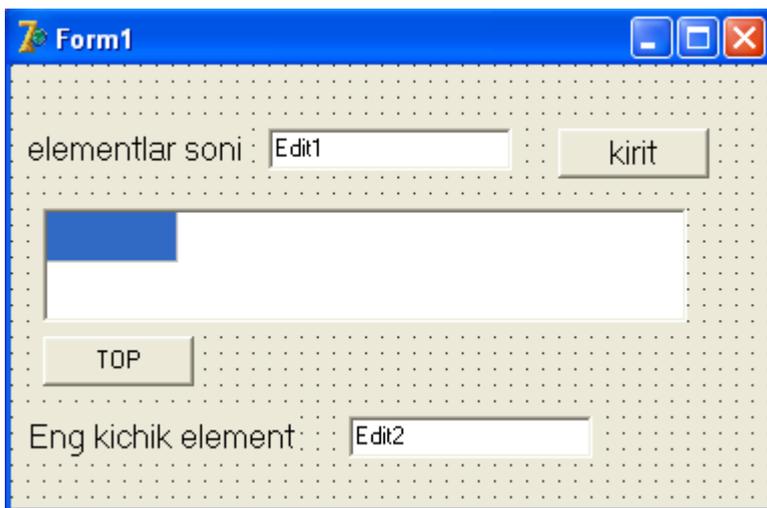
```
procedure TForm1.Button1Click(Sender: TObject);
VAR N:Integer;
begin
N:=StrToInt(Edit1.Text);
StringGrid1.ColCount:=N;
```

```

end;
procedure TForm1.Button2Click(Sender: TObject);
label 1;
var a,a1,k:real; I:byte;
begin
1:for I:=0 TO strtoint(Edit1.Text)-2 DO begin
a:=StrToFloat(StringGrid1.Cells[i,0]);
a1:=StrToFloat(StringGrid1.Cells[i+1,0]);
if a>a1 then
begin k:=a;a:=a1;a1:=k;
StringGrid1.Cells[i,0]:=FloatToStr(a);
StringGrid1.Cells[i+1,0]:=FloatToStr(a1);
end;end;
FOR I:=0 TO strtoint(Edit1.Text)-2 DO begin
a:=StrToFloat(StringGrid1.Cells[i,0]);
a1:=StrToFloat(StringGrid1.Cells[i+1,0]);
if a>a1 then goto 1;end;
end;end.

```

4.6-масала. A(N) массивнинг энг кичик элементини топинг.



Дастур матни

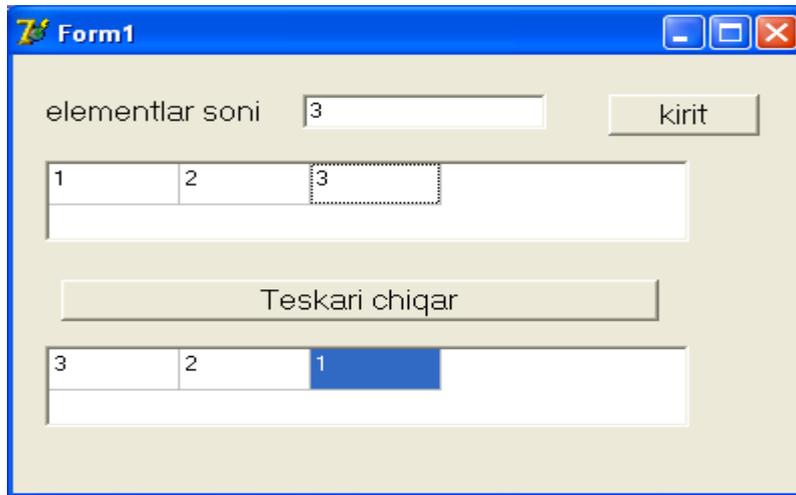
```

procedure TForm1.Button1Click(Sender: TObject);
VAR N:Integer;
begin
N:=StrToInt(Edit1.Text);
StringGrid1.ColCount:=N;
end;
procedure TForm1.Button2Click(Sender: TObject);
var I:byte;m:real;
begin
m:=strtofloat(StringGrid1.Cells[0,0]);
FOR I:=1 TO strtoint(Edit1.Text)-1 DO
if m>strtofloat(stringGrid1.Cells[I,0]) then
m:=strtofloat(StringGrid1.Cells[i,0]);
Edit2.Text:= FloatToStr(m);

```

end;end.

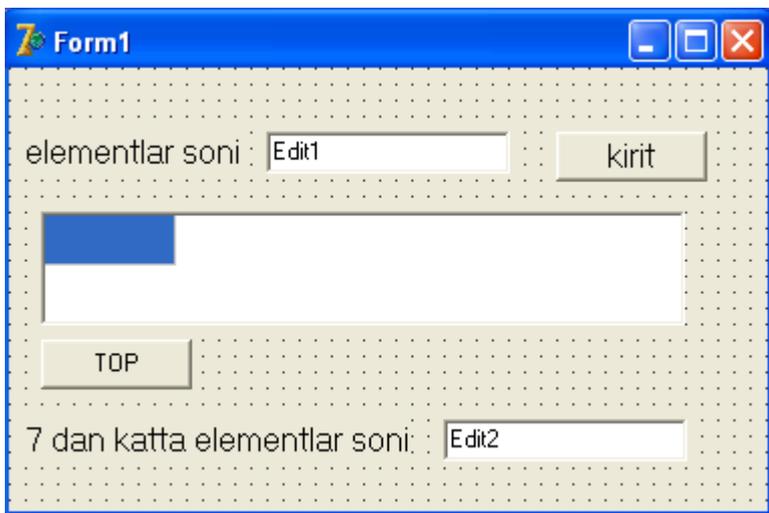
4.7-масала. $A(1), A(2), \dots, A(N)$ кетма-кетликда берилган жадвал элементларини тескари тартибда экранга чиқаринг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
begin
N:=StrToInt(Edit1.Text);
StringGrid1.ColCount:=N;
StringGrid2.ColCount:=N;
end;
procedure TForm1.Button2Click(Sender: TObject);
var a,a1,k:real;I:byte;
begin
for I:=0 TO n-2 div 2 DO begin
a:=StrToFloat(StringGrid1.Cells[i,0]);
a1:=StrToFloat(StringGrid1.Cells[n-i-1,0]);
begin k:=a;a:=a1;a1:=k;
StringGrid2.Cells[i,0]:=FloatToStr(a);
StringGrid2.Cells[i+1,0]:=FloatToStr(a1);
end;end;end;end.
```

4.8-масала. Бир ўлчовли массивнинг элементлари $A(N)$ берилган. Абсолют қиймати 7 дан катта бўлган массив элементлари сонини аниқланг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
VAR N:Integer;
begin
N:=StrToInt(Edit1.Text);
StringGrid1.ColCount:=N;
end;
procedure TForm1.Button2Click(Sender: TObject);
var I,t:byte;
begin
t:=0;
FOR I:=0 TO strtoint(Edit1.Text)-1 DO
if strtfloat(stringGrid1.Cells[I,0])>7 then t:=t+1;
Edit2.Text:=intToStr(t);
end;end.

```

4.9-мисол. StringGrid жадвал компонентасидан фойдаланиб, N та соннинг йигиндисини топиш дастурини тузинг.

Дастур матни

```

procedure tform1.button1click(sender:tobject);
var n:integer;
begin
n:=strtoint(edit1.text);
stringgrid1.colcount:=n;
end;
procedure tform1.button2click(sender:tobject);
const n=4;
var i:integer;
a:array[0..n] of real;
s:real;
begin
s:=0;
for i:=0 to n do
begin
a[i]:= strtoint(stringgrid1.cells[i,0]);
s:=s+a[i];

```

```

end;
edit2.text:=floattostr(s);
end;
end.

```

4.10-мисол. Мемо компонентасидан фойдаланиб, ҳафта кунларини чоп этиш дастурини тузинг.

Дастур матни

```

unit hafta kunlari;
interface
uses windows,messages,sysutils,classes, graphics, controls,forms,dialogs,stdctrls;
type
tform1=class(tform)
button1: tbutton;
memo 1:tmemo;
procedure button1click(sender:tobject);
private
{ private declarations }
public
{ public declarations }
end;
var
form1:tform1;
implementation
{$r*.dfm}
procedure tform1.button1click(sender:tobject);
var kun:array[1..7] of string[11];
i:integer;
begin
kun[1]:=’dushanba’;
kun[2]:=’seshanba’;
kun[3]:=’chorshanba’;
kun[4]:=’payshanba’;
kun[5]:=’juma’;
kun[6]:=’shanba’;
kun[7]:=’yakshanba’;
for i:=1 to 7 do
memo1.lines.add(kun[i]);
end;
end.

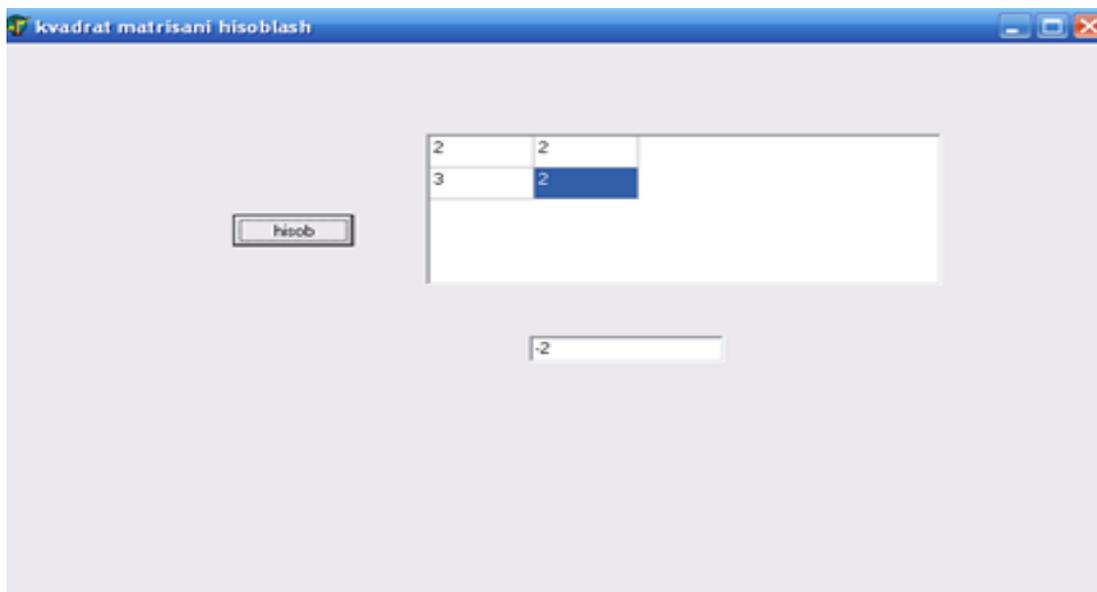
```

4.11-мисол. Мемо ва StringGrid компонентасидан фойдаланиб, массив элементларининг энг кичигини топиш дастурини тузинг.

Дастур матни

```
Procedure TForm1.Button1Click(Sender:Tobject);
Const n=4;
Var
a:array[0..n] of integer;
min,i:integer;
Begin
For i:=0 to n do
Begin
A[i]:= strtoint(StringGrid1.cells[I,0]);
End;
Min:=0;
For i:=1 to n do
Begin
If a[i]<a[min] then
begin
min:=i;
End;
Memo1.lines.add(floattostr(a[min]));
End;
End;
End.
```

4.12. Квадрат матрицани ҳисоблаш дастури



Дастур матни

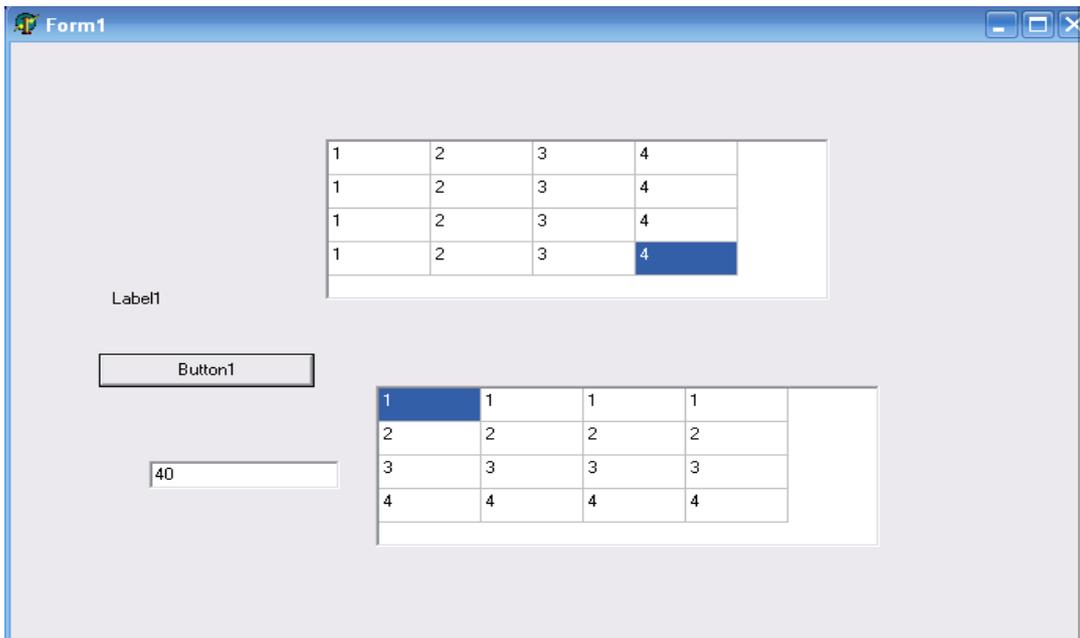
```
procedure TForm1.BitBtn1Click(Sender: TObject);
```

```

const n=1;
m=1;
var i,j:integer;
y:real;
a:array[0..n,0..m] of real;
begin
for i:=0 to n do
begin
for j:=0 to n do
begin
a[i,j]:=strtoint(stringgrid1.Cells[i,j]);
end;
end;
y:=a[0,0]*a[1,1]-a[0,1]*a[1,0];
edit1.text:=floattostr(y);
end;
end.

```

4.13. Квадрат матрицани транспонирлаш дастури



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
const n=3; m=3;
var i,j:integer;
s:real;
a:array[0..n,0..m] of real;

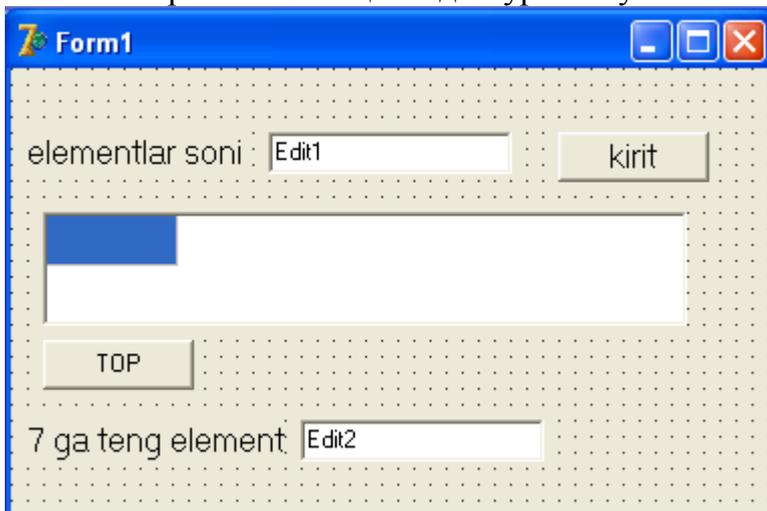
```

```

begin
s:=0;
for i:=0 to n do
begin
for j:=0 to m do
begin
a[i,j]:=strtofloat(stringgrid1.Cells[i,j]);
stringgrid2.Cells[j,i]:=floattostr(a[i,j]);
end;
end;
for i:=0 to n do
begin
for j:=0 to m do
begin
s:=s+a[i,j];
end;
end;
edit1.text:=floattostr(s);
end;
end.

```

4.14-масала. n та элементдан иборат бир ўлчовли массивнинг элементлари ичида 7 га тенг элемент борлигини аниқлаш дастурини тузинг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
VAR N:Integer;
begin
N:=StrToInt(Edit1.Text);
StringGrid1.ColCount:=N;
end;
procedure TForm1.Button2Click(Sender: TObject);
var I,t:byte;
begin t:=0;
FOR I:=0 TO strtoint(Edit1.Text)-1 DO
if strttofloat(stringGrid1.Cells[I,0])=7 then t:=t+1;

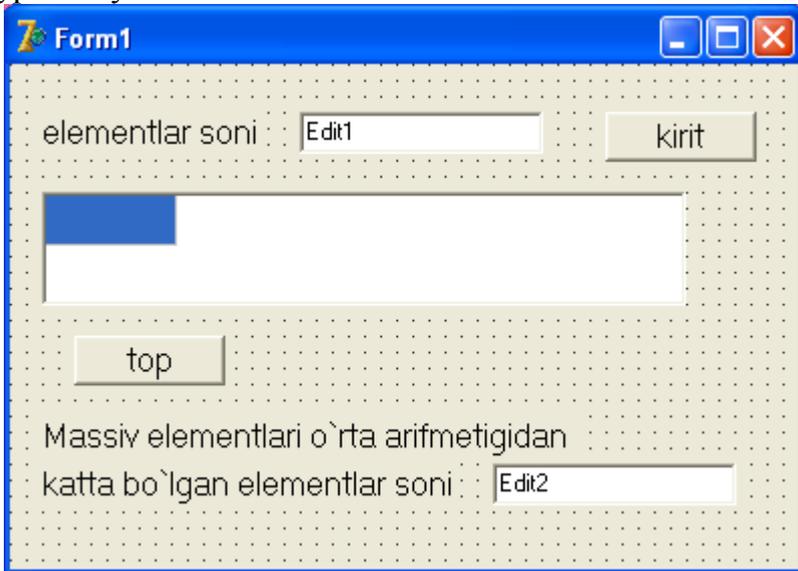
```

```

if t>0 then Edit2.Text:='bor' else Edit2.Text:='yo`q';
end;
end.

```

4.15-масала. n та элементдан иборат бир ўлчовли массив берилган. Бу массив элементларининг ўрта арифметик қийматидан катта бўлган элементлар сонини топиш дастурини тузинг.



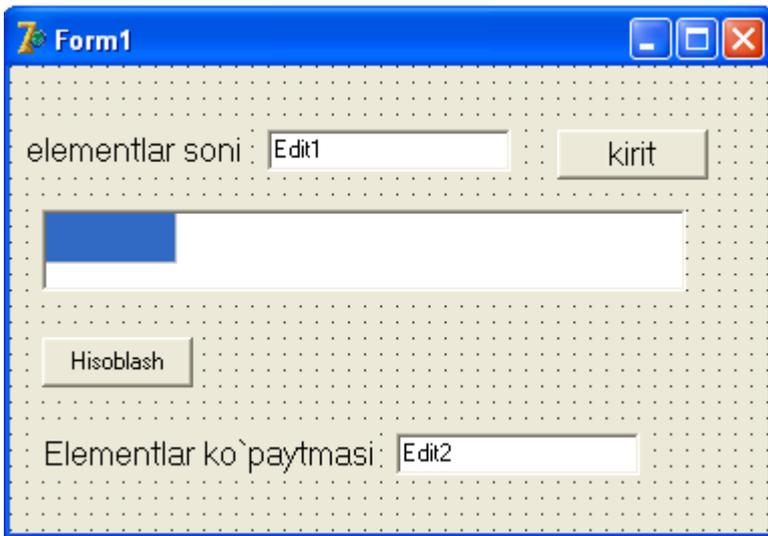
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
VAR N:Integer;
begin
N:=StrToInt(Edit1.Text);
StringGrid1.ColCount:=N;
end;
procedure TForm1.Button2Click(Sender: TObject);
var S:REAL;
I,k:INTEGER;
begin
S:=0;k:=0;
FOR I:=0 TO strtoint(Edit1.Text)-1 DO
S:=S+STRTOFLOAT(StringGrid1.Cells[I,0]);
s:=s/strtoint(Edit1.Text);
FOR I:=0 TO strtoint(Edit1.Text)-1 DO
if STRTOFLOAT(StringGrid1.Cells[I,0])>s then k:=k+1;
Edit2.Text:=inttostr(k);
end;
end.

```

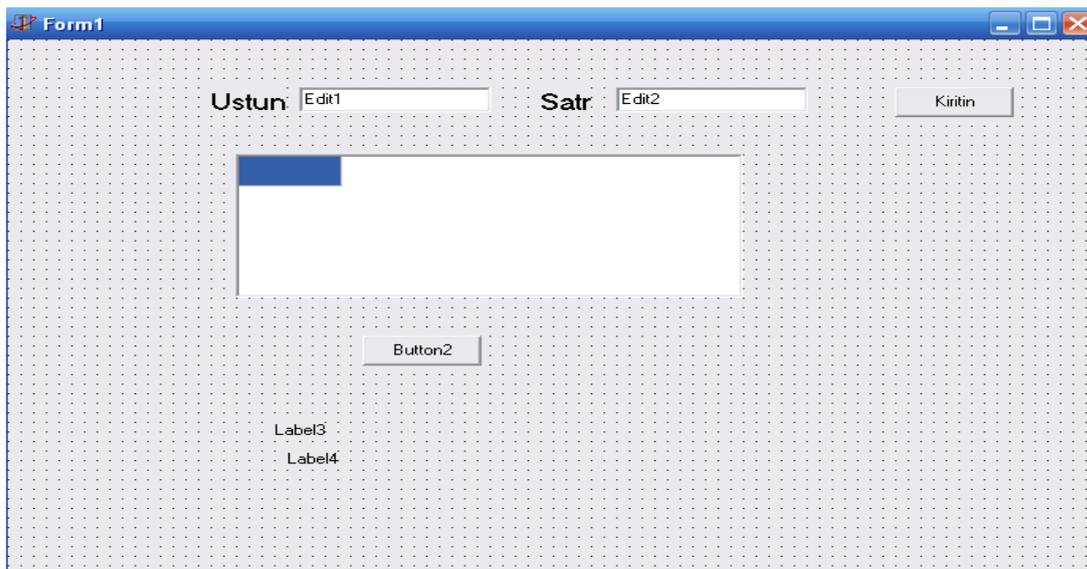
4.16-масала. $A(N)$ жадвал элементлари кўпайтмасини топиш дастурини тузинг.



Дастур матни

```
procedure tform1.button1click(sender: tobject);
var n:integer;
begin
n:=strtoint(edit1.text);
stringgrid1.colcount:=n;
end;
procedure tform1.button2click(sender: tobject);
var s:real;
i:integer;
begin
s:=1;
for i:=0 to strtoint(edit1.text)-1 do
s:=s*strtofloat(stringgrid1.cells[i,0]);
edit2.text:= floattostr(s);
end;
end.
```

4.17-масала. 2 дан 9 гача бўлган натурал сонларни кўпайтириш жадвалини синаб кўриш дастрини тузиш.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var n,m:integer;
begin
n:=StrToInt(Edit1.Text);
m:=StrToInt(Edit2.Text);
StringGrid1.RowCount:=m;
StringGrid1.ColCount:=n;
end;
procedure TForm1.Button2Click(Sender: TObject);
const
SIZE = 10;
var
a: array[1..SIZE] of integer;
min: integer; { минимал элемент номери i дан массивнинг энг юкори чегараси
гача }
j: integer; { минимал элемент билан таккосланаётган элемент номери }
buf: integer; { массив элементларини алмаштиришда кулланиладиган буфер }
i, k: integer;
begin // массивни киритиш
for i := 1 to SIZE do
a[i] := StrToInt(StringGrid1.Cells[i - 1, 0]); label2.caption := "";
for i := 1 to SIZE - 1 do
begin
{ a[1] дан a[SIZE] гача бўлган минимал элементни кидириш }
min := i;
for j := i + 1 to SIZE do
if a[j] < a[min] then
min := j;
{ a [min] билан a[i] нинг уринларини алмаштириш }

```

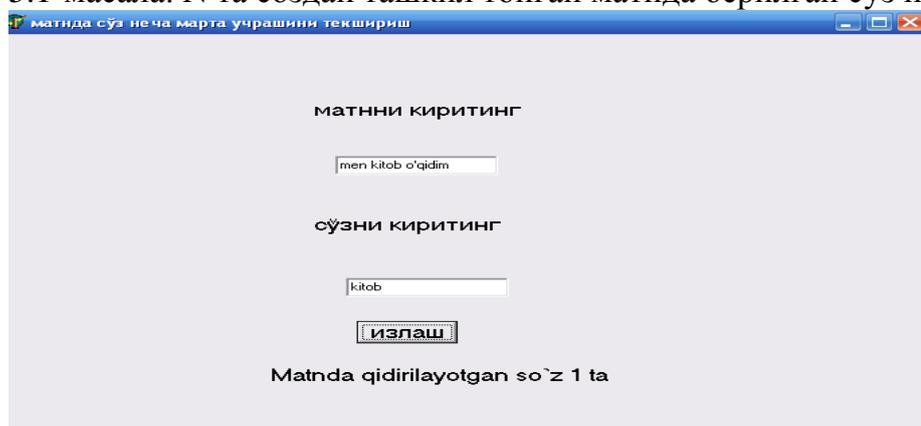
```

buf := a[i];
a[i] := a[min];
a[min] := buf;
{ массивни чиқариш }
for k := 1 to SIZE do
Label2.caption := label2.caption + ' ' + IntToStr(a[k]);
Label2.caption := label2.caption + #13;
end;
Label2.caption := label2.caption + #13 + 'МассМВ сараланди.';
end;

```

5. Матн билан ишлаш алгоритмларига оид дастурлар.

5.1-масала. N та создан ташкил топган матнда берилган сўз неч марта учраши аниқлансин.



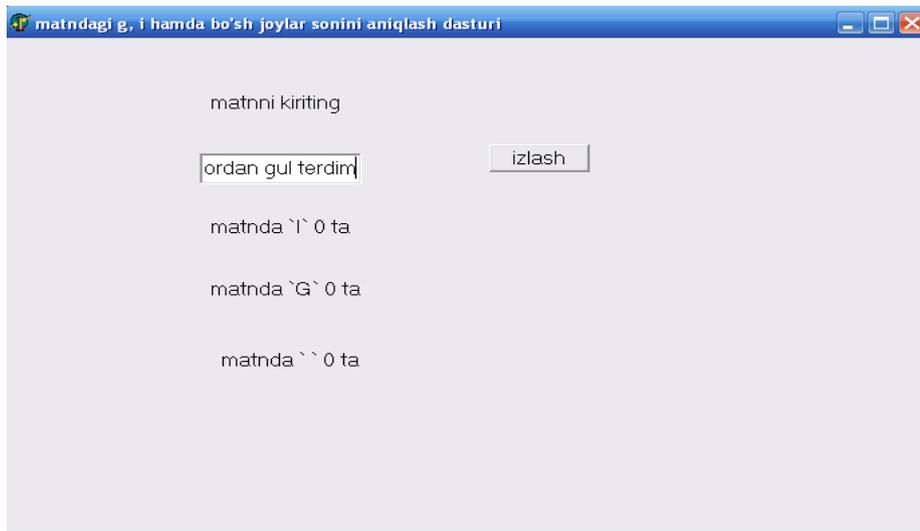
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var s:string; a:string[12];
b,m,n,i:integer;
begin
s:=edit1.Text;
a:=edit2.Text;
m:=length(s); n:=length(a); b:=0;
for i:= 1 to m-n+1 do
if copy(s,i,n)=a then b:=b+1;
if b=0 then label3.Caption:='Matnda izlanayotgan so`z yo`q' else
label3.Caption:='Matnda qidirilayotgan so`z '+inttostr(b)+' ta';
end; end.

```

5.2-масала. Берилган матндаги “G” ва “I” ҳарфлар ҳамда бўш жойлар сони аниқлансин.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var s:string[50]; b:shortstring;
a,n,m,t,i:integer;
begin
s:=edit1.Text;
a:=length(s);n:=0; m:=0; t:=0;
for i:=1 to a do begin
b:=copy(s,i,1);
if b='I' then n:=n+1;
if b='G' then m:=m+1;
if b=' ' then t:=t+1;end;
label2.Caption:='matnda `i` '+inttostr(n)+' ta';
label3.Caption:='matnda `g` '+inttostr(m)+' ta';
label4.Caption:='matnda `` '+inttostr(t)+' ta';
end;
end.

```

5.3-масала. Берилган матндаги белгилар сони иккилантирилсин.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var s,s1:string[100]; b:shortstring;
a,j,i:integer;
begin
s:=edit1.Text;
a:=length(s);
for i:=1 to a do begin
b:=copy(s,i,1);
for j:=1 to 2 do s1:=concat(s1,b); end;
edit2.Text:=s1;
end; end.

```

5.4-масала. Берилган сўз иккиёклама бўлишини аниқланг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
label 1,2;
var s:string[20];
a,i,t:integer;
begin
s:=edit1.Text;
a:=length(s); t:=0;
for i:=1 to a div 2 do
if copy(s,i,1)=copy(s,a+1-i,1) then t:=t+1
else goto 1;
if t=a div 2 then begin
label2.Caption:='so`z ikkiyoqlama';goto 2;end;
1:label2.Caption:='so`z ikkiyoqlama emas';
2:end;
end.

```

5.5-масала. Берилган матндаги ҳамма “И” ҳарфлар олиб ташлансин.



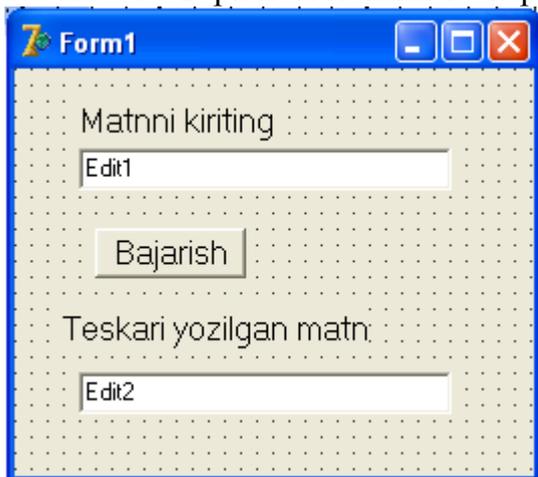
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var s:string[30]; i,n:byte;
begin
s:=edit1.Text;
n:=length(s);
for i:=1 to n do
if copy(s,i,1)='1' then delete(s,i,1);
edit2.Text:=s;
end; end.

```

5.6-масала. Берилган матнни тескарисига ёзинг.



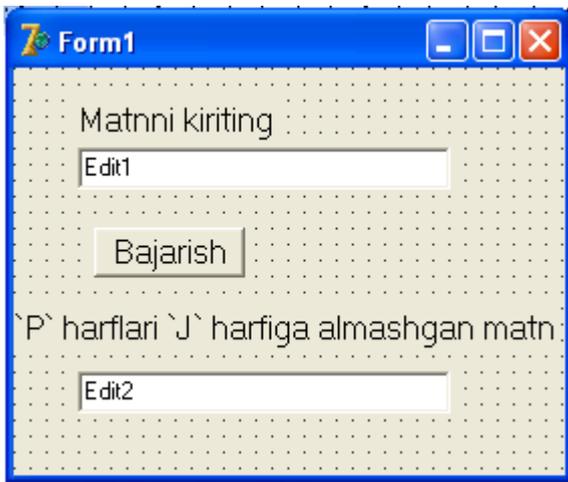
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var s,s1:string[30]; a:shortstring;
i,n:byte;
begin
s:=edit1.Text;
n:=length(s); s1:="";
for i:=1 to n do begin
a:=copy(s,n-i+1,1); s1:=concat(s1,a);end;
edit2.Text:=s1;
end; end.

```

5.7-масала. Берилган матндаги “P”харфларни “J”харфларга алмаштиринг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var s:string[30]; i,n:byte;
begin
s:=edit1.Text;
n:=length(s);
for i:=1 to n do
if copy(s,i,1)='P' then begin
delete(s,i,1); insert('J',s,i); end;
edit2.Text:=s;
end; end.

```

5.9-масала. Паскал тилидаги берилган операторларнинг инглизча-ўзбекча лу\атини тузинг. Бунда инглизча сўз киритилганда унинг таржимаси натижа сифатида олиниши керак.



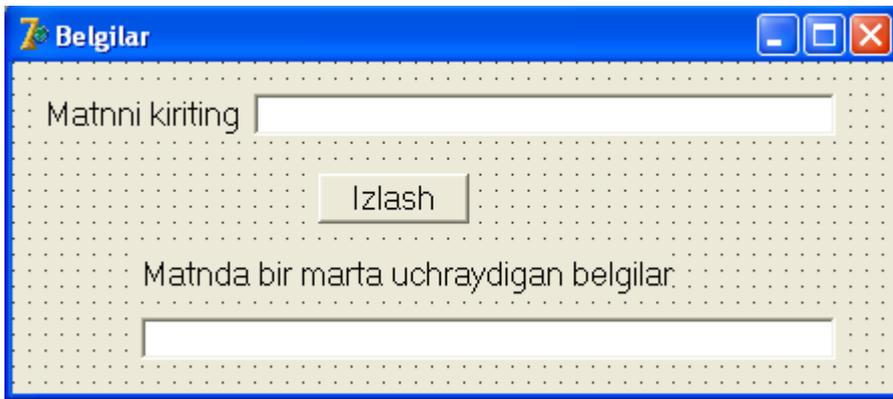
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var s:string[10];
begin
s:=edit1.text;
Label2.Caption:='noma`lum so`z';
if s='READ' then label2.Caption:='kiritish';
if s='IF' then Label2.Caption:='Agar';
if s='FOR' then Label2.Caption:='Uchun';
if s='WRITE' then Label2.Caption:='chiqarish';
if s='STOP' then Label2.Caption:='To`xta';
if s='RUN' then Label2.Caption:='Bajar';
end;end.

```

5.10-масала. Берилган матнда фақат бир марта учрайдиган белгиларни (матнда қандай учраса, шу тартибда) ажратинг.



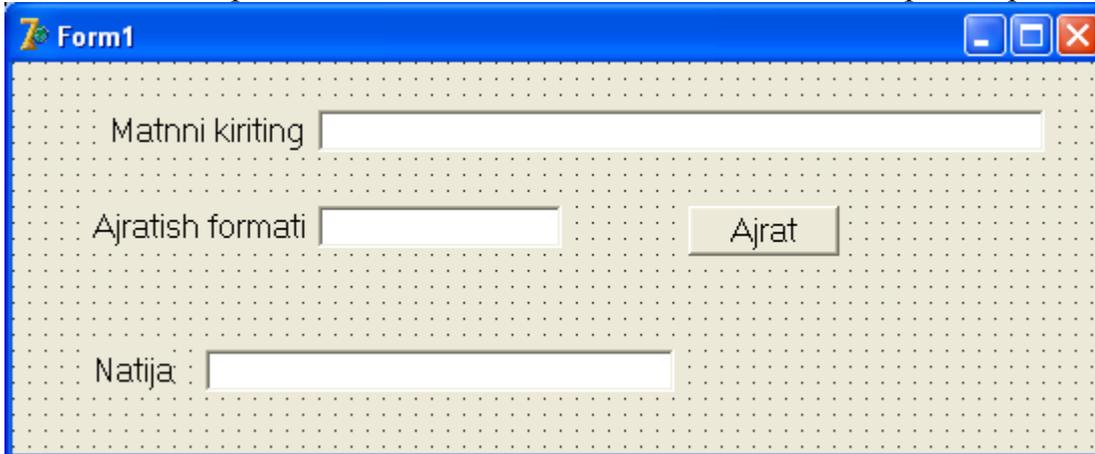
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var i,j,n,t:integer;s,s1:string;
begin
s:=edit1.Text;
n:=length(s); t:=0;
for i:=1 to n do begin t:=0;
for j:=1 to n do
if copy(s,i,1)=copy(s,j,1) then t:=t+1;
if t=1 then s1:=concat(s1,copy(s,i,1));end;
edit2.Text:=s1;
end;end.

```

5.11-масала. Берилган матннинг чап томонидан n та белгиларни ажратинг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a:integer;s:string;
begin
s:=edit1.text;
a:=strtoint(edit2.Text);
s:=copy(s,1,a);
edit3.Text:=s;
end;end.

```

5.12-масала. Берилган матннинг ўнг томонидан n та белгиларни ажратинг.

Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a:integer;s:string;
begin
s:=edit1.text;
a:=strtoint(edit2.Text);
delete(s,1,a);
edit3.Text:=s;
end;end.

```

5.13-масала. Берилган матннинг орасидан k-символдан n-символгача бўлган белгиларни ажратинг.

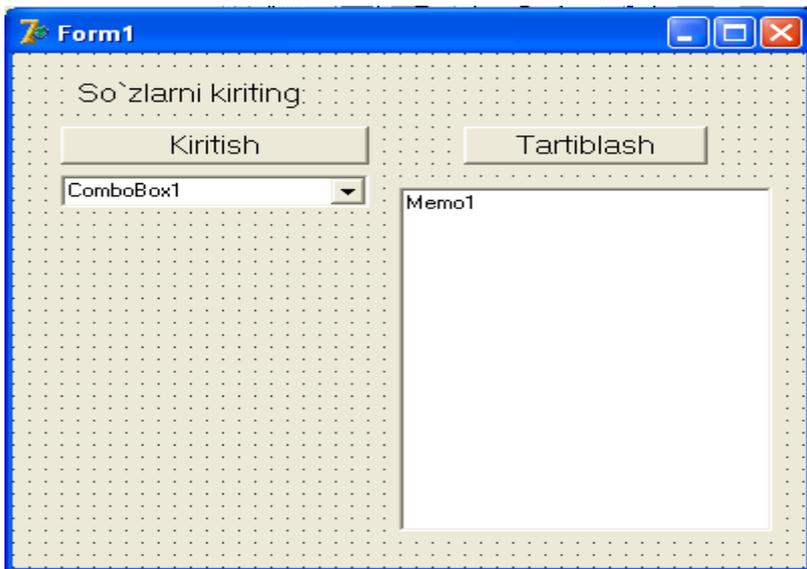
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var k,n:byte; s:string;
begin
s:=edit1.text;
k:=strtoint(Edit2.Text);
n:=strtoint(Edit3.Text);
s:=copy(s,k,n-k);
Edit4.Text:=s;
end;
end.

```

5.15-масала. Берилган сўзларни алфавит бўйича тартиблаш дастурини тузинг.



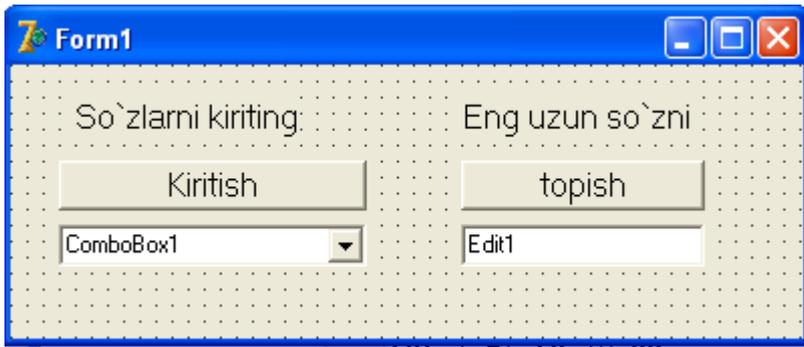
Дастур матни

```

var
  Form1: TForm1;
a:array[1..10] of string[20];
i:integer; b:string[20];
implementation
{$R *.dfm}
procedure TForm1.FormCreate(Sender: TObject);
begin
i:=0;
combobox1.Clear;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
combobox1.Items.Add(combobox1.Text);
i:=i+1;
a[i]:=combobox1.text;
combobox1.SetFocus;
end;
procedure TForm1.Button2Click(Sender: TObject);
label 2;
begin
2:for i:=1 to 9 do
if copy(a[i],1,1)>copy(a[i+1],1,1) then begin
b:=a[i];a[i]:=a[i+1];a[i+1]:=b;end;
for i:=1 to 9 do
if copy(a[i],1,1)>copy(a[i+1],1,1) then goto 2;
memo1.Clear;
for i:=1 to 10 do
memo1.Lines.Add(a[i]);
end; end.

```

5.16-масала. Берилган n та сўзлардан энг узунини аниқланг.



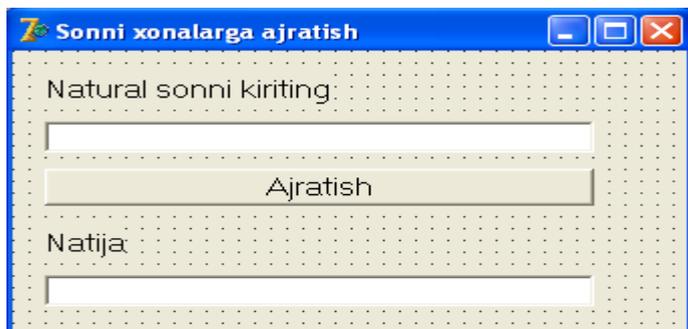
Дастур матни

```

var
  Form1: TForm1;
a:array[1..10] of string[20];
n,i,maxx,t:integer;
implementation
{$R *.dfm}
procedure TForm1.FormCreate(Sender: TObject);
begin
i:=0;
combobox1.Clear;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
combobox1.Items.Add(combobox1.Text);
i:=i+1;
a[i]:=combobox1.text;
combobox1.SetFocus;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
maxx:=length(a[1]);
for i:=2 to 10 do
if maxx<length(a[i]) then begin
maxx:=length(a[i]);t:=i;end;
edit1.text:=a[t];
end; end.

```

5.17-масала. Берилган натурал соннинг хона бирликларини ажратиб ёзинг.



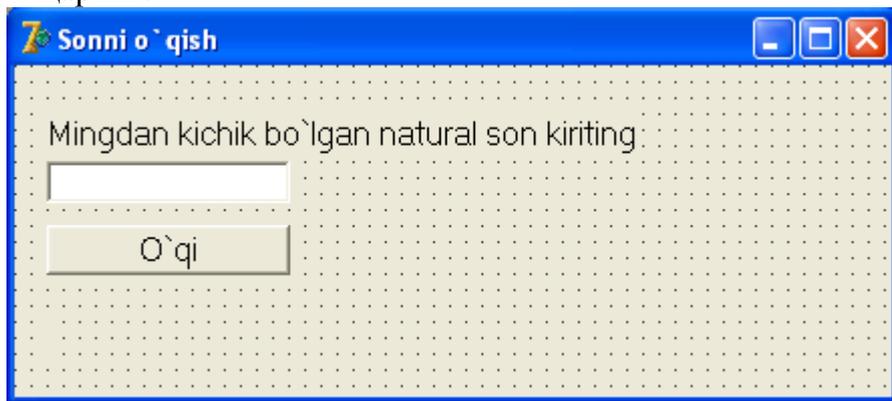
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var s:string; n,i:integer;
begin
s:=Edit1.Text;
n:=length(s);
for i:=1 to n div 3 do
insert(' ',s,n-3*i+1);
edit2.Text:=s;
end;end.

```

5.18-масала. Мингдан кичик бўлган берилган натурал сонни қандай ўқилишини босиб чиқаринг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var s,t:string[50];n:byte;
begin
s:=edit1.Text;
n:=length(s);label3.Caption:=' ';
if n=3 then begin
Label2.Caption:='Bu uch xonali son va u quyidagicha o`qiladi';
if copy(s,n-2,1)='1' then t:='bir yuz';
if copy(s,n-2,1)='2' then t:='ikki yuz';
if copy(s,n-2,1)='3' then t:='uch yuz';
if copy(s,n-2,1)='4' then t:='to`rt yuz';
if copy(s,n-2,1)='5' then t:='besh yuz';
if copy(s,n-2,1)='6' then t:='olti yuz';
if copy(s,n-2,1)='7' then t:='yetti yuz';
if copy(s,n-2,1)='8' then t:='sakkiz yuz';

```

```

if copy(s,n-2,1)='9' then t:='to`qqiz yuz';
label3.Caption:=label3.Caption+' '+t;
if copy(s,n-1,1)='1' then t:='o`n';
if copy(s,n-1,1)='2' then t:='yigirma';
if copy(s,n-1,1)='3' then t:='o`ttiz';
if copy(s,n-1,1)='4' then t:='qirq';
if copy(s,n-1,1)='5' then t:='ellik';
if copy(s,n-1,1)='6' then t:='oltmish';
if copy(s,n-1,1)='7' then t:='yetmish';
if copy(s,n-1,1)='8' then t:='sakson';
if copy(s,n-1,1)='9' then t:='to`qson';
label3.Caption:=label3.Caption+' '+t;
if copy(s,n,1)='1' then t:='bir';
if copy(s,n,1)='2' then t:='ikki';
if copy(s,n,1)='3' then t:='uch';
if copy(s,n,1)='4' then t:='to`rt';
if copy(s,n,1)='5' then t:='besh';
if copy(s,n,1)='6' then t:='olti';
if copy(s,n,1)='7' then t:='yetti';
if copy(s,n,1)='8' then t:='sakkiz';
if copy(s,n,1)='9' then t:='to`qqiz';
label3.Caption:=label3.Caption+' '+t;end;
if n=2 then begin
Label2.Caption:='Bu ikki xonali son va u quyidagicha o`qiladi';
if copy(s,n-1,1)='1' then t:='o`n';
if copy(s,n-1,1)='2' then t:='yigirma';
if copy(s,n-1,1)='3' then t:='o`ttiz';
if copy(s,n-1,1)='4' then t:='qirq';
if copy(s,n-1,1)='5' then t:='ellik';
if copy(s,n-1,1)='6' then t:='oltmish';
if copy(s,n-1,1)='7' then t:='yetmish';
if copy(s,n-1,1)='8' then t:='sakson';
if copy(s,n-1,1)='9' then t:='to`qson';
label3.Caption:=label3.Caption+' '+t;
if copy(s,n,1)='1' then t:='bir';
if copy(s,n,1)='2' then t:='ikki';
if copy(s,n,1)='3' then t:='uch';
if copy(s,n,1)='4' then t:='to`rt';
if copy(s,n,1)='5' then t:='besh';
if copy(s,n,1)='6' then t:='olti';
if copy(s,n,1)='7' then t:='yetti';
if copy(s,n,1)='8' then t:='sakkiz';
if copy(s,n,1)='9' then t:='to`qqiz';
label3.Caption:=label3.Caption+' '+t;end;
if n=1 then begin
Label2.Caption:='Bu bir xonali son va u quyidagicha o`qiladi';
if copy(s,n,1)='1' then t:='bir';
if copy(s,n,1)='2' then t:='ikki';
if copy(s,n,1)='3' then t:='uch';
if copy(s,n,1)='4' then t:='to`rt';
if copy(s,n,1)='5' then t:='besh';
if copy(s,n,1)='6' then t:='olti';
if copy(s,n,1)='7' then t:='yetti';

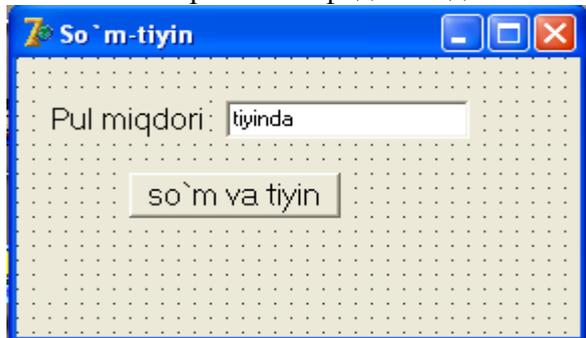
```

```

if copy(s,n,1)='8' then t:='sakkiz';
if copy(s,n,1)='9' then t:='to`qqiz';
label3.Caption:=label3.Caption+' '+t;end;
end;end.

```

5.19-масала. Тийинларда берилган пулни сўм ва тийин билан ифодаланг. Бунда тийинлар икки хонали сонлар билан ифодаланади.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var s:string[15];n:byte;
begin
s:=Edit1.Text;
n:=length(s); insert(' so`m ',s,n-1);
label2.Caption:=s+' tiyin';
end;end.

```

6. График чизишга доир дастурлар тузиш

6.1-масала. Параллел горизонтал чизиқлар чизиш дастури тузилсин.



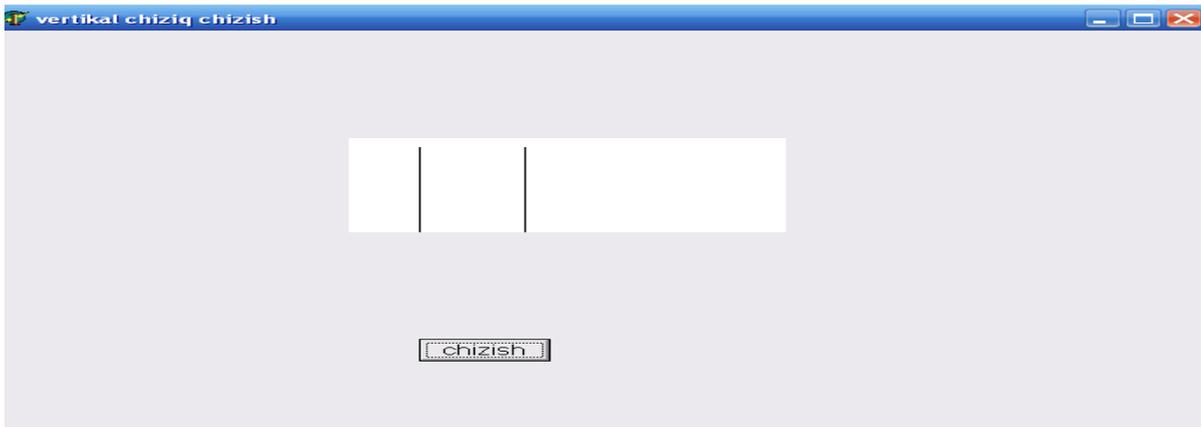
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
begin
with image1.Canvas do begin
moveto(10,40);
LineTo(300,40);
moveto(10,80);
LineTo(300,80);
end;
end;
end.

```

6.2-масала. Параллел вертикал чизиқлар чизиш дастури тузилсин.



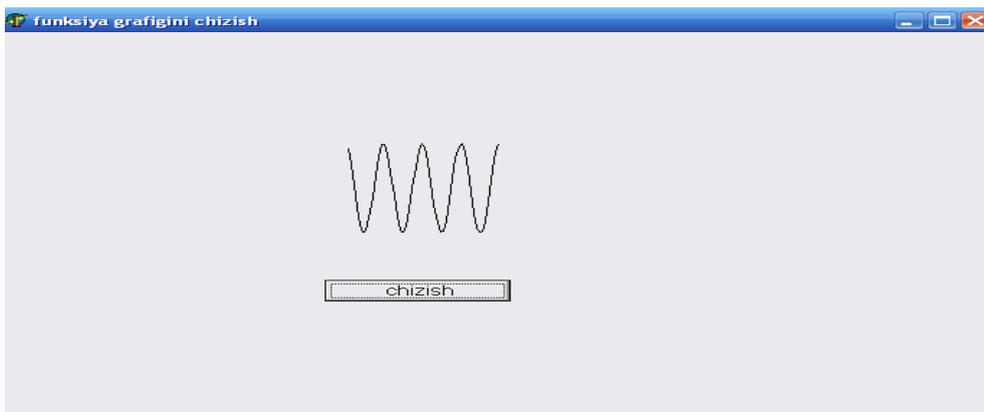
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
begin
with image1.Canvas do begin
moveto(40,10);LineTo(40,140);
moveto(100,10);LineTo(100,140);
end;
end;
end.

```

6.3-масала. Функция графигини чизиш дастурини тузинг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var
x, y: array[1..50] of double;
i: integer;
scalex, scaley, ymin, ymax, xmin, xmax: double;
begin
for i := 1 to 50 do
begin
y[i] := sin(i * 0.5);
x[i] := i;
end;
xmin := x[1];
xmax := x[1];
ymin := y[1];
ymax := y[1];
for i := 2 to 50 do

```

```

begin // или используйте ymin:=MinValue(y); и т.д.
if y[i] < ymin then
  ymin := y[i];
if y[i] > ymax then
  ymax := y[i];
if x[i] < xmin then
  xmin := x[i];
if x[i] > xmax then
  xmax := x[i];
end;
scalex := paintbox1.Width / (xmax - xmin);
scaley := paintbox1.Height / (ymax - ymin);
with paintbox1.canvas do
begin
moveto(trunc(scalex * (x[1] - xmin)), paintbox1.height - trunc(scaley * (y[1] - ymin)));
for i := 2 to 50 do
Lineto(trunc(scalex * (x[i] - xmin)), paintbox1.height - trunc(scaley * (y[i] - ymin)));
end;
end;
end.

```

Экранда берилган функция графигини чизиш программасини ёзинг.
TChart компонента воситасида график чизиш.

Функция графигини чизиш $y=f(x)$ функция қийматини, белгиланган қадамда $[X_{\min}, X_{\max}]$ оралиқда ҳисоблашдан бошланади. Ҳосил бўлган жадвал махсус икки ўлчамли массив $Series_k$ (k – график номери) $TChart$ компонентига ва Add методи ёрдамида узатилади. $PChart$ компоненти $Series_k$ объектига узатилган графикни тасвирлашдаги барча вазифани бажаради: ўқни белгилайди, координаталар ўқини чизади, ўқларни ва графикни белгилайди, узатилган жадвални график кўринишида акс эттиради.

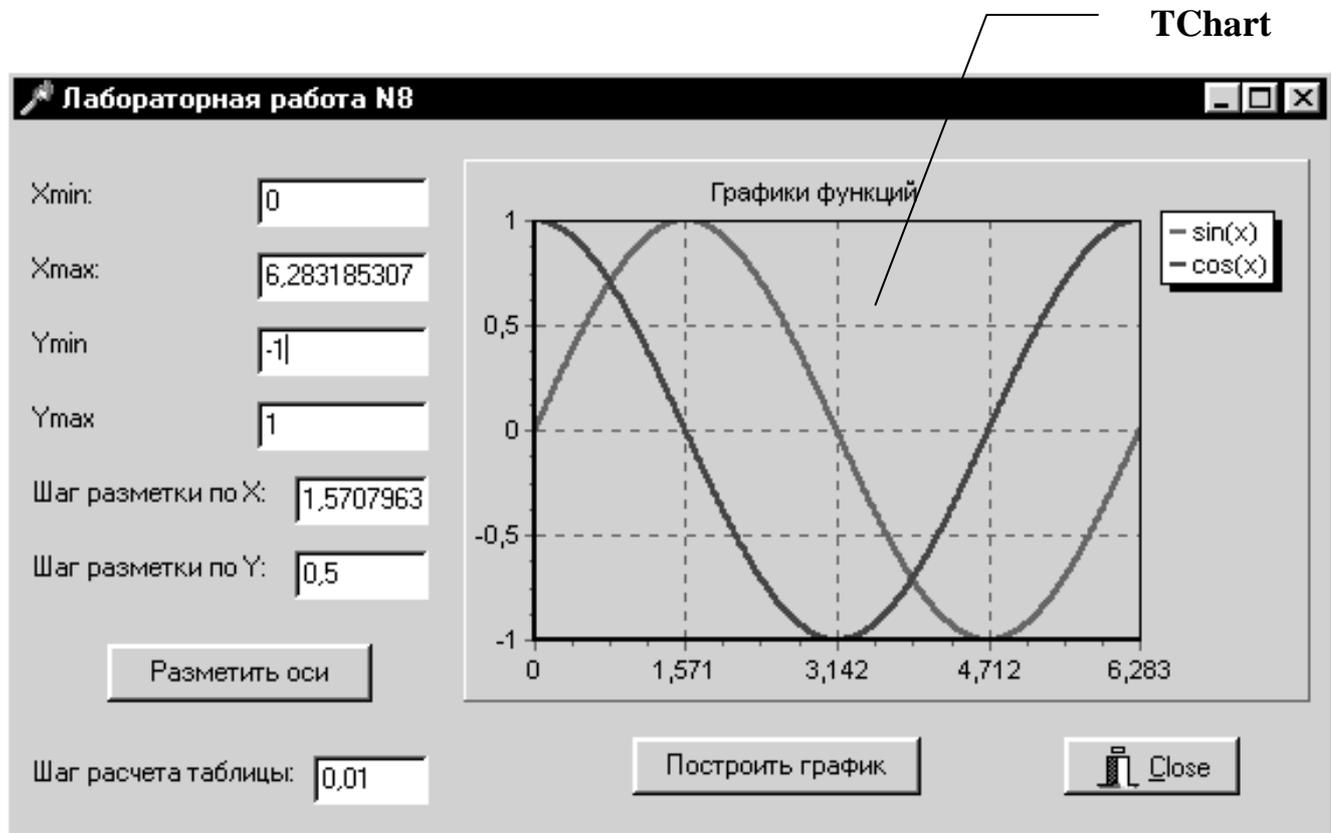


Рис.8.1

1-масала.Задание: составить программу, отображающую графики функций $\sin(x)$ и $\cos(x)$ на интервале $[X_{min}, X_{max}]$. Предусмотреть возможность изменения разметки координатных осей, а также шага построения таблицы.

8.2.1. Настройка формы

Панель диалога программы организуется в виде, представленном на рис.8.1.

Для ввода исходных данных используются окна TEdit. Компонент TChart вводится в форму путем нажатия пиктограммы  в меню компонентов Standard.

8.2.2. Работа с компонентом TChart

Для изменения параметров компонента TChart необходимо дважды щелкнуть по нему мышью в окне формы. Появится окно редактирования EditingChat1 (рис. 8.2). Для создания нового объекта Series1 щелкнуть по кнопке Add на странице Series. В появившемся диалоговом окне TeeChart Gallery выбрать пиктограмму с надписью Line (график выводится в виде линий). Если нет необходимости представления графика в трехмерном виде, отключить независимый переключатель 3D. После нажатия на кнопку ОК появится новая серия с название Series1. Для изменения названия нажать кнопку Title... В появившемся однострочном редакторе набрать имя отображаемой функции - “sin(x)”. Аналогичным образом создать объект Series2 для функции cos(x).

Для изменения надписи над графиком на странице Titles в многострочном редакторе набрать: “Графики функций”.

Для разметки осей выбрать страницу Axis и научиться устанавливать параметры настройки осей.

Нажимая различные кнопки меню, познакомиться с другими возможностями

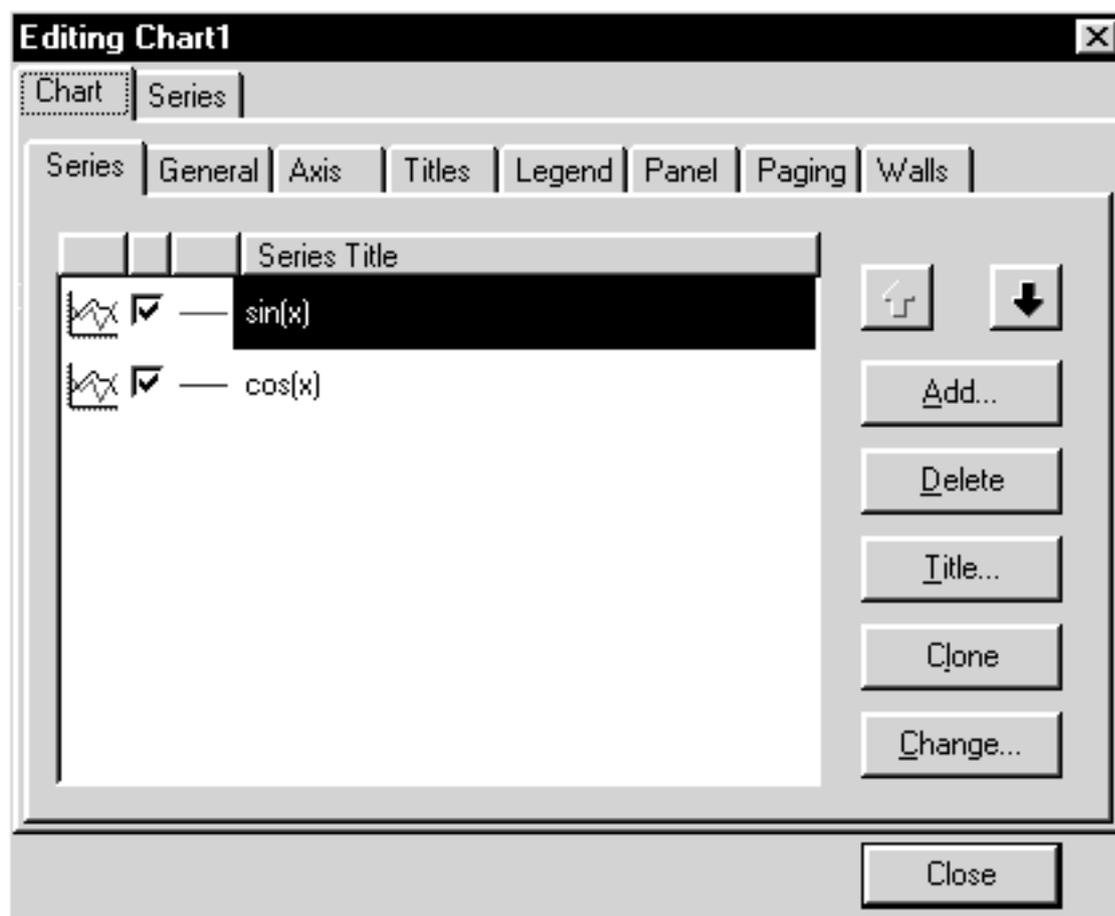


Рис.8.2

EditingChat.

8.2.3. Написание программы обработки события создания формы

В данном месте программы устанавливаются начальные пределы и шаг разметки координатных осей. Когда свойство Chart1.BottomAxis.Automatic имеет значения False, автоматическая установка параметров осей не работает.

8.2.4. Написание программ обработки событий нажатия на кнопки

Процедура TForm1.Button1Click обрабатывает нажатие кнопки “Установить оси”. Процедура TForm1.Button2Click обрабатывает нажатие кнопки “Построить график”. Для добавления координат точек (X,Y) из таблицы значений в двумерный массив объекта Seriesk используется процедура Series1.AddXY(Const AXValue, AYValue: Double; Const AXLabel: String; AColor: TColor) : Longint;, где AXValue, AYValue – координаты точки по осям X и Y; AXLabel может принимать значение ‘’; AColor задает цвет линий (если равен clTeeColor, то принимается цвет, определенный при проектировании формы).

Текст программы имеет вид:

```
unit tem8;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, TeeProcs, TeEngine, Chart, Buttons, StdCtrls, Series;
type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
    Button1: TButton;
    Button2: TButton;
    BitBtn1: TBitBtn;
    Chart1: TChart;
    Series2: TLineSeries;
    Label6: TLabel;
    Edit6: TEdit;
    Label7: TLabel;
    Edit7: TEdit;
    Series1: TLineSeries;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
  Xmin,Xmax,Ymin,Ymax,Hx,Hy,h : extended;

implementation
```

```

{$R *.DFM}
procedure TForm1.FormCreate(Sender: TObject);
begin
    {Установка начальных параметров координатных осей}
    Xmin:=0;
    Xmax:=2*pi;
    Ymin:=-1;
    Ymax:=1;
    Hx:=pi/2;
    Hy:=0.5;
    h:=0.01; // Установка шага расчета таблицы
    {Вывод данных в окна однострочных редакторов}
    Edit1.Text:=FloatToStr(Xmin);
    Edit2.Text:=FloatToStr(Xmax);
    Edit3.Text:=FloatToStr(Ymin);
    Edit4.Text:=FloatToStr(Ymax);
    Edit5.Text:=FloatToStr(Hx);
    Edit6.Text:=FloatToStr(Hy);
    Edit7.Text:=FloatToStr(h);
    Chart1.BottomAxis.Automatic:=False; // Отключение автоматического определения
        // параметров нижней оси
    Chart1.BottomAxis.Minimum:=Xmin; // Установка левой границы нижней оси
    Chart1.BottomAxis.Maximum:=Xmax; // Установка правой границы нижней оси
    Chart1.LeftAxis.Automatic:=False; // Отключение автоматического определения
        // параметров левой оси
    Chart1.LeftAxis.Minimum:=Ymin; // Установка нижней границы левой оси
    Chart1.LeftAxis.Maximum:=Ymax; // Установка верхней границы левой оси
    Chart1.BottomAxis.Increment:=Hx; // Установка шага разметки по нижней оси
    Chart1.LeftAxis.Increment:=Hy; // Установка шага разметки по левой оси
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
    {Чтение данных из окон однострочных редакторов}
    Xmin:=StrToFloat(Edit1.Text);
    Xmax:=StrToFloat(Edit2.Text);
    Ymin:=StrToFloat(Edit3.Text);
    Ymax:=StrToFloat(Edit4.Text);
    Hx:=StrToFloat(Edit5.Text);
    Hy:=StrToFloat(Edit6.Text);
    Chart1.BottomAxis.Minimum:=Xmin; // Установка левой границы нижней оси
    Chart1.BottomAxis.Maximum:=Xmax; // Установка правой границы нижней оси
    Chart1.LeftAxis.Minimum:=Ymin; // Установка нижней границы левой оси
    Chart1.LeftAxis.Maximum:=Ymax; // Установка верхней границы левой оси
    Chart1.BottomAxis.Increment:=Hx; // Установка шага разметки по нижней оси
    Chart1.LeftAxis.Increment:=Hy; // Установка шага разметки по левой оси
end;
procedure TForm1.Button2Click(Sender: TObject);
var x,y1,y2: extended;
begin
    {Очистка графиков}

```

```

Series1.Clear;
Series2.Clear;
Xmin:=StrToFloat(Edit1.Text);
Xmax:=StrToFloat(Edit2.Text);
h:=StrToFloat(Edit7.Text); // Шаг расчета таблицы для графика
x:=Xmin; // Начальное значение по оси X
repeat
y1:=sin(x); // Расчет функции
Series1.AddXY(x,y1,"clTeeColor"); // Вывод точки на график
y2:=cos(x); // Расчет функции
Series2.AddXY(x,y2,"clTeeColor"); // Вывод точки на график
x:=x+h; // Увеличение значения X на величину шага
Until (x>Xmax);
end;

end.

```

Математик формулалар

Язык Object Pascal имеет ограниченное количество встроенных математических функций. Поэтому при необходимости использовать другие функции следует применять известные соотношения. В таблице приведены выражения наиболее часто встречающихся функций через встроенные функции языка Object Pascal.

Функция	Соотношение	Соотношение на языке Object Pascal
$\log_a(x)$	$\frac{\ln(x)}{\ln(a)}$	Ln(x)/Ln(a)
x^a	$e^{a \cdot \ln(x)}$	Exp(a*Ln(x))
$Tg(x)$	$\frac{\sin(x)}{\cos(x)}$	Sin(x)/Cos(x)
$Ctg(x)$	$\frac{\cos(x)}{\sin(x)}$	Cos(x)/Sin(x)
$ArcSin(x)$	$ArcTg\left(\sqrt{\frac{x}{1-x^2}}\right)$	ArcTan(Sqrt(x/(1-sqr(x))))
$ArcCos(x)$	$\frac{\pi}{2} - ArcSin(x)$	Pi/2- ArcTan(Sqrt(x/(1-sqr(x))))
$ArcCtg(x)$	$\frac{\pi}{2} - ArcTg(x)$	Pi/2-ArcTan(x)

$Sh(x)$	$\frac{e^x - e^{-x}}{2}$	$(\text{Exp}(x) - \text{Exp}(-x))/2$
$Ch(x)$	$\frac{e^x + e^{-x}}{2}$	$(\text{Exp}(x) + \text{Exp}(-x))/2$
$Csc(x)$	$\frac{1}{\sin(x)}$	1/Sin(x)
$Sc(x)$	$\frac{1}{\cos(x)}$	1/Cos(x)

Приложение 1

Приложение 1. Команды основного меню

В меню File находятся команды для выполнения операций с проектами, модулями и файлами.

Команда	Описание
New	Позволяет выбрать тип элемента из репозитория (архива, в котором хранятся заготовки для новых программ) и создать его
New Application	Создает новый проект, состоящий из формы, модуля и файла проекта
New Form	Создает новую форму и подключает ее к проекту
New Data Module	Создает новый модуль данных и подключает его к проекту
Open	Открывает ранее созданный проект, модуль, форму или текстовый файл
Reopen	Вызывает список ранее загружавшихся проектов и форм для выбора и повторной загрузки
Save	Сохраняет текущую форму или модуль или файл
Save As	Сохраняет текущую форму с новым именем
Save Project As	Сохраняет текущий проект с новым именем
Save All	Сохраняет все открытые файлы, проект и используемые им модули
Close	Закрывает текущую форму
Close All	Закрывает все открытые файлы
Use Unit	Добавляет имя указанного модуля в список используемых модулей (USES) текущего активного модуля
Add to Project	Добавляет файл к проекту
Remove From Project	Удаляет файл из проекта
Print	Выводит содержимое активного файла на печать
Exit	Завершает работу Delphi

В меню Edit расположены команды, осуществляющие операции редактирования, работы с областью обмена данными, отмены действий и управления отображением компонентов.

Команда	Описание
Undo	Отменяет ранее выполненные действия
Redo	Восстанавливает отмененные действия
Cut	Вырезает выделенный объект и помещает его в буфер обмена данными
Copy	Копирует выделенный объект и (или) фрагмент текста программы и помещает его в буфер обмена данными
Paste	Копирует содержимое буфера обмена данными в редактор или форму
Delete	Удаляет выбранный объект или фрагмент программы
Select All	Выделяет все компоненты формы или весь текст программы
Align to Grid	Выравнивает выбранный компонент по сетке
Bring to Front	Перемещает выбранный компонент поверх других компонентов
Send to Back	Перемещает выбранный компонент под другие компоненты
Align	Выравнивает компоненты
Size	Изменяет размер выделенных компонентов
Scale	Изменяет размер всех компонентов в форме
Tab Order	Изменяет порядок табуляции компонентов в активной форме
Creation Order	Задаёт порядок создания невидимых компонентов
Lock Controls	Запрещает перемещение компонентов внутри формы
Add To Interface	Позволяет определить новую процедуру, функцию или свойство компонента ActiveX

Меню Search предоставляет команды для поиска и замены, а также команды для поиска указанных символов и строк, содержащих ошибки, найденные компилятором.

Команда	Описание
Find	Поиск указанного фрагмента текста
Find in files	Поиск указанного текста в нескольких файлах, задаваемых в диалоговой панели
Replace	Поиск указанного фрагмента текста и замена его новым текстом
Search Again	Повторный поиск или повторная замена
Incremental Search	Поиск текста по мере его ввода
Go to Line Number	Перемещение курсора на строку с указанным номером
Show Last Compile Error	Перемещение курсора на строку, содержащую ошибку, найденную компилятором
Find Error	Поиск ошибки времени исполнения (run-time error)
Browse Symbol	Показывает характеристики указанного символа программы по его имени

В меню View содержатся команды для отображения различной информации и вызова менеджера проектов, инспектора объектов, браузера объектов и других информационных утилит.

Команда	Описание
Project Manager	Менеджер проектов (Project Manager)
Project Source	Отображает исходный текст файла проекта
Object Inspector	Инспектор объектов (Object Inspector)
Alignment Palette	Палитра выравнивания компонентов
Browser	Броузер объектов (Object Browser)
Breakpoints	Список точек останова (Breakpoints List)
Call Stack	Стек вызовов (Call Stack)
Watches	Список точек слежения за переменными (Watch List)
Threads	Список потоков команд и их статус
Modules	Список модулей, загружаемых при выполнении данного проекта
Component List	Список компонентов
Window List	Список открытых окон
Toggle Form/Unit	Переключает активность из окна формы в окно текста программы и обратно
Unit	Показывает окно текста программы
Forms	Показывает окно формы
Type library	Отображает содержимое библиотеки типов для компонентов ActiveX, серверов ActiveX и других COM-объектов
New Edit Window	Открывает новое окно с текстом текущей программы
SpeedBar	Отображает (прячет) панель быстрого доступа
Component Palette	Отображает (прячет) палитру компонентов

В меню Project содержатся команды для компиляции и сборки проектов, а также для установки опций текущего проекта.

Команда	Описание
Add to Project	Добавляет файл к проекту
Remove from Project	Удаляет файл из проекта
Import Type Library	Импортирует в проект библиотеку типов элементов ActiveX
Add To Repository	Добавляет проект в репозиторий объектов
Compile	Компилирует модули, исходный текст которых изменился после последней компиляции
Build All	Компилирует все модули и создает исполняемую программу
Syntax Check	Проверяет синтаксическую правильность программы
Information	Отображает информацию о проекте
Web Deployment Options	Позволяет задать опции для внедрения компонента ActiveX или активной фирмы на Web-узел
Web Deploy	Внедряет компонент ActiveX или активную фирму на Web-узел
Options	Задаёт опции компилятора и компоновщика, управляет рабочими каталогами

В меню Run расположены команды для отладки программ. Эти команды позволяют управлять различными функциями устроенного отладчика.

Команда	Описание
Run	Компилирует и выполняет программу
Parameters	Задаёт параметры командной строки
Register ActiveX Server	Регистрирует сервер ActiveX в реестре Windows
Unregister ActiveX Server	Удаляет информацию о ранее зарегистрированном сервере ActiveX в реестре Windows
Step Over	Пошагово выполняет программу
Trace Into	Пошагово выполняет программу с заходом в подпрограммы
Trace To Next Source Line	Пошагово выполняет программу до следующей строки исходного текста
Run To Cursor	Выполняет программу до строки в окне редактора, на которой находится курсор
Show Execution Point	Отображает оператор, на котором было прервано выполнение программы
Program Pause	Приостанавливает выполнение программы
Program Reset	Завершает выполнение программы
Add Watch	Добавляет точку слежения за переменными
Add Breakpoint	Добавляет точку останова
Evaluate/Modify	Позволяет узнать или изменить значение переменной

В меню Component содержатся команды для создания компонентов, установки новых компонентов, импорта компонентов ActiveX, создания нового компонента на базе существующего и установки пакетов.

Команда	Описание
New Component	Вызывает окно эксперта компонентов
Install Component	Помещает компонент в существующий или новый проект
Import ActiveX Control	Импортирует компонент ActiveX
Create Component Template	Сохраняет компонент как шаблон для создания других компонентов
Install Package	Устанавливает пакеты, необходимые для прогона программы
Configure Palette	Вызывает диалоговую панель конфигурации палитры компонентов

Меню Database содержит средства для работы с базами данных.

Команда	Описание
Explore	Вызывает инструмент исследования баз данных - Database Explorer или SQL Database (в зависимости от версии DELPHI)
SQL Monitor	Вызывает инструмент запросов к БД – SQL Monitor
Form Wizard	Вызывает окно эксперта форм для создания формы, отображающей наборы данных из удаленных или локальных БД

Из меню Tools доступны средства настройки среды, дополнительные утилиты, входящие в состав Delphi, а также репозиторий объектов.

Команда	Описание
---------	----------

Environment Options	Вызывает диалоговую панель настройки среды
Repository	Вызывает репозиторий
Configure Tools	Вызывает диалоговую панель редактирования опции Tools
Package Collection Editor	Вызывает окно редактора пакетов
Image Editor	Вызывает окно редактора графики
Database Desktop	Вызывает инструмент обслуживания БД – Database Desktop

Меню Workgroups содержит средства для работы с коллективными проектами.

Команда	Описание
Browse PVCS Projects	Показывает окно коллективной работы нескольких программистов над одним проектом программы
Mange Archive Directories	Показывает диалоговое окно управления архивом коллективного проекта программы
Add Project to Version Control	Сохраняет текущую версию коллективного проекта
Set Data Directories	Показывает диалоговое окно выбора каталогов для размещения версий коллективного проекта

В меню Help содержатся команды для вызова различных разделов справочной системы и отображения диалоговой панели «О программе».

Команда	Описание
Contents	Отображает содержание справочной системы
Keyword Search	Выполняет поиск справки по ключевому слову
What's New	Отображает справку по новым возможностям продукта
Getting Started	Выводит онлайн-вариант книги «Getting Started»
Using Object Pascal	Выводит онлайн-вариант книги «Using Object Pascal»
Developing Applications	Выводит онлайн-вариант книги «Developing Applications»
Object and Component Reference	Выводит онлайн-вариант книги «Object and Component Reference»
Borland Home Page	Соединяет с главной страницей Web-узла фирмы Borland
Delphi Home Page	Соединяет со страницей Web-узла фирмы Borland, посвященной Delphi
Borland Programs and Services	Соединяет со страницей Web-узла фирмы Borland, посвященной программам и сервисам
About	Отображает диалоговую панель «О программе»

Paskal dasturlash tilida uchraydigan standart funksiyalar jadvali.

Funksiyani paskaldagi ifodasi	Funksiyaning matematik ifodasi	Argumentning turi	Funksiyaning turi
abs(x)	x	REAL, INTEGER	REAL, INTEGER
sqr(x)	x^2	REAL, INTEGER	REAL, INTEGER
sqrt(x)	\sqrt{x}	REAL, INTEGER	REAL, INTEGER

exp(x)	e^x	REAL, INTEGER	REAL REAL
ln(x)	$\ln x$	REAL, INTEGER	REAL, REAL
sin(x)	$\sin x$	REAL, INTEGER	REAL, REAL
cos(x)	$\cos x$	REAL, INTEGER	REAL, REAL
arctan(x)	$\arctg x$	REAL, INTEGER	REAL, REAL
round(x)	x ni yaxlitlash	REAL	INTEGER
trunc(x)	x ni butun qismini olish	REAL	INTEGER
pred(x)	x dan oldingi qiymatni olish.	INTEGER, CHAR BOOLEAN	INTEGER, CHAR BOOLEAN
succ(x)	x dan keyingi qiymatni olish.	INTEGER, CHAR BOOLEAN	INTEGER, CHAR BOOLEAN
a div b	A ni B ga bo'lib butun qismini olish	INTEGER, INTE- GER	INTEGER
a mod b	A ni B ga bo'lib qoldig'ini olish	INTEGER INTEGER	INTEGER
chr(x)	x ni tartib nomeriga ko'ra simvolini aniqlash	INTEGER	CHAR
ord(x)	x simvolini tartib nomerini aniqlash	CHAR	INTEGER
odd(x)	x ning toq yoki juftligini aniqlash	INTEGER x-toq, x-juft	BOOLEAN TRUE, FALSE

x^n ko'rinishdagi ifodani ko'p uchraydi, uni matematik almashtirishlardan foydalanib $x^{n \cdot \ln x}$ ko'rinishda yozsa bo'ladi. Bu ifodani Paskaldagi ko'rinishini $\exp(n \cdot \ln(x))$ shaklda yozish mumkin.

II. TAJRIBA ISHI TOPSHIRIQLARI

Topshiriq: 1) Quyidagi topshiriqlarni algoritm, blok-sxemasi va dasturini Paskal dasturlash tilida tuzing:

1.	$a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y} \quad b = 1 + \frac{z^2}{3 + z^3/5}$	x = 1,426 y = -1,220, z = 3,5
2.	$j = \left x^{y/x} - \sqrt[3]{y/x} \right \quad \psi = (y-x) \frac{y-z/(y-x)}{1+(y-x)^2}$	x = 1,825 y = 18,225 z = -3,298
3.	$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}, \quad \varphi = x(\sin x^3 + \cos^2 y)$	x = 0,335 y = 0,025
4.	$y = e^{-bt} \sin(at + b) - \sqrt{ bt + a }, \quad S = b \sin(at^2 \cos 2t) - 1$	a = -0,5, b = 1,7 t = 0,44
5.	$\omega = \sqrt{x^2 + b} - b^2 \sin^3(x + a)/x$ $y = \cos^2 x^3 - \frac{x}{\sqrt{a^2 + b^2}}$	a = 1,5 b = 15,5 x = -2,9
6.	$S = x^3 \lg^2(x + b)^2 + \frac{a}{\sqrt{x + b}}, \quad Q = \frac{bx^2 - a}{Ax}$	a = 16,5, b = 3,4 j = 0,61
7.	$R = x^2(x + 1)/b - \sin^2(x + a), \quad S = \sqrt{xb/a} + \cos^2(x + b)^3$	a = 0,7, b = 0,05 x = 0,5

8. .	$y = \sin^3(x^2 + a)^2 - \sqrt{x/b}$, $Z = \frac{x^2}{A} + \cos(x+b)^2$	a = 1,1, b = 0,004 x = 0,2
9. .	$f = \sqrt[3]{mtgt + c \sin t }$, $z = m \cos(bt \sin t) + c$	m = 2, c = -1 t = 1,2, b = 0,7
10. .	$y = btg^2 x - \frac{A}{\sin^2(x/a)}$, $S = b \sin(at^2 \cos 2t) - 1$	a = 3,2 b = 17,5, x = -4,8
11. .	$a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}$, $b = 1 + \frac{z^2}{3 + z^3/5}$	x = 1,4 y = -1,2, z = 3,05
12. .	$j = \left x^{y/x} - \sqrt[3]{y/x} \right $, $\psi = (y-x) \frac{y - z/(y-x)}{1 + (y-x)^2}$	x = 1,8 y = 18,2 z = -3,02
13. .	$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}$, $\varphi = x(\sin x^3 + \cos^2 y)$	x = 0,303 y = 0,02
14. .	$y = e^{-bt} \sin(at + b) - \sqrt{ bt + a }$, $S = b \sin(at^2 \cos 2t) - 1$	a = -0,05, b = 1,17 t = 0,24
15. .	$\omega = \sqrt{x^2 + b} - b^2 \sin^3(x+a)/x$, $y = \cos^2 x^3 - \frac{x}{\sqrt{a^2 + b^2}}$	a = 1,15 b = 15,05 x = -2,19
16. .	$S = x^3 tg^2(x+b)^2 + \frac{a}{\sqrt{x+b}}$, $Q = \frac{bx^2 - a}{Ax}$	a = 1,5, b = 3,14 j = 0,65
17. .	$R = x^2(x+1)/b - \sin^2(x+a)$, $S = \sqrt{xb/a} + \cos^2(x+b)^3$	a = 0,17, b = 0,5 x = 0,15
18. .	$y = \sin^3(x^2 + a)^2 - \sqrt{x/b}$, $Z = \frac{x^2}{A} + \cos(x+b)^2$	a = 1,01, b = 0,04 x = 0,12
19. .	$f = \sqrt[3]{mtgt + c \sin t }$, $z = m \cos(bt \sin t) + c$	m = 2, c = -1 t = 1,02, b = 0,17
20. .	$y = btg^2 x - \frac{A}{\sin^2(x/a)}$, $S = b \sin(at^2 \cos 2t) - 1$	a = 3,02, b = 17,15 x = -4,28
21. .	$a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}$ $b = 1 + \frac{z^2}{3 + z^3/5}$	x = 1,4, y = -1,2 z = 3,5
22. .	$j = \left x^{y/x} - \sqrt[3]{y/x} \right $, $\psi = (y-x) \frac{y - z/(y-x)}{1 + (y-x)^2}$	x = 1,18 y = 18,02 z = -3,2
23. .	$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}$, $\varphi = x(\sin x^3 + \cos^2 y)$	x = 0,3 y = 0,012

24 .	$y = e^{-bt} \sin(at + b) - \sqrt{ bt + a }$, $S = b \sin(at^2 \cos 2t) - 1$	a = -0,15, b = 1,07 t = 0,4
25 .	$\omega = \sqrt{x^2 + b} - b^2 \sin^3(x + a)/x$, $y = \cos^2 x^3 - \frac{x}{\sqrt{a^2 + b^2}}$	a = 1,15, b = 15,05 x = -2,19
26 .	$S = x^3 \operatorname{tg}^2(x + b)^2 + \frac{a}{\sqrt{x + b}}$, $Q = \frac{bx^2 - a}{Ax}$	a = 16,15 b = 3,24
27 .	$R = x^2(x + 1)/b - \sin^2(x + a)$, $S = \sqrt{xb/a} + \cos^2(x + b)^3$	a = 0,17, b = 0,5 x = 0,5
28 .	$y = \sin^3(x^2 + a)^2 - \sqrt{x/b}$, $Z = \frac{x^2}{A} + \cos(x + b)^2$	a = 1,11, b = 0,04 x = 0,12
29 .	$f = \sqrt[3]{m \operatorname{tg} t + c \sin t }$, $z = m \cos(bt \sin t) + c$	m = 2, c = -2 t = 1,02, b = 0,17
30	$y = b \operatorname{tg}^2 x - \frac{A}{\sin^2(x/a)}$, $y = \cos^2 x^3 - \frac{x}{\sqrt{a^2 + b^2}}$	a = 3,12, b = 17,5 x = -4,08

2) Quyidagi topshiriqlarni algoritm, blok-sxemasi va dasturini Paskal dasturlash tilida tuzing:

1	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} \\ \operatorname{tg}(x^3 + 1) + \sqrt{\sin x} \\ \operatorname{arctg}(\cos x) \end{cases}$	x < 1 1 ≤ x ≤ 2,5 x > 2,5
2	$y = \begin{cases} 3,2 + \ln x^2 + 1 \\ 10^{-3} + \operatorname{tg}(x^2 + 3,5) \\ 0,1 + \operatorname{arctg} \frac{x^3 + 1}{x^2 - 1} \end{cases}$	x ≤ 2,1 2,1 < x ≤ 4 x > 4
3	$y = \begin{cases} 1 + \sin x^2 - 2 \\ \operatorname{tg} \sqrt{x^2 + 1} + \sin x \\ \operatorname{arcsin}(\cos(x^3 + 1)) \end{cases}$	x ≤ 1,5 1,5 ≤ x ≤ 10 x > 10
4	$y = \begin{cases} \ln ax^2 + b \\ e^{\sqrt{x^3 + b}} + \lg x^3 \\ e^x + \lg(ab + 1) \end{cases}$	x ≤ a a < x ≤ b x > b
5	$y = \begin{cases} 3,2 + \ln x^2 + 1 \\ 10^{-3} + \operatorname{tg}(x^2 + 3,5) \\ 0,1 + \operatorname{arctg} \frac{x^3 + 1}{x^2 - 1} \end{cases}$	x ≤ 2,1 2,1 < x ≤ 4 x > 4
6	$y = \begin{cases} \sqrt{e^{2x-1}} + \cos x \\ \sqrt{x^2 + 1} + \sin x \\ 5,6 + \ln \frac{x}{x + 5} \end{cases}$	x < 1 1 ≤ x ≤ 2 x > 2
8	$y = \begin{cases} e^{1+ 2,6+\ln x^3 } \\ \operatorname{tg}^2(x^3 + 3,5) \\ \sin^3\left(\sqrt{x + \frac{2,7}{x}}\right)^2 + \sqrt{\cos x} \end{cases}$	x ≤ 2,6 2,6 ≤ x < 6
9	$y = \begin{cases} \operatorname{arcsin} \sqrt{x} + \cos x \\ e^{\sqrt{x^3 + ab}} + \ln^2 x \\ \operatorname{tg}(ax^3 + bx) + \sqrt{\cos x} \end{cases}$	x < a a ≤ x ≤ b x > b

10	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} \\ \operatorname{tg}(x^3 + 1) + \sqrt{\sin x} \\ \operatorname{arctg}(\cos x) \end{cases}$	$x < 3$ $3 \leq x \leq 4,5$ $x > 4,5$
11	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} \\ \operatorname{tg}(x^3 + 1) + \sqrt{\sin x} \\ \operatorname{arctg}(\cos x) \end{cases}$	$x < 4$ $4 \leq x \leq 5,5$ $x > 5,5$
12	$y = \begin{cases} 3,2 + \ln x^2 + 1 \\ 10^{-3} + \operatorname{tg}(x^2 + 3,5) \\ 0,1 + \operatorname{arctg} \frac{x^3 + 1}{x^2 - 1} \end{cases}$	$x \leq 3,1$ $3,1 < x \leq 5$ $x > 5$
13	$y = \begin{cases} 1 + \sin x^2 - 2 \\ \operatorname{tg}\sqrt{x^2 + 1} + \sin x \\ \arcsin(\cos(x^3 + 1)) \end{cases}$	$x \leq 2,5$ $2,5 \leq x \leq 12$ $x > 12$
14	$y = \begin{cases} \ln ax^2 + b \\ e^{\sqrt{x^3 + b}} + \lg x^3 \\ e^x + \lg(ab + 1) \end{cases}$	$x \leq a$ $a < x \leq b$ $x > b$
15	$y = \begin{cases} 3,2 + \ln x^2 + 1 \\ 10^{-3} + \operatorname{tg}(x^2 + 3,5) \\ 0,1 + \operatorname{arctg} \frac{x^3 + 1}{x^2 - 1} \end{cases}$	$x \leq 3,1$ $3,1 < x \leq 6$ $x > 6$
16	$y = \begin{cases} \sqrt{e^{2x-1}} + \cos x \\ \sqrt{x^2 + 1} + \sin x \\ 5,6 + \ln \frac{x}{x+5} \end{cases}$	$x < 2$ $2 \leq x \leq 4$ $x > 4$

17	$y = \begin{cases} e^{x^2-1} + \ln(x^2 + 1) \\ 0,25 \ln(x+1) + \frac{1}{x+1} \\ \frac{1}{1+ \cos x } + \sqrt{\cos x} \end{cases}$	$x < 1$ $1 \leq x \leq 2$ $x > 2$
18	$y = \begin{cases} e^{1+ 2,6+\ln x^3 } \\ \operatorname{tg}^2(x^3 + 3,5) \\ \sin^3\left(\sqrt{x} + \frac{2,7}{x}\right)^2 + \sqrt{\cos x} \end{cases}$	$x \leq 3,6$ $3,6 \leq x < 7$ $x \geq 7$
19	$y = \begin{cases} \arcsin \sqrt{x} + \cos x \\ e\sqrt{x^3 + ab} + \ln^2 x \\ \operatorname{tg}(ax^3 + bx) + \sqrt{\cos x} \end{cases}$	$x < a$ $a \leq x \leq b$ $x > b$
20	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} \\ \operatorname{tg}(x^3 + 1) + \sqrt{\sin x} \\ \operatorname{arctg}(\cos x) \end{cases}$	$x < 2$ $2 \leq x \leq 3,5$ $x > 3,5$
21	$y = \begin{cases} \sin(x^2 + 1) + \sqrt{\cos x} \\ \operatorname{tg}(x^3 + 1) + \sqrt{\sin x} \\ \operatorname{arctg}(\cos x) \end{cases}$	$x < 6$ $6 \leq x \leq 7,5$ $x > 7,5$
22	$y = \begin{cases} 3,2 + \ln x^2 + 1 \\ 10^{-3} + \operatorname{tg}(x^2 + 3,5) \\ 0,1 + \operatorname{arctg} \frac{x^3 + 1}{x^2 - 1} \end{cases}$	$x \leq 3,1$ $3,1 < x \leq 6$ $x > 6$
23	$y = \begin{cases} 1 + \sin x^2 - 2 \\ \operatorname{tg}\sqrt{x^2 + 1} + \sin x \\ \arcsin(\cos(x^3 + 1)) \end{cases}$	$x \leq 3,5$ $3,5 \leq x \leq 13$ $x > 13$

2 4	$y = \begin{cases} \ln ax^2 + b \\ e^{\sqrt{x^3+b}} + \lg x^3 \\ e^x + \lg(ab+1) \end{cases}$	$\mathbf{x \leq a}$ $\mathbf{a < x \leq b}$ $\mathbf{x > b}$	2 8	$y = \begin{cases} e^{1+ 2,6+\ln x^3 } \\ tg^2(x^3+3,5) \\ \sin^3\left(\sqrt{x+\frac{2,7}{x}}\right)^2 + \sqrt{\cos x} \end{cases}$	$\mathbf{x \leq 4,6}$ $\mathbf{4,6 \leq x < 8}$ $\mathbf{x \geq 8}$
2 5	$y = \begin{cases} 3,2 + \ln x^2+1 \\ 10^{-3} + tg(x^2+3,5) \\ 0,1 + arctg \frac{x^3+1}{x^2-1} \end{cases}$	$\mathbf{x \leq 4,1}$ $\mathbf{4,1 < x \leq 6}$ $\mathbf{x > 6}$	2 9	$y = \begin{cases} \arcsin \sqrt{x} + \cos x \\ e\sqrt{x^3+ab} + \ln^2 x \\ tg(ax^3+bx) + \sqrt{\cos x} \end{cases}$	$\mathbf{x < a}$ $\mathbf{a \leq x \leq b}$ $\mathbf{x > b}$
2 6	$y = \begin{cases} \sqrt{e^{2x-1}} + \cos x \\ \sqrt{x^2+1} + \sin x \\ 5,6 + \ln \frac{x}{x+5} \end{cases}$	$\mathbf{x < 2}$ $\mathbf{2 \leq x \leq 4}$ $\mathbf{x > 4}$	3 0	$y = \begin{cases} \sin(x^2+1) + \sqrt{\cos x} \\ tg(x^3+1) + \sqrt{\sin x} \\ arctg(\cos x) \end{cases}$	$\mathbf{x < 2}$ $\mathbf{2 \leq x \leq 4,5}$ $\mathbf{x > 4,5}$
2 7	$y = \begin{cases} e^{x^2-1} + \ln(x^2+1) \\ 0,25 \ln(x+1) + \frac{1}{x+1} \\ \frac{1}{1+ \cos x } + \sqrt{\cos x} \end{cases}$	$\mathbf{x < 0}$ $\mathbf{0 \leq x \leq 2}$ $\mathbf{x > 2}$			

Mavzu: Takrorlanuvchi jarayonlarni algoritmlari va dasturlash.

Topshiriq: a) Quyidagi topshiriqlar algoritmi, blok-sxemasini va dasturini Paskal dasturlash tilida tuzing.

№	Funksiyani qiymatini hisoblash uchun algoritmi va dastur tuzing.	Yig'indini hisoblash uchun algoritmi va dastur tuzing.	Cheksiz qatorni 0,005 aniqlikda hisoblash algoritmi va dasturini tuzing
1	$Y = a^3 \sin x^2$ $\Delta X = 0,1; -1 \leq X \leq N$	$S = \sum_{i=4}^N \frac{0,4}{i(i+1) + \sqrt{i}}$	$Z = \sum_{i=1}^{\infty} \frac{1}{i(i+1) + 2 \cos i}$
2	$Y = \sqrt{a} \cos x$ $\Delta X = 0,2; 5 \leq X \leq N$	$S = \sum_{i=1}^N \frac{\cos i}{i^2 + 5}$	$Z = \sum_{i=2}^{\infty} \frac{1}{i^2 + 5 + \ln i}$
3	$Y = a^{\frac{1}{3}} tg x$ $\Delta X = 0,1; 2 \leq X \leq N$	$S = \sum_{i=4}^N \frac{5 + e^j}{6j + 2,7}$	$Z = \sum_{i=1}^{\infty} \frac{5}{6i^4 + 2,7 + \sqrt{i}}$
4	$Y = 2 + a \ln x$ $\Delta X = 0,2; -3 \leq X \leq N$	$S = \sum_{i=5}^N \frac{1}{\ln i + 1,7}$	$Z = \sum_{i=3}^{\infty} \frac{1}{\ln i + i^2}$
5	$Y = 3,2a\sqrt{x+1}$ $\Delta X = 0,5; 1 < X < N$	$S = \sum_{j=1}^N \frac{1}{\sqrt{K} + 1,2}$	$Z = \sum_{k=1}^{\infty} \frac{4}{3k^3 + \sin K}$

6	$Y = 5e^{xa} + 2.3$ $\Delta X = 0.3; 0 \leq X \leq N$	$S = \sum_{i=1}^N \frac{1}{2e^i + 3.4}$	$Z = \sum_{i=1}^{\infty} \frac{1}{2e^i + tg^i}$
7	$Y = ax^3 + 2.4e^x$ $\Delta X = 0.4; -5 \leq X \leq N$	$S = \sum_{j=2}^N \frac{\sqrt{5.4}}{3j^3 + j^{\frac{1}{3}}}$	$Z = \sum_{k=1}^{\infty} \frac{4}{3k^3 + \sin k}$
8	$Y = e^{ax} + 2.1x$ $\Delta X = 0.2; -2 \leq x \leq N$	$S = \sum_{j=5}^N \frac{1}{1 + \ln^2 j}$	$Z = \sum_{j=1}^{\infty} \frac{8}{\sqrt{j} + j^4}$
9	$Y = tgax + \sqrt{x}$ $\Delta X = 0.3; -0.3 \leq X \leq N$	$S = \sum_{L=2}^N \frac{1}{\sqrt{5} + L^2}$	$Z = \sum_{j=1}^{\infty} \frac{\ln j}{j^6}$
10	$Y = e^{ax} - e^{ x }$ $\Delta X = 0.5; -0.5 \leq x \leq N$	$S = \sum_{i=1}^N \frac{\cos^2 i}{i^3}$	$Z = \sum_{i=1}^{\infty} \frac{3.4}{e^{2i}}$
11	$Y = \frac{a + 0.27}{x + \sqrt{1+x}}$ $\Delta X = 0.8; -0.9 \leq X \leq N$	$S = \sum_{i=1}^N \frac{(-1)^i}{i + tgi}$	$Z = \sum_{i=3}^{\infty} \frac{1}{i^i + 2.7}$
12	$Y = \frac{\sqrt{1+x+0.3}}{a^2x}$ $\Delta X = 0.3; 0 \leq x \leq N$	$S = \sum_{i=3}^N \frac{3}{(-1)^i + 2i^2}$	$Z = \sum_{j=1}^{\infty} \frac{\sqrt{j}}{1 + j^6}$
13	$Y = \sin ax + e^a$ $\Delta X = 0.9; -2 \leq X \leq N$	$S = \sum_{i=1}^N \frac{35}{\sqrt{i} + \sqrt{i}}$	$Z = \sum_{i=2}^{\infty} \frac{3}{i + 2\sqrt{i}}$
14	$Y = a + e^x + \cos x$ $\Delta X = 0.4; -0.8 \leq x \leq N$	$S = \sum_{i=2}^N \frac{e^i + 1}{i}$	$Z = \sum_{k=1}^{\infty} \frac{k}{e^k + k^2}$
15	$Y = a^2 \sqrt{\sqrt{x+1}} + 2$ $\Delta X = 0.4; 0.3 \leq X \leq N$	$S = \sum_{i=1}^N \frac{1}{ \cos i - i}$	$Z = \sum_{j=1}^{\infty} \frac{0.7}{2 + \ln k }$
16	$Y = \cos ax + \sin a$ $\Delta X = 0.3; 0.1 \leq x \leq N$	$S = \sum_{i=5}^N \frac{1}{\cos i + \sin i}$	$Z = \sum_{i=1}^{\infty} \frac{67}{i^4 + \sqrt{3i}}$
17	$Y = 2.7a^4 + \ln x^2$ $\Delta X = 0.2; -1 \leq X \leq N$	$S = \sum_{j=1}^N \frac{2.7}{\ln j^3 + 1}$	$Z = \sum_{i=6}^{\infty} \frac{ \cos i }{i^2}$
18	$Y = 35.9a^{\frac{1}{4}} \sin x^2$ $\Delta X = 0.1; 0 \leq x \leq N$	$S = \sum_{k=3}^N \frac{k + 0.4}{\sqrt{k} + \sqrt{k}}$	$Z = \sum_{j=1}^{\infty} \frac{0.4}{\sqrt{j} + 100}$
19	$Y = \sqrt{a+2} \ln x^3$ $\Delta X = 0.4; 0.02 \leq X \leq N$	$S = \sum_{k=1}^N \frac{0.999k}{k^2 + k^{0.7}}$	$Z = \sum_{i=3}^{\infty} \frac{2.4}{ i + e^{i+1}}$
20	$Y = \sqrt{a+2.1} \ln x^3$ $\Delta X = 0.4; 0.02 \leq x \leq N$	$S = \sum_{j=1}^N \frac{k+1}{k^2 + 2}$	$Z = \sum_{L=1}^{\infty} \frac{100}{L^2 + \ln L }$
21	$Y = e^a tgx + 3$ $\Delta X = 0.3; -0.05 \leq X \leq N$	$S = \sum_{i=1}^N \frac{0.4i}{i \cos i + 3i}$	$Z = \sum_{j=1}^{\infty} \frac{ \cos k + \sin k }{k}$
22	$Y = e^{ax} + 2.91x$ $\Delta X = 0.1; 1 \leq x \leq N$	$S = \sum_{i=2}^N \frac{\sqrt{i+2}}{3i}$	$Z = \sum_{j=1}^{\infty} \frac{0.9}{e^{j+2} + 1}$
23	$Y = \cos^2(2x+a)$ $\Delta X = 0.2; 0.04 \leq X \leq N$	$S = \sum_{k=1}^N \frac{\sqrt{ \cos k }}{k}$	$Z = \sum_{i=1}^{\infty} \frac{0.4}{\sqrt{i^3 + 1}}$

24	$Y = 3\ln^2(ax^2 + 3)$ $\Delta X = 0,04; 0,4 \leq x \leq N$	$S = \sum_{j=1}^N \frac{0,9}{\sqrt{j^2 + \sqrt{1+j}}}$	$Z = \sum_{i=5}^{\infty} \frac{e^{0,4}}{\ln i^2}$
25	$Y = 0,2ax^2 \operatorname{tg} \frac{x}{a}$ $\Delta X = 0,1; 0,1 \leq X \leq N$	$S = \sum_{k=1}^N \frac{k^{\frac{1}{3}}}{k^3 + \sin k}$	$Z = \sum_{i=1}^{\infty} \frac{2,4+i}{i^i + 2}$
26	$Y = ax^2 + \sin \pi x$ $\Delta X = 0,03; 0,03 \leq x \leq N$	$S = \sum_{k=2}^N \frac{(-1)^{k+1}}{k + \sqrt{0,2k}}$	$Z = \sum_{j=1}^{\infty} \frac{k}{\ln k^5}$
27	$Y = \frac{ax^2 3x + 1}{x}$ $\Delta X = 0,04; 1 \leq X \leq N$	$S = \sum_{i=1}^N \frac{i(i+2)}{i + \sqrt{i+1}}$	$Z = \sum_{k=3}^{\infty} \frac{0,5k}{k^3 + e^k}$
28	$Y = \frac{\sin x}{a\pi x} + 0,4$ $\Delta X = 0,4; 0,08 \leq X \leq N$	$S = \sum_{i=1}^N \frac{\sin(\pi i)}{i^2}$	$Z = \sum_{j=5}^{\infty} \frac{0,9}{e^{j+3} + 3,4}$
29	$Y = a^{\frac{1}{6}} e^{x^2}$ $\Delta X = 0,25; -2 \leq x \leq N$	$S = \sum_{i=3}^N \frac{\cos(\pi i)}{\pi i}$	$Z = \sum_{j=1}^{\infty} \frac{99k}{k^{\frac{1}{3}} + \sqrt{k+k^4}}$
30	$Y = \cos \pi ax + \sin \pi ax$ $\Delta X = 0,5; -3 \leq x \leq N$	$S = \sum_{i=1}^N \frac{0,45}{\cos i + \sin i}$	$Z = \sum_{j=1}^{\infty} \frac{0,49}{\sqrt[4]{k+0,27} + k}$

TAJRIBA ISHI №4

Mavzu: Massivlar. Prosedura va funksiyalar qatnashgan dasturlar algoritmlari va dasturlari.

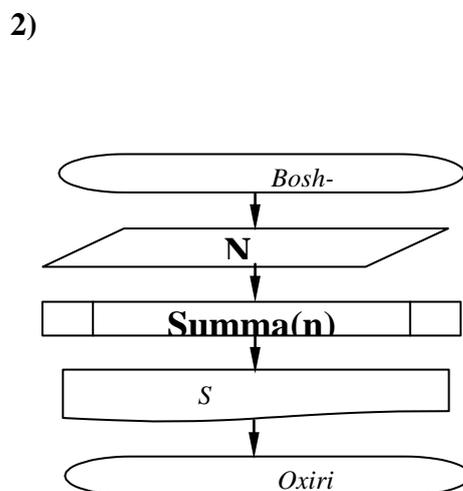
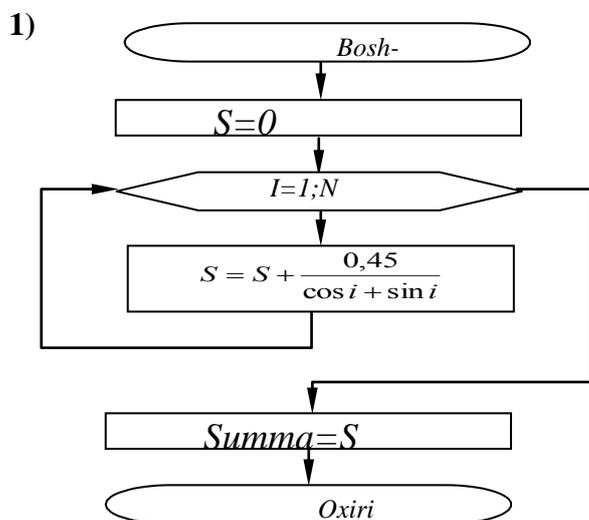
Ishdan maqsad: **Massivlar, prosedura va funksiyalar qatnashgan dasturlar algoritmlari va dasturlarini o`rganish, bilim, malaka va ko`nikmalar hosil qilish.**

1-topshiriqqa doir namuna

1) Yig`indini hisoblash uchun algoritm va dastur tuzing. Dasturlashda funksiyadan foydalaning.

$$S = \sum_{i=1}^N \frac{0,45}{\cos i + \sin i}$$

Funksiya blok-sxemasi (1) va funksiyaning asosiy blok-sxemada ishlatilishi (2)



Dasturi

```

Program perviy;
Uses crt;
Var
  n:integer;
Function summa(n:integer):real;
Var i:integer; S:real;
Begin
  S:=0;
  For I:=1 to n do
    S:=S+0.45/(cos(i)+sin(i));
  Summa:=S;
End;
Begin
  clrscr;
  write('Sikl sonini kiriting n= '); readln(n);
  Writeln('S= ', Summa(n):6:4);
  Readln
End.

```

{funksiya boshlandi}

{funksiya tugadi}

{Dastur asosiy qismi boshlanishi}

{chiqarish operatorida e'lon qilinyapti}

{Dastur asosiy qismi tugashi}

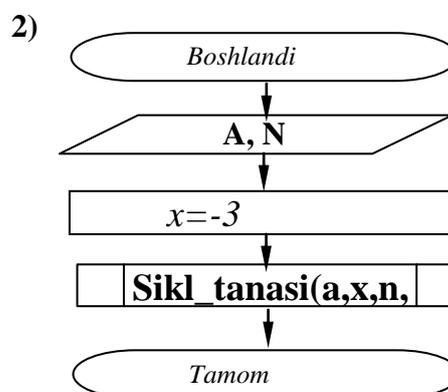
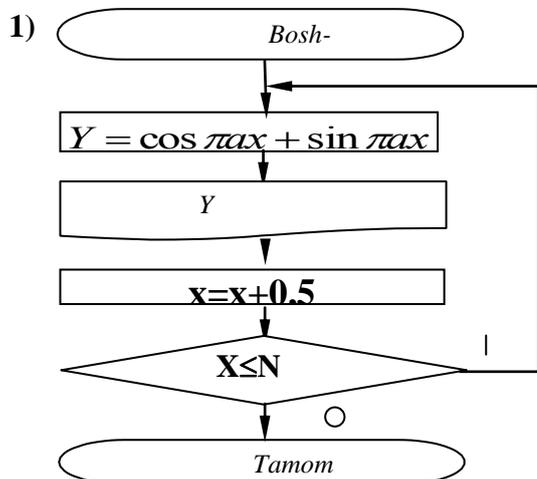
Dastur natijasi:

Sikl sonini kiriting n= 100
S= 2.5267

2-topshiriqqa doir namuna

2) Quyidagi funktsiyani qiymatini hisoblash uchun proseduradan foydalanib algoritm va dastur tuzing. $Y = \cos \pi ax + \sin \pi ax$ bunda $\Delta X = 0,5$; qadam bilan $-3 \leq x \leq N$ gacha o'zgaradi.

Prosedura blok-sxemasi (1) va proseduraning asosiy blok-sxemada ishlatilishi (2)



Dasturi

```

Program Vtoroy;
Uses crt;
Var A,n,x,y:real;
Procedure sikl_tanasi(a,x,n,y:real);
Begin
  Repeat
    Y:=cos(pi*a*x)+sin(pi*a*x);
    Writeln('x=',x:3:1,' y=',y:4:2);

```

{prosedura boshlanishi}

```

    X:=x+0.5;
    Until x>N;
End;
Begin Clrscr;
    Write('a=');Readln(a);
    Write('x=-3 ga teng, N ning qiymatini x ning boshlang`ich qiymatidan katta kiriting
n=');Readln(n);
    X:=-3;
    Sikl_tanasi(a,x,n,y);
    Readln
End.

```

{prosedura oxiri}
 {Dastur asosiy qismi}

{prosedura e`lon qilinishi}

{Dastur tugashi}

Dastur natijasi:

```

a=0.1
x=-3 ga teng, N ning qiymatini x ning boshlang`ich qiymatidan katta kiriting n=3
x=-3.0 y=-0.22
x=-2.5 y=0.00
x=-2.0 y=0.22
x=-1.5 y=0.44
x=-1.0 y=0.64
x=-0.5 y=0.83
x=0.0 y=1.00
x=0.5 y=1.14
x=1.0 y=1.26
x=1.5 y=1.34
x=2.0 y=1.40
x=2.5 y=1.41
x=3.0 y=1.40

```

II. TAJRIBA ISHI TOPSHIRIQLARI

Topshiriq: Quyidagi topshiriqlar algoritm, blok-sxemasi va dasturini Paskal dasturlash tilida tuzilsin.

a) Oldingi tajriba ishida berilgan 1 va 2 topshiriqlarni prosedura va funksiyadan foydalanib dasturini tuzilsin.

b) Massivlarni qayta ishlash algoritmi va dasturi tuzilsin. Dastur tuzishda qism dasturdan foydalanilsin.

№	Massiv nomi va o'lchami	Berilishi	Sharti va chegarasi
1.	$X(10)$	X massivning elementlar soni va yig'indisini hisoblang.	$0 \leq X_i \leq 1$
2.	$A(20)$	A massiv elementlari o'rta arifmetik qiymatini hisoblang	$A_i > 0$
3.	$X(15)$	X massiv elementlarini Y massivga ko'chirib yozing va ularni sonini chiqaring.	$-1 \leq X_i \leq 1$
4.	$B(10)$	B massivning eng katta elementini va uning tartib nomerini aniqlang.	$B_i > 0$
5.	$C(40)$	C massivning eng kichik elementi va uning nomerini aniqlang.	$C_i < 0$
6.	$D(80)$	D massivning eng katta va eng kichik elementlarini toping.	-
7.	$Y(20)$	Y massiv elementlarining o'rta geometrigini hisoblang.	$Y_i > 0$
8.	$Z(30)$	R massivda Z massivning avval musbat so'ngra manfiy elementlarini joylashtiring.	-
9.	$N(50)$	N massivni 3 ga karrali elementlarini aniqlang.	$n:/3+3=n_i$
10	$X(N)$	X massivni elementlari soni va yig'indisini hisoblang.	$x_i > 0, N \leq 30$
11	$A(10,15)$	Matrisaning har bir ustunidagi musbat elementlar soni va yig'indisini hisoblang. Natijani ikki qator ko'rinishida bosmadan chiqaring.	$a_{ij} > 0$
12	$A(N,M)$	Matrisaning har bir qatori elementlari sonini va yig'indisini hisoblang. Natijani ikki ustun ko'rinishida bosmadan chiqaring.	$N \leq 20$ $M \leq 15$
13	$B(N,N)$	Matrisaning bosh diagonalidagi va bu diagonal pastidagi elementlar soni va yig'indisini hisoblang.	$N \leq 12$
14	$C(N,N)$	Matrisaning bosh diagonalidagi musbat elementlar soni va yig'indisini hisoblang.	$C_{ij} > 0$ $N \leq 12$
15	$D(K,K)$	Matrisaning manfiy elementlari o'rniga nollarni yozing va uni bosmadan chiqaring.	$K \leq 10$
16	$D(10,10)$	Matrisaning manfiy elementlari o'rniga nollarni musbat elementlari o'rniga esa birlarni yozing.	-
17	$F(N,M)$	Matrisaning manfiy elementlari o'rniga uning kvadratini yozing va uni bosmadan chiqaring.	$N \leq 20$ $M \leq 10$
18	$F(10,8)$	Matrisaning elementlarini o'sish tartibida joylashtiring.	-
19	$N(10,10)$	Matrisaning har bir qatori elementlar sonini va yig'indisini hisoblang. Natijani ikkilik matrisa ko'rinishida ekranga va bosmaga chiqaring.	$n_{ij}/5+5=n_{ij}$
20	$N(10,10)$	Matrisaning har bir qatori elementlari sonini va yig'indisini topib ularni mos ravishda ko'paytirib ekranga va bosmaga chiqaring.	-
21	$N(10,15)$	Matrisadagi barcha haqiqiy sonlarni uning kvadrati bilan almashtiring.	-
22	$N(10,15)$	Matrisaning elementlariga birni qo'shib, uning elementlarini kamayish tartibida joylashtiring.	-

23	$N(10,15)$	Matrisaning elementlarini ishorasini o`zgartirib, uni o`shish tartibida joylashtiring.	-
24	$F(10,8)$	Matrisadagi barcha haiqiqy sonlarni kvadratiga 5 sonini qo`shib o`shish tartibida tartiblang.	-
25	$F(10,6)$	Matrisaning eng katta va eng kichik elementini topib joyini almashtiring.	-
26	$F(10,9)$	Matrisa elementlarini kvadratga ko`tarib uni kamayish tartibida joylashtiring.	-
27	$F(10,11)$	Matrisa qator elementlarini qo`shib yangi ustun matrisa hosil qiling va uni o`shish tartibida joylashtiring.	-
28	$F(12,12)$	Matrisani elementlarini yig`indisini kvadratini ustun matrisa ko`rinishida joylashtirib, uni kamayish tartibida joylashtiring.	-
29	$F(M, N)$	Matrisaning elementlarini sonini aniqlab uning har bir elementini shu songa ko`paytirib chiqing.	$N \leq 20$ $M \leq 10$
30	$F(10,11)$	Matrisani elementlari yig`indisini toping va bu sonni ushbu matrisa elementini har biriga qo`shib chiqing.	$N \leq 20$ $M \leq 10$

Фойдаланилган адабиётлар

1. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык Паскаль.-М.:Наука, 1988.-320с.
2. Абрамов С.А.,Гнезделова Капустина Е.Н.и др. Задачи по программированию. - М.: Наука, 1988.
3. Вирт Н. Алгоритмы + структуры данных программа.-М.:Мир,1985.-405с.
4. Культин Н.Б. Программирование в Turbo Pascal 7.0 и Delphi. СПб.: БХВ-Петербург, 2001.-416с.
5. Кэнту М. Delphi 5 для профессионалов.- СПб: Питер, 2001. -944 с.
6. Немнюгин С.А. Turbo pascal, учебник. Изд. Питер., 2001, -496 с.
7. Ставровский А.Б. Турбо Паскаль. 7.0 и Delphi. 2-е изд. 2001, -416с.
8. Файсман А. Профессиональное программирование на Турбо-Паскаль. Ташкент 1992.
9. Шумаков П.В.Delphi3и разработка приложений баз данных.-М.:«НОЛИДЖ»,1998.-704
10. Пилшиков В.Н. Упражнения по языку Паскаль-М.: МГУ, 1986.
11. Фаронов В.В. Програмирование в персональном ЭВМ в среде Турбо-Паскаль.-М.:МГТУ,1990. -443с.
12. Информатика фанидан амалий ва тажриба машғулотлари. Ш. Н. Носирова, Ш. Р. Ёринов. Навоий-2000
13. Ҳисоблаш математикаси ва программалаш. Абдукодиров А. А., Тошкент-Ўқитувчи 1992.
14. Численные методы для ПЭВМ на языках Бейсик, Фортран, Паскаль. Томск-МП. «Раceso» 1992
15. Ҳисоблаш усулларида машқлар ва лаборатория ишлари. А. Абдухамидов, С. Худойназаров. Тошкент. -Ўқитувчи. , 1994.
16. IBM PC шахсий компьютерларида ишлаш. С. И. Раҳмонқулова Тошкент-Шарк. , 1996.
17. Паскаль алгоритмик тили. Ш. Н. Носирова, З. У. Турсинбаева. Навоий-1999.
18. Алгоритмический язык Паскаль. Ш. Н. Носирова, З. У. Турсинбаева. Г. А. Қулахмедова. Навоий-1999.