

O'ZBEKISTON RESPUBLIKASI ALOQA VA AXBOROTLASHTIRISH AGENTLIGI

TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI

QARSHI FILIALI

«Informatika va axborot texnologiyalari» kafedrası

**“Ob`ektga yo`naltirilgan dasturlash tillari”
fanidan tajriba mashg`ulotlar uchun**

**USLUBIY KO`RSATMA
(Vizual C++)**

QARSHI – 2009 y

Tuzuvchilar:

«Informatika va axborot texnologiyalari» kafedrası

Uslubiy qo'rsatma «Informatika va axborot texnologiyalari» kafedrası majlisining __ -sonli yig'lishida muhokama qilingan va o'quv jarayonida qo'llash uchun tavsiya qilingan (Protokal N__. “__” __2009 yil).

Uslubiy qo'rsatma TATU Qarshi filiali uslubiy kengashida muhokama qilingan va o'quv jarayonida qo'llash uchun tavsiya qilingan (Protokal N__. “__” __2009 yil).

“Ob`ektga yo'naltirilgan dasturlash tillari (Visual C++)” fanidan tajriba mashg'ulotlar uchun uslubiy ko'rsatma. - Qarshi, TATU Qf, 2009. –44с.

Услубий кўрсатмада Borland C++ Builder ва Visual C++ дастурлаш тили ҳақида қисқача назарий маълумот берилган. У 5 та лаборатория ишини ўз ичига олиб, ҳар бир мавзуда 30 тадан мустақил ишлаш учун топшириқлар берилган.

Услубий кўрсатма 5140900 – “Информатика ва ахборот технологиялари” бакалаврият йўналишларининг 2-курсидаги «Объектга йуналтирилган дастурлаш тиллари» фани бўйича лаборатория машғулотида фойдаланишга мўлжалланган.

Тақризчилар:

ТАТУ ҚФ «Информатика ва ахборот технологиялари»
кафедраси

ҚМИИ “Ахборот технологиялари ва математик
моделлаштириш” кафедраси мудири, проф. А.Х. Эргашев

М у н д а р и ж а

- 1- ТАЖРИБА ИШИ. Мавзу: Чизикли дастурларни компьютерда амалга ошириш
- 2- ТАЖРИБА ИШИ. Мавзу: Тармоқланувчи дастурларни компьютерда амалга ошириш
- 3- ТАЖРИБА ИШИ. Мавзу: Такрорланувчи дастурни компьютерда амалга ошириш
- 4- ТАЖРИБА ИШИ. Мавзу: С++ дастурий муҳитида массивлар билан ишлашни амалга ошириш
- 5- ТАЖРИБА ИШИ. Мавзу: С++ муҳитида файлларни юклаш ва сақлашларни амалга ошириш

1- ТАЖРИБА ИШИ

Мавзу: Borland C++ Builder дастурлаш тилида берилган дастурий лойхаларни амалга ошириш.

Ишнинг мақсади. Borland C++ Builder дастурлаш тилида ишлаш хақида қисқача назарий маълумотлар ва математик функцияларни ечиш қуникмаларини ҳосил қилиш.

Масаланинг қўйилиши: 1). Borland C++ Builder дастурлаш тилида ишлаш хақида қисқача назарий тушунчага эга бўлиш ва математик функцияларни ечиш қуникмаларини ҳосил қилиш;

2). Берилган топширикни қуйида келтирилган режа асосида бажариш;

3) Иш юзасидан ҳисобат тузиш.

Қисқача назарий маълумотлар.

Tayanch iboralar: Borland C++ Builder ишчи соҳаси, Object Inspector, Object TreeView Form1, Caption, Icon, Object TreeView, Object Inspector, Properties, Events, VCL(Visual Component library) палетраси, Инкапсуляция, Абстракция, Ворислик, Полиморфизм..

Функциялар, формулалар. Математик индукция

Функциялар ва формулалар

Программалаш яратилаётган проектни ёзишнинг махсус усули. Бир қарашда дастурни қайта ишлаш учун компьютер хотирасига киритиш етарли, аммо унинг стили ва айниқса ўзига хос ёзилишининг аҳамияти йўқдек бўлиб кўринади. Аслида эса дастур тушунарли ва ўқиш учун осон бўлиши керак. Зарурат туғилганда уни тушуна олиш ёки унга керакли ўзгартиришлар киритиш мумкин бўлиши керак. Ҳам табиий, ҳам математик тилларда тасвирланган ифодалар воситаларидан оқилона фойдаланиш сифатли дастурлар яратишнинг етарли шартларидан ҳисобланади.

Дастур тузиш ижодий иш ҳисобланади, чунки ҳатто аниқ ифодаланган мақсадга етиш учун усулларни излаш ҳам умумий ҳолда янги билим ва қўникмаларни ишлаб чиқиш ва қўллашни талаб қилади. Баъзи хусусий ҳолларда системалироқ ва формалроқ программалаш процедураларини топиш мумкин. Агар программалаш учун қўйилган масала алгоритм кўринишида тасвирланган бўлса, у ҳолда программалаш алгоритмининг ёзилиш тилидан бажарувчи томонидан тушунарли бўлган тилга кўчиришдан иборат бўлади. Баъзи бир математик моделларда ўтказиш масаласи тугал ечилади. Алгоритмларнинг ёзувларини ўтказувчи систематик процедураларни излаш ёки дастурни масаланинг шартларидан ва қўшимча маълумотлар асосида яратиш **синтез дастури** ва унинг **дастури транслятсия қилиш** хусусий ҳолининг мавзуси (предмети) ҳисобланади.

Дастурлаш методикаси дастурнинг зарур спецификациялари усуллари асосий эътиборни қаратади, чунончи спецификацияга жойлаштирилган маълумотлардан унумли фойдаланиш дастурнинг ишончилигини оширади. Дастурнинг тўғрилигини текширишни осонлаштирувчи унинг структурасини яратиш дастурлашнинг асосий аспекти ҳисобланади.

Дастур параметрлари асосида дастурнинг асосий ҳисоблаш ишларининг усуллари тайёр формулалар асосида ҳисоблашлардан иборат бўлади. У катталиқ берилган сонлар устида бирор-бир амалларни бажарувчи формула ёрдамида тасвирланган бўлсин. Амаллар қуйидагилардан бири бўлиши мумкин:

- ✓ Арифметик амаллар (қўшиш, айириш, кўпайтириш, бўлиш, бутун бўлиш, қолдиқни ҳисоблаш);
- ✓ Математик функциялар (илдиз топиш, натурал логарифм, тригонометрик функциялар, кўрсаткичли функциялар ва бошқалар).

Масалан, қуйидаги функциянинг

$$y = e^x - (\sqrt{x} + 4) * (\sin(x) + \cos(x-1))$$

x нинг берилган баъзи қийматларидаги қийматлар жадвали ҳисоблансин.

Ҳисоблашни бошлашдан олдин бажариладиган амалларнинг кетма-кетлигини аниқлаб олиш керак бўлади. Бу функция учун бундай кетма-кетлик қуйидагича бўлади:

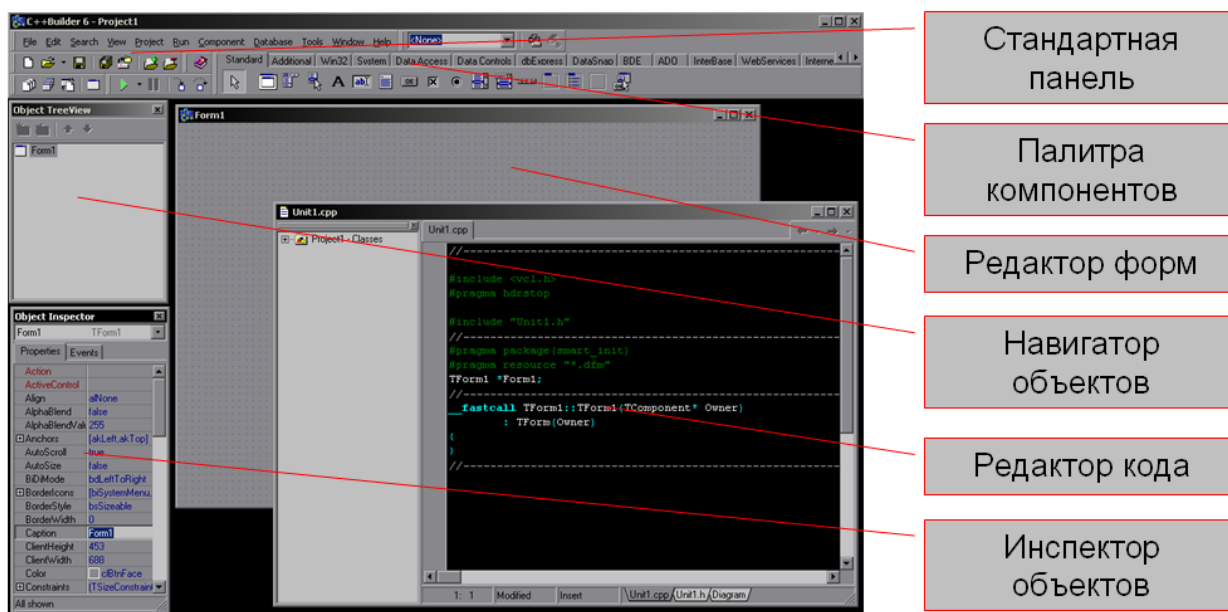
1. \sqrt{x} нинг қийматини ҳисоблаймиз.
2. Олинган натижага 4 ни қўшамиз ва йиғиндини R1 ячейкага ёзиб қўямиз.
3. $\sin(x)$ ни ҳисоблаймиз ва натижани R2 ишчи ячейкага ёзиб қўямиз.
4. $x-1$ нинг қийматини ҳисоблаймиз.

5. Вычисляем значение $\text{Cos}(x-1)$ нинг қийматини ҳисоблаймиз ва натижани R3 ишчи ячейкага ёзиб қўямиз.
6. R2 ва R3 ишчи ячейкаларнинг йиғиндисини ҳисоблаймиз.
7. Ҳосил қилинган натижани R1 ячейканинг қийматига кўпайтирамиз ва натижани R4 ячейкага ёзиб қўямиз (бу мақсадда R1 ячейкани ишлатиш ҳам мумкин, чунки унинг қиймати ишлатиб бўлинди).
8. e^x функциянинг қийматини ҳисоблаймиз.
9. Ҳосил қилинган натижадан R1 ишчи ячейканинг қийматини айирамиз.

Замонавий программалаш тилларида мураккаб ифодаларни ҳисоблаш мос турларни ўзгартириш билан регистр хотирада транслятор ёрдамида бажарилади. Лекин программалашга ўргатиш мақсадида ўргатиш ва программалашнинг биринчи босқичларида формулалар бўйича ҳисоблаш кетма-кетлигига еътибор бериш тавсия етилади. Ундан ташқари, турларнинг ўзгаришини кузатиш (маълумотлар турлари кейинроқ кўриб ўтилади), киритилган маълумотларнинг хатолигини ва мос равишда олинган натижанинг тўғри ишончли рақамлари сонини (ёки хатолигини) баҳолаш фойдалидир.

Dastur yaratish muhiti

Дастур яратиш умумлашган муҳити Редактор форм – Шакллар муҳаррири, Инспектор объектов – Объектлар инспектори, Палитра компонентов – Компонентлар палитраси, Администратор проекта – Проект администратори ва тўла умумлашган Редактор кода – Кодлар муҳаррири ҳамда кодлар ва ресурслар устидан тўлиқ назоратни таъминлайдиган , дастур иловаларини тезкор яратадиган Отладчик - инструментов - Созлаш-инструментлари кабиларни бирлаштиради.



Компонентлар

Компоненталарни шаклга ўрнатиш учун компонентлар палитрасидаги керакли пиктограмма танланади, сўнгра шаклнинг компонента жойланиши керак бўлган жойи танланади. Шундан сўнг компоненталар хоссаларини объектлар инспектори ёрдамида таҳрирлаш мумкин. Пропертиес бандида компоненталар хоссаларининг рўйхати (чапта) ва бу хоссаларнинг қийматлар рўйхати (ўнггда) жойлашган.

Компоненталар кўринадиган (визуал) ва кўринмайдиган (визуал бўлмаган) ларга бўлинади. Визуал компоненталар бажарилиш пайтида проектлаш пайтидагидек пайдо бўлади. Бунга кнопкалар ва таҳрирланувчи майдонлар мисол бўла олади. Визуал бўлмаган компоненталар проектлан вақтида шаклдаги пиктограмма кўринишида пайдо бўлади. Улар бажарилиш пайтида ҳеч қачон кўринмайди, аммо маълум функционаликга ега бўлади (масалан, берилганларга мурожатни таъминлайди, индоснинг стандарт мулоқатларини чақиради).

Object Inspector-Properties, Хоссалар

Хоссалар компоненталарнинг ташқи кўриниши ва табиатини аниқловчи атрибутлар ҳисобланади. Хоссалар устунидаги кўп хоссалар компоненталари олдиндан ўрнатилган (по

умолчанию) қийматларга ега бўлади (масалан, кнопклар баландлиги). Компоненталар хоссалари хоссалар варағи (Properties) да акс еттирилади. Объектлар инспектори компоненталарнинг нашр етилган (публишед) хоссаларини акс еттирилади. публишед-хоссалардан ташқари компоненталар умумий (public), фақат иловаларнинг бажарилиш пайтидагина мурожат қилиш мумкин бўлган нашр қилинган хоссаларга ега бўлади. Хоссалар рўйхати объектлар инспектори хоссалар варағида жойлашади. Хоссаларни проектлаш пайтида аниқлаш мумкин ёки иловаларнинг бажарилиш пайтида кўринишини ўзгартириш учун код ёзиш мумкин. Компонента хоссаларини проектлаш пайтида аниқлаш учун шаклдаги компонента танланади, объектлар инспектори хоссалари варағи очилади, аниқланадиган хосса танланади ва зурур бўлса хоссалар муҳаррири ёрдамида ўзгартирилади (бу киритиш учун оддий майдон ёки сон, осилиб тушувчи рўйхат, очилувчи рўйхат, мулоқат панели ва бошқалар бўлиши мумкин).

Бирор компонентанинг хоссаларини дастурнинг бажарилиш пайтида ўзгартириш учун «Имя Компонента» → «Название свойства» тавсифига ўзарувчидек мурожат қилиш керак, яъни қийматларни ўзимиз хоҳлагандек ўқишимиз ёки алмаштиришимиз мумкин.

Object Inspector-Events, Ходисалар

Объектлар инспекторининг ходисалар варағи (Events) компоненталар томонидан таниладиган ходисалар рўйхатини кўрсатади. Ҳар бир компонентанинг ўзининг шахсий ходисаларни қайта ишловчи наборга ега бўлади. С++ Builder да ходисаларни қайта ишловчи функцияларни ёзиш ва ходисаларни бу функция билан боғлашга тўғри келади. Бирор бир ходисага қайта ишловчи ёзиб, сиз дастурга бу ходиса рўй берганда ёзилган функциянинг бажарилишини топширасиз.

Ходисани қайта ишловчини қўшиш учун шаклда ходисани қайта ишловчи компонента танланади. Сўнгра ходисалар варағида объектлар инспектори очилиб (Event банди) ходисанинг қаторидаги қийматлар устунида сичқоннинг чап тугмаси икки марта босилади. Бу билан С++ Builder ни ходисаларни қайта ишлаш прототипини генератсия қилишга ва уни кодлар муҳарририда кўринишига мажбур қилади. Бу ҳолда бўш функция номи генератсия қилинади ва муҳаррир код киритилиши зарур бўлган жойда очилади. Курсор буйруқлар қавслари ичига жойлашади { ... }. Сўнгра ходиса содир бўлганда бажарилиши керак бўлган код киритилади. Ходисаларни қайта ишловчи функция номидан кейин кўрсатиладиган параметрларга эга бўлиши мумкин.

Қуйида ходисаларни қайта ишловчи протседуранинг шундай бўш каркаси кўрсатилган:

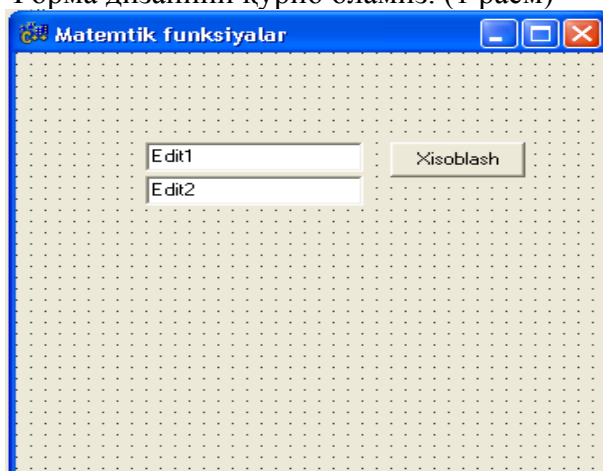
```
void __fastcall TForm1::Button1Click(TObject *Sender)
```

```
{  
  
}
```

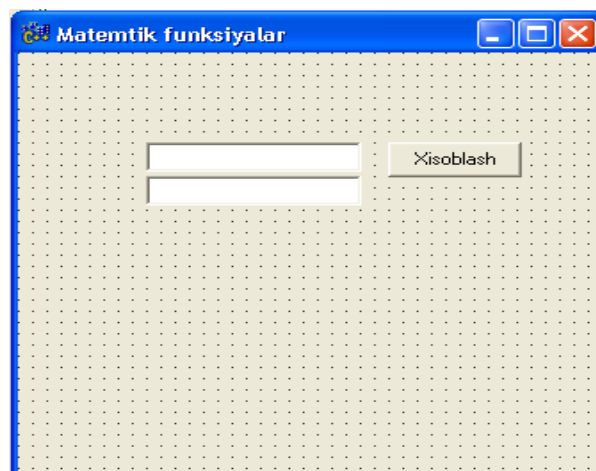
$$y = e^x - (\sqrt{x} + 4) * (\sin(x) + \cos(x-1))$$

ушбу функцияни ечишни амалга ошириш учун қуйидаги амалларни бажарамиз.

Форма дизанини қуриб оламиз. (1 расм)



1 расм



2 расм

Формада иккита қиймат киритиш ва натижа чиқариш учун TEdit библиотекасининг Edit1 (қиймат киритиш) ва Edit2 (натижа чиқариш) компоненталарини:

Ҳисоблаш амалини бажариш учун битта кнопкани TButton библиотекасининг Button1 (кнопка) компонентаси олиб форманинг керакли жойига ўрнатилади.

Edit1 ва Edit2 компоненталарининг ёзувларини учуриш учун ушбу компоненталарни белгилаб Object Inspector-Properties, (Хоссалар) бўлимининг Text сатридан Edit1 ва Edit2 ёзувларини ўчириб ташлаймиз. Кнопканинг устига “Ҳисоблаш” сўзини ёзиш учун Button1 компонентасини белгилаб Object Inspector-Properties, (Хоссалар) бўлимининг Caption сатридан Button1 ёзувининг ўрнига Ҳисоблаш ёзамиз.(2 расм)

Ҳисоблашни амалга ошириш учун Button1 компонентасининг Object Inspector-Events, Ходисалар бўлимининг OnClick сатрига сичқонча тугмасини икки марта тез босамиз Кодлар муҳарририда Қуйида ходисаларни қайта ишловчи процедуранинг шундай бўш каркаси пайдо бўлади:


```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
}
}
```

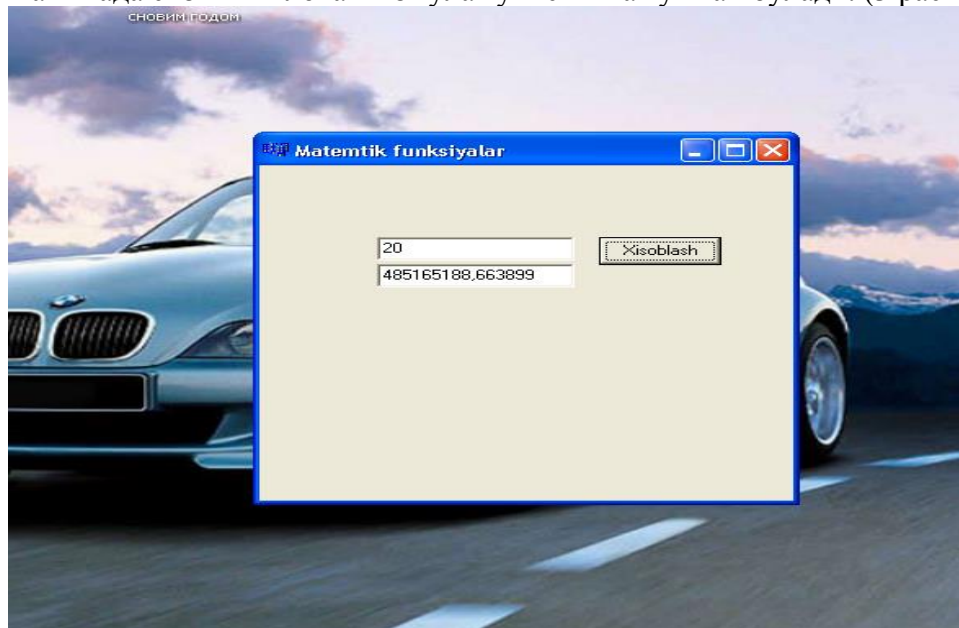
Бу жойга $y = e^x - (\sqrt{x + 4}) * (\sin(x) + \cos(x-1))$ функцияни ишлаш қаторини киритамиз. Логарифмик ва тригонаматрик функцияларни ечишни амалга ошириш учун ушбу **# include <math.h>** препроцессор директивасини эълон қилиш керак булади. Буни Кодлар муҳаррирининг энг юқори қисмига ёзамиз.

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
Edit2->Text=FloatToStr(exp(StrToFloat(Edit1->Text))-
(sqrt(StrToFloat(Edit1->Text))+4)*sin(StrToFloat(Edit1->Text))
+cos(StrToFloat(Edit1->Text)-1));
}
}
```

Бу ерда FloatToStr сонли типни матнли типга айлантириб ўқиш, StrToFloat матнли типни сонли типга айлантириб ўқиш.

Яъни бу Button1 босилганда Edit2 компонентасига натижани чиқар деган команда.

Дастурни ишга тушириш учун **F9** тугмасини ёки ускуналар панелидан  тугмаси босилади. Натижада сизнинг иловангиз тула туқис ишга тушган бўлади. (3 расм)



3 расм

Энди дастурнинг **Borland C++ Builder** алгоритмик тилидаги матнини келтирамиз.

```
#include <vcl.h>
#include <math.h>
#pragma hdrstop
#include "Unit1.h"
#include "winuser.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    Edit2->Text=FloatToStr(exp(StrToFloat(Edit1->Text))-
        (sqrt(StrToFloat(Edit1->Text))+4)*sin(StrToFloat(Edit1->Text))
        +cos(StrToFloat(Edit1->Text)-1));
}
//-----
```

Лаборатория ишини бажарилиш тартиби.

1. Ишнинг мақсадини тушиниш.
2. Масалани қуйилиши ўрганиш.
3. Масалани ечиш кетма-кетлигини аниқлаш.
4. Масалани ечиш алгоритмини тузиш.
5. Масалани дастурини тузиш.
6. C++ муҳитида дастур матнини киритиш.
7. Компьютерда натижани олиш.
8. Олинган натижани таҳлил қилиш.
9. Бажарилган иш бўйича ҳисобот тайёрлаш.

Ҳисоботнинг мазмуни

1. Ишнинг назарий асослари.
2. Масаланинг ечиш алгоритми.
3. Масалани ечиш дастури.
4. Ҳисоб натижалари ва олинган натижаларни таҳлили.

Топшириқ вариантлари:

Берилган функцияларнинг қийматининг ҳисоблаш алгоритми ва дастурини тузинг.

1. Лабоаратория ишини бажариш учун топшириқ вариантлари.

Қўйидаги ифодалар қийматини ҳисоблаш алгоритми ва дастури тузилсин.

1. $y = \sqrt{\cos x^2 + \sin^2(x^2 + 2x + 1)}$
2. $y = a \ln \left(\sqrt{1+x} + \cos^2 x \right)$
3. $y = e^{\sqrt{1+x}} - \ln|ctgx| + 0.75$
4. $z = 5x + 3x^2 \sqrt{1-x^3 \sin 2x(2x^2 + 9)}$
5. $y = e^{\sin \sqrt{x}} + \sqrt{1+x^2}$
6. $y = |x - 5cd| + e^{\sqrt{\sin x + 1}}$
7. $y = \frac{1 - x \ln|x|}{e^x}, x = \frac{(u+v)^2}{\frac{3}{4} + u^2 + v^2}$

$$8. y = \sin^2 x / (x^2 + z^2)$$

$$9. y = \frac{x^3 + e^x - \sin^2 x}{\sqrt{\cos x + |x|}}$$

$$10. y = \frac{1}{\cos x} + \ln \left| \operatorname{tg} \frac{x}{2} \right| + \frac{a + dx}{c + dx}$$

$$11. y = \frac{\sin x + a}{\cos^2 x + b^2}, a = \frac{\sqrt{x} + e^x}{\ln|x-3|}$$

$$12. z = \frac{\ln|x| + y^2}{2 \operatorname{arctg} x} - n * \operatorname{tg} x$$

$$13. y = \sqrt{a^2 + e^2} + 4\sqrt{a/b}$$

$$, a = \left[2 * \sin\left(\operatorname{arct} \frac{x}{2}\right) \right] e = x^3$$

$$14. s = \frac{e.c.\sin a}{2}, e = \sqrt{\frac{1 - \cos a}{2}}$$

$$C = \sqrt{\frac{1 + \cos A}{2}}$$

$$15. S = p(p-a) \operatorname{tg} \frac{A}{2}, p = (A+B+C)/2$$

$$, A = \sqrt{\frac{1 - \cos A}{1 + \cos B}}$$

$$16. y = 2\sqrt{x^2 + \cos x} + \frac{e^{x+3}}{2}$$

$$17. y = \frac{\ln(|x|+2)}{x^4+1} + e^{x+1}$$

$$18. y = x \cos x + \sin^2 x - \ln(|x|+1)$$

$$19. u = (1+z) * \frac{x + \frac{y}{z}}{a - \frac{1}{1-x^2}}, z = \frac{\sin^2 x}{x^2 + y^2}$$

$$20. w = \frac{1}{a^2 \sqrt{a^2 - 1}}, a = \frac{u^3 + e^4 - \sin^3 x}{\sqrt{\cos x + |x|}}$$

$$21. y = \sqrt{\frac{\sqrt{a^2 + b^2 + c^2}}{a^2 + b^2}} - \frac{1}{2a} * e^{\frac{|x-a|}{b}}$$

$$22. y = \frac{\sqrt{|\ln|x+1|+d}}{\sin a + \sin d}, a = \sqrt{\frac{3vh}{nr}}, d = \frac{1}{3} nr^2 h.$$

$$23. s = p^2 \operatorname{tg} \frac{A}{2} * \operatorname{tg} \frac{B}{2} * \operatorname{tg} \frac{C}{2}, p = (A+B+C)/2.$$

$$24. S = p(p-a) \operatorname{tg} \frac{A}{2}, p = (A+B+C)/2$$

$$, A = \sqrt{\frac{1 - \cos A}{1 + \cos B}}$$

$$25. y = e^{\sin \sqrt{x}} + \sqrt{1 + x^2}$$

$$26. y = \sin^2 x / (x^2 + z^2)$$

$$27. y = \frac{\sin x + a}{\cos^2 x + b^2}, a = \frac{\sqrt{x} + e^x}{\ln|x-3|}$$

$$28. y = \sqrt{\frac{\sqrt{a^2 + b^2 + c^2}}{a^2 + b^2}} - \frac{1}{2a} * e^{\frac{|x-a|}{b}}$$

2- ТАЖРИБА ИШИ.

Мавзу: Тармоқланувчи дастурларни компьютерда амалга ошириш

Ишнинг мақсади: Тармоқланувчи таркибли ҳисоблаш жараёнларига мос алгоритм ва дастур тизим бўйича талабаларда амалий билим ва кўникмалар ҳосил қилиш.

Масаланинг қўйилиши. Шартли ва шартсиз ўтиш Switch танлаш операторлири ёрдамида дастур тизимини ўрнатиш.

Қисқача назарий маълумотлар. Тармоқланувчи жараёнда бир неча операторлар кетма-кетлигидан шарт бажарилишига қараб битта оператор танлаб олинади. Бу кўринишдаги дастурларни тизим учун шартсиз, шартли ва танлаш операторларидан фойдаланилади.

Шартсиз ўтиш оператори. Бу операторнинг қадимий кўриниши қўйидагича бўлади: бўлганидек, мураккаб конструксиялар бўлиши мумкин.

dofo “белги”;

Шартли оператор

Дастурда шоҳланишларни ташкил қилиш учун, яъни айрим факторларга боғлиқ ҳолда турли хатти-ҳаракатлар бажарилиши учун, if оператори қўлланади.

Оператор қуйидаги форматга ега:

```
if (ифода){оператор-1;}[else {оператор-2;}]
```

if операторининг бажарилиши ифодани ҳисоблашдан бошланади. Кейин бажарилиш қуйидаги схема бўйича амалга оширилади.:

агар ифода ҳақиқий бўлса (яъни нолдан дан бошқа), бу ҳолда оператор-1 бажарилади.

агар ифода сохта бўлса (яъни 0 га тенг), бу ҳолда оператор-2 бажарилади.

агар ифода сохта бўлса-ю, оператор-2 бо‘лмаса (шарт бўлмаган конструксия квадрат қавслар ичига олинган), бу ҳолда if дан кейин турган оператор бажарилади.

Мисол:

```
If (i<j)
```

```
{
```

```
    i++;
```

```
}
```

```
Else
```

```
{
```

```
    j=i-3;
```

```
    i++;
```

```
}
```

Бу мисол яна шуни ҳам кўрсатадики, оператор-1 ўрнида, худди оператор-2 ўрнида Солиб қўйилган if операторларидан фойдаланишга ҳам йўл қўйилади. if оператори if конструксияси таркибига ёки бошқа if операторининг else конструксиясига киритилган бўлиши ҳам мумкинга.

Мисоллар:

```
int t=2;
```

```
int b=7;
```

```
int r=3;
```

```
if(t>b)
```

```
{
```

```
if(b<r)
```

```
{
```

```
    r=b;
```

```
}
```

```
}
```

```
Else
```

```
{
```

```
    r=t;
```

```
return (0);
```

```
}
```

Ушбу дастурнинг бажарилиши натижасида r=2 бўлади.

Бунда дастурнинг бажарилиши жараёни **dofo** хизматчи сўзидан кейин ёзилган белгидан давом этади. Бундан ташқари, шартсиз ўтиш операторида ишлатиладиган **белги label** хизматчи сўзиёрдамида дастур салаҳасида тасвирлаб қўйилиши лозим.

Шартли ўтиш оператори. Шартли ўтиш оператори берилган бирор шартнинг бажарилиши ёки бажарилмаслигига қараб оператордан бирини танлашга имкон беради. Бунда танлаш дастур умумий кўринишлари қўйидагича бўлади:

if (ифода){оператор-1;} **1 кўриниши**

if (ифода){оператор-1;}[else{оператор-2;}] **2 кўриниши**

- 1) Кўринишидаги оператор агар **<шарт>** ўрнидаги мантиқий ифоданинг қиймати true (паст) бўлса у ҳолда then хизматчи сўзидан кейинги операторни акс ҳолда шартли ўтиш операторидан кейинги операторга бошқаришни ўтказиш учун ишлатилади. Шартли ўтиш операторининг бу кўриниши тўлиқсиз ўтиш оператори деб аталади.
- 2) Кўринишдаги оператор шартли ўтиш операторининг тўлиқ кўриниши деб аталади. Бу оператор ишлатиладиган then ва else хизматчи сўзлардан кейин ёзилган икки оператордан бири танлаб олинади. Бунда агар **<шарт>** ўрнидаги мантиқий ифоданинг қиймати рост бўлса у ҳолда then дан кейинги оператор, акс ҳолда эса, else дан кейинги оператор бажарилади.

Танлаш оператори (Case). Маълумки, шартли ўтиш операторида берилган шартнинг бажарилиши-бажарилишмаслигига қараб икки ҳолдан биттаси яъни then дан ёки else дан кейинги оператор танланар эди.

Баъзан шундай масалалар учрайдики, шартнинг бажарилиши-бажарилмаслигига қараб, бир нечта операторлардан (икки ёки ундан ортиқ) бирини танлашга тўғри келади. Буни йўқоридаги шартли операторлардан фойдаланиб ҳам амалга ошириш мумкин. Лекин бундай дастур фойдаланувчи учун ҳам ўқитилиши ноқулай, ҳам ҳажми катта бўлади.

Паскал дастурлаш тилида уни қулай кўринишида ёзиш оператори мавжуд бўлиб, қийматига қараб тармоқланишга берилган бир неча ҳоллардан фақат биттасигина танлаб олинади. Бу операторнинг умумий кўриниши қўйидагича бўлади.

Switch <ифода>

Case <ифода>of

<Берилганлар рўйхати>: <операторлар>;

END;

Операторда ифода яни тинловчи сифатида олинган ифоданинг қиймати бутун, сатрли (**белгили**), мантиқий ёки фойдаланувчи томондан аниқланган, чегараланган ва санаб ўтиладиган турлардан бири бўлиши мумкин.

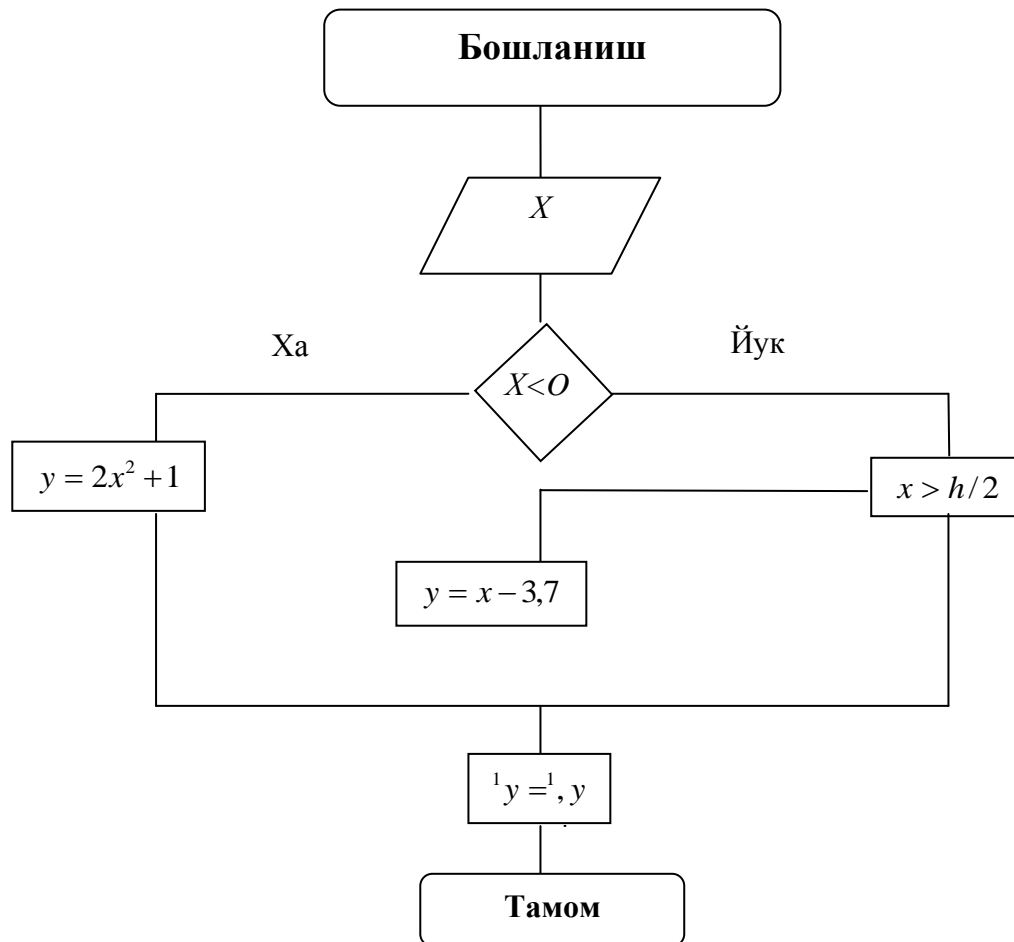
Бу оператор бажарилганда даставвал **<ифода>** қиймати аниқланади. Шундан сўнг унинг қиймати белгилар рўйхатида берилган қайси белгининг қиймати билан устма - уст тушса, шу белгили оператор бажарилади. Бу оператор бажарилиб бўлгандан кейин бошқариш шу танлаш оператордан кейинги операторда кейинги операторга узатилади. Агар ифоданинг қиймати белгиланган биронтаси билан ҳам устма-уст тушмаса у ҳолда бу оператор аниқланмаган бўлади. Белгилар операторлардан ики нукта (:) билан ажратиб ёзилади. Ифода қийматининг **тўри, белги тўри** билан бир хил бўлиши керак.

Услубий кўрсатма.

- 1- **топшириқ.** қўйдаги тармоқланувчи функцияни қийматларини ҳисобласа алгоаритми ва дастури тузилсин:

$$y = \begin{cases} 2x^2 + 1, & \text{агар } x < 0 \text{ бўлса,} \\ x - 3,7, & \text{агар } x > \frac{n}{2} \text{ бўлса,} \\ \sin x - 1, & \text{агар } 0 \leq x \leq \frac{n}{2} \text{ бўлса,} \end{cases}$$

Ечиш. 1. Қўйилган масалага мос ечиш алгоритмининг блок схемаси тузамиз.



2). Энди дастурнинг **Borland C++ Builder** алгоритмик тилидаги матнини келтирамиз.

```

//-----
#include <vcl.h>
#include <math.h>
#pragma hdrstop
#include "Unit1.h"
#include "winuser.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
if (Edit1->Text<FloatToStr(0))
{
Edit2->Text= FloatToStr (2*sqr(StrToFloat(Edit1->Text))+1);
}
else
if (Edit1->Text>FloatToStr(3.14159/2))
{
Edit2->Text= FloatToStr (StrToFloat(Edit1->Text)-3,7)
}
}
}

```

```

    }
else
    {
        Edit2->Text= FloatToStr(sin (StrToFloat(Edit1->Text))-1);
    }
}
//-----

```

Шартли буйруқ

Дастурда тармоқланишни амалга ошириш, яъни баъзи факторларга боғлиқ ҳолда турли амаллар бажарилиши учун **if** буйруғи ишлатилади.

Буйруқ қуйидаги форматга ега:

if (ифода){ 1 - оператор;} [else { 2 - оператор;}]

if буйруғининг бажарилиши ифоданинг қийматини ҳисоблашдан бошланади. Сўнгра иш қуйидаги схема асосида амалга оширилади:

- агар ифода рост бўлса (яъни 0 дан фарқли), у ҳолда 1 - оператор бажарилади.
- агар ифода ёлғон бўлса (яъни 0 га тенг), у ҳолда 2 - оператор бажарилади.
- агар ифода ёлғон ва 2 - оператор ёъқ бўлса (квадрат қавсга зарур бўлмаган конструктсия киритилади), у ҳолда иф дан кейинги буйруқ бажарилади.

Мисол:

```

if (i < j)
{
    i++;
}
else
{
    j = i-3;
    i++;
}

```

Бу мисол 1 - операторнинг ўрнида ҳам, 2 - операторнинг ўрнида ҳам мураккаб конструктсия қатнашиши мумкинлигини билдиради. Ичма-ич if буйруғини ишлатиш имконияти ҳам мавжуд. if буйруғи бошқа if буйруғининг if ёки else конструктсиялари ичида қатнашиши ҳам мумкин.

Мисоллар:

```

int t = 2;
int b = 7;
int r = 3;
if (t>b)
{
    if (b < r)
    {
        r = b;
    }
}
else
{
    r = t;
    return (0);
}

```

Бу дастур бажарилганда r нинг қиймати 2 га тенг бўлади.

Мисол.

Дастур таснифи

Масалани қўйилиши: Стандарт ўлчовли (8x8) шахмат тахтасига буғлой донлари қуйидагича Қўйилади: биринчи майдонга битта дон, кейинги ҳар бир майдонга олдинги майдонга қўйилган

доннинг икки бараварида дон қўйилади, яъни биринчи майдонга битта, иккинчи майдонга иккита, учинчига тўртта ва ҳаказо. Тахтанинг барча майдонларидаги донларнинг умумий сонини топмиг.

Зарур кўникмалар

Мазкур дастурни ёзиш учун қуйидаги кўникмаларга ега бўлиш зарур:

1. Шакллар яратиш учун камида стандарт панелнинг оддий компоненталаридан ташкил топган дастурлар яратиш муҳидан фойдаланишни билиш. Таймер системали компонентадан фойдаланишни билиш.
2. Ўзгарувчилар турларини ва уларнинг қийматлари чегарасини билиш.
3. Сонли ўзгарувчиларни сатрга ўтказувчи стандарт процедураларни билиш.
4. Шартли буйруқни ишлатишни билиш.

Муаммолар

1. Мазкур дастурнинг асосий муаммоси **unsigned __int64** туридаги сатр ўзгарувчига ўтказувчи стандарт функцияни ишлатишнинг ва шунча катта соннинг бошқа тур ўзгарувчисига саклаш имкониятининг ёқлигидадир. Бу муаммони ҳал қилиш учун **unsigned __int64** тур ўзгарувчини иккита **__int64** турига ва сўнгра мос равишда сатрга ўтказувчи стандарт функцияларни (яъни шу ном билан, аммо бошқа тур узатувчи параметрлар билан) юкловчи функция ташкил қилинган. Дастур кодида бу функция алоҳида изоҳ билан ажратиб кўрсатилган.

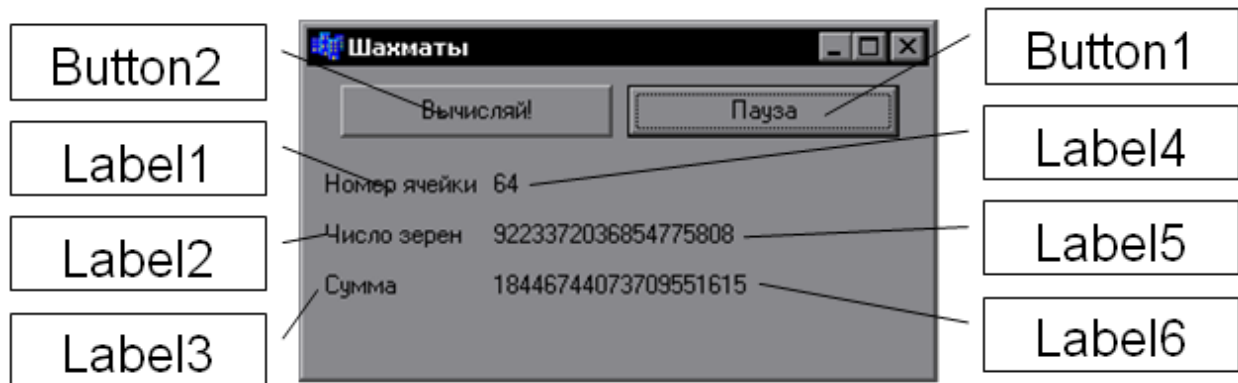
2. Мазкур дастурнинг иккинчи муаммоси қуйидагилардан иборат: агар шаклга фақат натижавий қийматларни чиқарсак, у ҳолда дастурнинг кўрсатмали ишлаши ёқолади. Бу еса бизни сикллардан фойдаланишдан маҳрум қилади.

Бу муаммони сиклларни глобал ўзгарувчилардан фойдаланиб ишлатишни симулятсия қилиш ёъли билан ҳал қилинади. Шундай қилиб, сиклнинг танаси алоҳида процедура сифатида ташкил қилинади ва бирор-бир ходисада у процедура ишлатилади, масалан, фойдаланувчи томонидан тугмачалар босилганда ёки таймернинг ҳолатларида.

Масаланинг ечими

Шакл

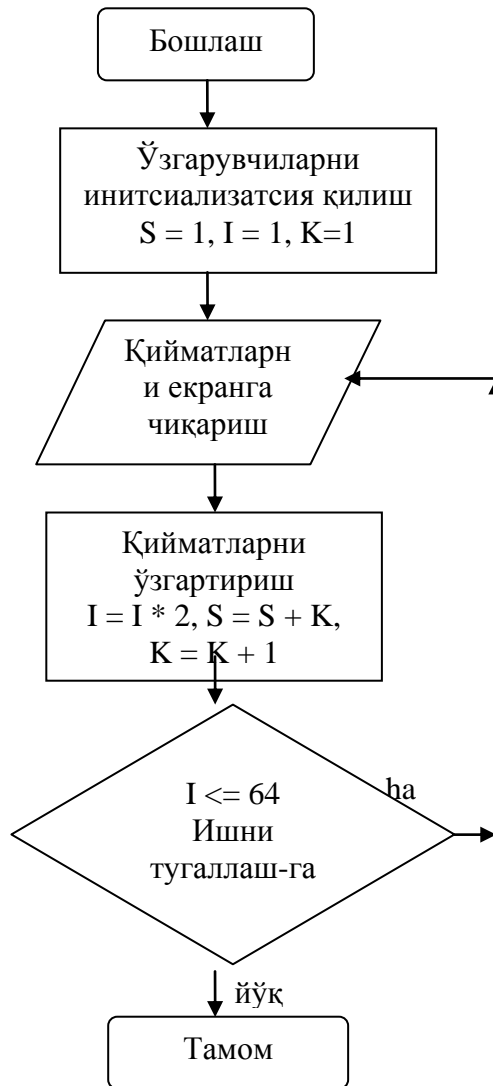
Бу дастурни амалга ошириш учун қуйидаги компоненталаришлатилади: «Метка» (Label), «Кнопка» (Button) ва «Таймер» (Timer). Биринчи иккита компонента Стандарт бандида, таймер еса System бандида жойлашган.



Натижаларни тасвирлаш учун Label синфи компоненталарининг “Caption” хоссаси қийматларини ўзгартириш зарур.

Тугмачалар учун **onClick** ходисасининг ва таймер учун **onTimer** ходисасининг ҳаракатларни ҳосил қилинади (дастур коди ёзилади).

Блок хема



Дастур коди

```

/* __int64 тип билан ишлатиш учун IntToStr функциясини оддий юклаш */
AnsiString __fastcall IntToStr(unsigned __int64 Value)
{
    __int64 k = floor(Value/100000);
    __int64 l = Value - k*100000;

    if(k!=0)
        {return IntToStr(k)+IntToStr(l);}
    else
        {return IntToStr(l);}
}
//-----
/* Глобал ўзгарувчилар*/
unsigned __int64 s = 1, i = 1;
short j = 1;
char T = 0;
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    if(j<8*8) //Майдоннинг навбатдаги ғийматини ҳисоблаш
    {
        j++;
    }
}
  
```

```

i *= 2;
s += i;
}
Label4->Caption = IntToStr(j); // Уларни шаклга чиқариш
Label5->Caption = IntToStr(i);
Label6->Caption = IntToStr(s);
}
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
T = !T; // Таймердан фойдалдннишни ўзгартириш
if(!T) //Sarlavha
{Button2->Caption = "Пуск";}
else
{Button2->Caption = "Пауза";}
}
//-----
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
If (T)
{
TForm1:Button1Click(Form1);
} // Таймернинг тики
}

```

Лаборатория ишини бажарилиш тартиби.

1. Ишнинг мақсадини тушиниш.
2. Масалани қўйилиши ўрганиш.
3. Масалани ечиш кетма-кетлигини аниқлаш.
4. Масалани ечиш алгоритмини тузиш.
5. Масалани дастурини тузиш.
6. С++ муҳитида дастур матнини киритиш.
7. Компьютерда натижани олиш.
8. Олинган натижани таҳлил қилиш.
9. Бажарилган иш бўйича ҳисобот тайёрлаш.

Ҳисоботнинг мазмуни

1. Ишнинг назарий асослари.
2. Масаланинг ечиш алгоритми.
3. Масалани ечиш дастури.
4. Ҳисоб натижалари ва олинган натижаларни таҳлили.

Топшириқ вариантлари:

Берилган тенгламалар системасининг қийматининг ҳисоблаш алгоритми ва дастурини тузинг.

1. **Лабоаратория ишини бажариш учун топшириқ вариантлари.**
Қўйидаги ифодалар қийматини ҳисоблаш алгоритми ва дастури тузилсин.

$$1. y = \begin{cases} 2x+1 & x > 2 \\ 3x; & x < 2 \\ x^2 + 3 & x = 2 \end{cases}$$

$$2. y = \begin{cases} 2,7x + 3\sqrt{x-1,2^x}, & x < 1 \\ 0,5x + \ln|x+1,2|, & 1 \leq x = 3 \\ \sqrt[3]{x^2 + tg(\sqrt[3]{x^2 + 1,2x})}, & x > 3 \end{cases}$$

$$3. y = \begin{cases} \log_3(x^2 + 4), & x < 2 \\ 4 - x^3, & x = 2 \\ \arcsin, & x < 2 \end{cases}$$

$$4. y = \begin{cases} Lnx & x > 0 \\ o & x = o \\ Ln(-x), & x < o \end{cases}$$

$$5. y = \begin{cases} 3x^2 + 4\sqrt{a-x}, & a > x \\ \sin\sqrt{3a+x}, & a = x \\ e\sqrt{x-a} & a < x, (a=2,71) \end{cases}$$

$$6. y = \begin{cases} 5/x + 3x^2 + 3x^3 + cjsx, & |a| > x \\ \sin|2a - x^2| & |a| = x \\ \sqrt[5]{x^3 + tg^2 \sqrt{x-5a}}, & |a| < x, (a=1,52) \end{cases}$$

$$7. y = \begin{cases} \sin^2 x, & |x| < n/4 \\ \sin(tgx), & |x| \geq n/4 \end{cases}$$

$$8. y = \begin{cases} 3^{x-1} & x > 1 \\ (x-1)3, & x < 1 \\ 0, & x = 1 \end{cases}$$

$$9. y = \begin{cases} \text{Log}_3 x, & x > o \\ \sqrt{-x}, & x \leq 0 \end{cases}$$

$$10. y = \begin{cases} \sqrt[3]{x^2 - 8}, & x < 0 \\ x+1 & x = o \\ \log_2 x, & x > o \end{cases}$$

$$11. y = \begin{cases} \ln|x^2 + 50|, & x < o \\ \frac{4}{5}(\sqrt[3]{x + \sqrt{x^2 + 1}}), & o \leq x \leq 15 \\ 3x^2 + \sin l^x, & x > 15 \end{cases}$$

$$12. y = \begin{cases} \ln(x-1), & x > 1 \\ 0, & x < 1 \end{cases}$$

$$13. y = \begin{cases} l^{\sin|x-1|}, & x = \frac{n}{2} \\ \cos\left|x - \frac{n}{2}\right|, & x > \frac{n}{2} \\ \sqrt{+g\left(\frac{3n}{2} * x\right)}, & x < n/2 \end{cases}$$

$$14. y = \begin{cases} \ln|x^2 + 25|, & x < 10 \\ \frac{4}{5}(\sqrt{x + \sqrt{x^2 + 1}}), & 10 \leq x \leq 15 \\ 3x^2 + \sin l^{2x}, & x > 15 \end{cases}$$

$$15. y = \begin{cases} x^2 - a^2, & |x - a| < 1 \\ x^2 + a^2, & |x - a| = 1 \\ x^2 - 2ax + 1, & |x - a| > 1 \end{cases}$$

$$16. y = \begin{cases} \operatorname{Ln}|\sin x + 1|, & x < \frac{n}{6} \\ \cos|2x + 3n|, & -\frac{n}{6} \leq \frac{n}{6} \\ \operatorname{tg}x^2 + \operatorname{arc} x, & x > \frac{n}{6} \end{cases}$$

$$17. y = \begin{cases} ax^3 + 3\operatorname{Ln}|a + x|, & |ax| < 2 \\ \sqrt{ax + \sqrt{a^2 + x^2}}, & |ax| = 2 \\ e\sqrt{ax + \sin x}, & |ax| > 2. \quad (a = 6,25) \end{cases}$$

$$18. y = \begin{cases} \sqrt{x - 3x + 4,3}, & x < 1 \\ 2\sin x + x^3 + 1, & x \geq 2 \\ \sqrt{1,5x + x^2}, & 1 \leq x < 2 \end{cases}$$

$$18. y = \begin{cases} \operatorname{arc} \operatorname{tg} \frac{x}{\sqrt{3x + x^2 + 1}}, & x = \frac{n}{2} \\ 2\sqrt{x + \sin(x - 2)}, & x = \frac{n}{2} \\ \operatorname{tg} \sqrt{x + 1}, & x > \frac{n}{2} \end{cases}$$

$$19. y = \begin{cases} 2^x + \sin|x - a|, & |a - x| < 1, \\ e\sqrt{3a + x^2 + 3}, & |\leq|a - x| \leq 2, \\ 3x(x^2 + 3,5) + \log_3 x, & |a - x| > 2 \end{cases}$$

$$20. y = \begin{cases} \sqrt{x + \cos x + 10^{-6,7}}, & x < 4, \\ \sqrt{13a - 2bx + x^2}, & 4 \leq x \leq 10 \\ a^2 - e^2 + |x| + \operatorname{Ln}x, & x > 10 \end{cases}$$

$$21. y = \begin{cases} x^2 + \sqrt[3]{x + 10^{-1,6}}, & x < -3 \\ \sin x + \operatorname{Ln}|x|, & -3 \leq x \leq -2 \\ x^2 + \sqrt{e^{-x^2} - x^2 + 2}, & x > -2 \end{cases}$$

$$22. y = \begin{cases} 2\sqrt{\sin x + \operatorname{tg}x^2}, & 1 \leq x < 3 \\ x^3 - \sqrt[6]{x + xa}, & x > 3 \\ 1 + \frac{x}{a} + abc, & x = 3 \end{cases}$$

$$23. Z = \begin{cases} \sqrt[3]{\sin(x - 10^{-1,35})}, & x < 0,3 \\ \operatorname{Ln}(1 + \left|\frac{x}{y}\right|2,5), & |x| = 0,3 \\ e^{x+y} + |x + y|, & x = 0,3 \end{cases}$$

$$24. y = \begin{cases} x^b + \sin(x + a), & x > 0, \\ x^5 = x^2 + 2x + 7, & x = 0, \\ \sqrt{\sin \frac{n}{6} + \ln|3x|}, & x < 0 \end{cases}$$

$$25. y = \begin{cases} \sin(\ln|x| + \sqrt{|x|^3 + 4,7}), & x < -3, \\ x^a + e^{|x|} + \operatorname{tg}x, & -3 \leq x = 0 \\ 0,7 \sin x + \operatorname{Log}_5 x, & x > 0 \end{cases}$$

$$26. y = \begin{cases} e^{\sqrt{1+at}} + a^{b-c}, & x > 6 \\ 1,5 * 10^{-6} - b \cdot e^{x+7}, & 4 \leq x \leq 6 \\ \operatorname{Ln}|\sin x| + \frac{e^{x+1}}{a+b}, & x < 4 \end{cases}$$

$$27. y = \begin{cases} \sin x + 2\sqrt{e^x + |x|} - 1, & X < -3 \\ \operatorname{tg}x^2 + x^{a+b} + x^3, & -3 \leq x \leq 3 \\ 2abx - 3bx^2 - b^2, & x > 3 \end{cases}$$

$$28. Z = \begin{cases} \frac{a3x^2}{(x+4)^2}, & x \leq 1 \\ \frac{x^4 + y^x}{\sqrt{1+ax}}, & 1 < x \leq 2 \\ \frac{|x-y|}{(1+x)^a}, & x > 2 \end{cases}$$

3- ТАЖРИБА ИШИ.

Мавзу: Такрорланувчи алгоритмларни C++ дастури муҳитида дастурлаштириш

Ишнинг мақсади: Такрорланувчи алгоритмларни C++ дастури муҳитида ҳисоблаш жараёнларига мос алгоритм ва дастур тизим бўйича талабаларда амалий билим ва кўникмалар ҳосил қилиш.

Масаланинг қўйилиши. for, while ва continue операторлари ёрдамида дастур структураси тизимини ўрнатиш.

Қисқача назарий маълумотлар. Кўпгина такрорланувчи элементларга мос алгоритмга мос дастур кодини ёзиш учун қуйидаги буйруқлар ёрдамида ҳосил қилинадиган циклик структураларни ишлатишга тўғри келади.

for буйруғи

for буйруғи - цикллارни ташкил қилишнинг энг умумий (оммавий) усулидир. У қуйидаги кўринишга эга:

for (1-ифода; 2-ифода; 3-ифода) {тана}.

1-ифода одатда циклни бошқарувчи ўзгарувчининг бошланғич қийматини ўрнатиш учун ишлатилади. 2-ифода цикл танаси бажарилиши керак бўлган шартни ифодалайди. 3-ифода цикл танасининг бажарилганидан кейин ўзгарувчининг ўзгаришини бошқаради.

for буйруғининг бажарилиш схемаси қуйидагича:

1. 1-ифода ҳисобланади.
2. 2-ифода ҳисобланади.
3. Агар 2-ифода нолдан фарқи (рост) бўлса, у ҳолда цикл танаси бажарилади. Сўнгра 2-ифода бажарилади ва бошқариш 2-пунктга узатилади. Агар 2-ифоданинг қиймати нол (ёлғон) бўлса, у ҳолда бошқариш фор буйруғидан кейинги буйруққа узатилади.

Шу нарса аҳамиятлики, шартни текшириш ҳар сафар цикл бошида бажарилади. Бу нарса эса бажариш шarti бошидаёқ нолга тенг бўлганда цикл танасининг бирор марта ҳам бажарилмаслигини билдиради.

Мисол:

```
int i, b;
for (i=1; i<10; i++)
{
    b=i*i;
}
```

Бу мисолда 1 дан 9 гача бўлган сонларнинг квадратлари ҳисобланади.

Баъзи ҳолларда циклни бошқарувчи бир нечта ўзгарувчиларни ишлатишнинг имконияти мавжудлиги фор буйруғининг мослашувчанлигини оширади.

Мисол:

```
int top, bot;
char string[100], temp;
for ( top=0, bot=100 ; top < bot ; top++, bot--)
{
    temp=string[top];
    string[bot]=temp;
}
```

Белгилар сатрини тескари тартибда ёзувчи бу мисолда циклни бошқариш учун иккита топ ва бот ўзгарувчилари ишлатилади. Шунини таъкидлаш лозимки, бу ерда 1- ва 2- ифодадал ўрнида кетма-кет бажарилувчи ва бергул билан адратилиб ёзилган бир нечта ифодалар ишлатилган.

фор буйруғини ишлатишнинг бошқа варианты чексиз цикл ташкил қилишдир. Бундай циклни ташкил этиш учун бўш шартли ифодаларни ишлатиш мумкин. Циклдан чиқиш учун эса одатда қўшимча шартлар ёки брек буйруғи ишлатилади (бу буйруқ кейинроқ кўрилади).

Мисол:

```
for (;)
```

```
{
    ...
    ... break;
    ...
}
```

С тилининг синтаксисига биноан буйруқ ҳам, for буйруғининг танаси ҳам бўш бўлиши мумкин. Буйруқнинг шакли излашларни ташкил етишда қўлланилиши мумкин.

Мисол:

```
for (i=0; t[i]<10 ; i++);
```

Бу мисолда цикл ўзгарувчиси бўлган и ўзгарувчи қиймати 10 дан кичик бўлмаган t массив биринчи элементи номерининг қийматини қабул қилади.

While буйруғи

while цикл буйруғи шarti олдиндан берилган цикл буйруғи дейилади ва қуйидаги кўринишга ега:

```
while (ифода) {тана};
```

Ифода сифатида С тилининг ихтиёрий ифодасини ишлатиш мумкин. Тана сифатида ихтиёрий буйруқни, жумладан бўш ва таркибли (мураккаб) буйруқларни ҳам, ишлатиш мумкин. ҳиле буйруғининг ишлаш схемаси қуйидагича:

1. Ифода ҳисобланади.
2. Агар ифода ёлғон бўлса ҳиле буйруғининг бажарилиши тугалланади ва бошқариш навбатдаги буйруққа узатилади, акс ҳолда ҳиле буйруғининг танаси бажарилади.
3. Жараён 1-пунктдан давом еттирилади.

Қуйидаги кўринишдаги цикл буйруғи

```
for ( 1-ифода; 2-ифода; 3-ифода ) {тана};
```

while буйруғи билан қуйидагича алмаштирилади:

1-ифода;

```
while (2-ифода)
```

```
{
```

```
    тана
```

```
    3-ифода;
```

```
}
```

for буйруғининг бажарилишидаги каби while буйруғида ҳам аввало шартнинг бажарилиши текширилади. Шунинг учун ҳам буйруқ танасини бажариш шарт бўлмаган ҳолларда while буйруғидан фойдаланиш қулай.

for ва while буйруқларининг ичида маълум мос турлар билан еълн қилинган локал ўзгарувчиларни ишлатиш мумкин.

do while буйруғи

do while цикл буйруғи шarti охирида берилан цикл буйруғи дейилади ва цикл танасини камида бир марта бажариш зарур бўлган ҳолларда ишлатилади. Бу буйруқ қуйидаги кўринишга ега:

```
do {тело} while (выражение);
```

do while буйруғининг бажарилиш схемаси:

1. цикл танаси бажарилади (таркибли буйруқ бўлиши ҳам мумкин).
2. Ифода ҳисобланади.
3. Агар ифода ёлғон бўлса, у ҳолда do while буйруғининг бажарилиши тугалланилади ва навбатдаги буйруқ бажарилади. Агар ифода ёлғон бўлса, у ҳолда бажариш 1-пунктдан давом еттирилади.

while va do while буйруқлари ичма-ич жойлашган бўлиши ҳам мумкин.

Мисол:

```
int i,j,k;
...
i=0; j=0; k=0;
do
{
    i++;
    j--;
    while (a[k] < i)
    {
        k++;
    }
}
while (i<30 && j<-30);
```

break буйруғи

break буйруғи бирлашган switch, do, for, while цикллардан энг ичкисининг бажарилиши тугалланилишини таъминлайди. break буйруғи бажарилгандан сўнг бошқариш бажарилиши тугалланган циклдан кейинги буйруққа узатилади.

Шу ёъл билан муддатидан аввал циклдан чиқиш таъминланади.

Continue буйруғи

continue буйруғи ҳам break буйруғи каби фақатгина цикл буйруқларининг ичида ишлатилади. Аммо ундан фарқли равишда бажариш бажарилиши тугатилган циклдан кейинги буйруқдан емас, балки бажарилиши тугалланган циклдан бошланади.

Мисол:

```
int a,b;
for (a=1,b=0; a<100; b+=a, a++)
{
    if (b%2 != 0) continue;
    ...
    /* жуфт йиғиндиларни қайта ишлаш */
}
```

Бу мисолда кўп нуқта билан белгиланган амаллар b нинг тоқ қийматларидагина бажарилади. Чунки 1 дан a гача сонлар йиғиндиси тоқ бўлганда continue буйруғи қайта ишлаш буйруқларини бажармасдан, бошқаришни for циклининг танасини навбатдаги қиймат учун бажаришга узатади. Continue буйруғи ҳам break буйруғи каби ичма-ич циклларнинг энг ичкисининг ишини тўхтатади.

2 - Мисол.

Дастурнинг таснифи

Бу масала прогрессияни тасвирловчи тенглама билан тавсифланади.

Бу масалани тасвирловчи тенглама куйидагича ёзилади:

$$y_n = \begin{cases} n & \text{n жуфт бўлганда} \\ \sum_{i=0}^n i^2 & \text{n тоқ бўлганда} \end{cases}, \quad n = \overline{0, 100}.$$

Муаммолар

Худи олдингидаги каби бу дастурда ҳам ҳисоблашлар кўрсатмали бўлиши учун энг ички цикл глобал ўзгарувчилардан фойдаланиш билан алмаштирилган. Унинг танаси еса тугмачаларни босиш ёки таймер ёрдамида чақириладиган алоҳида протседурага кўчирилган.

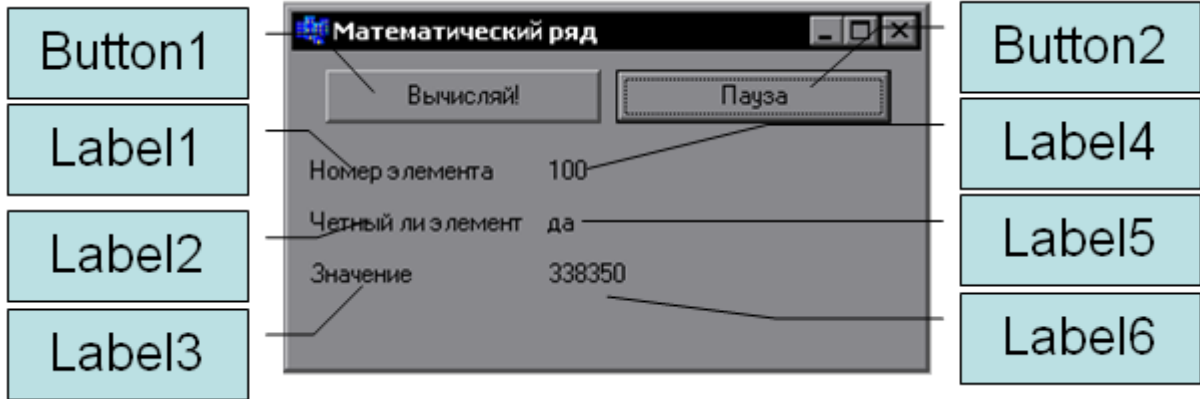
Зарур кўникмалар

Бу дастурни ёзишда аввалги дастурни ёзишда орттирилган билимлардан ташқаримураккаб ифодаларни ҳисоблаш учун циклларни ишлатишни ҳам талаб қилинади.

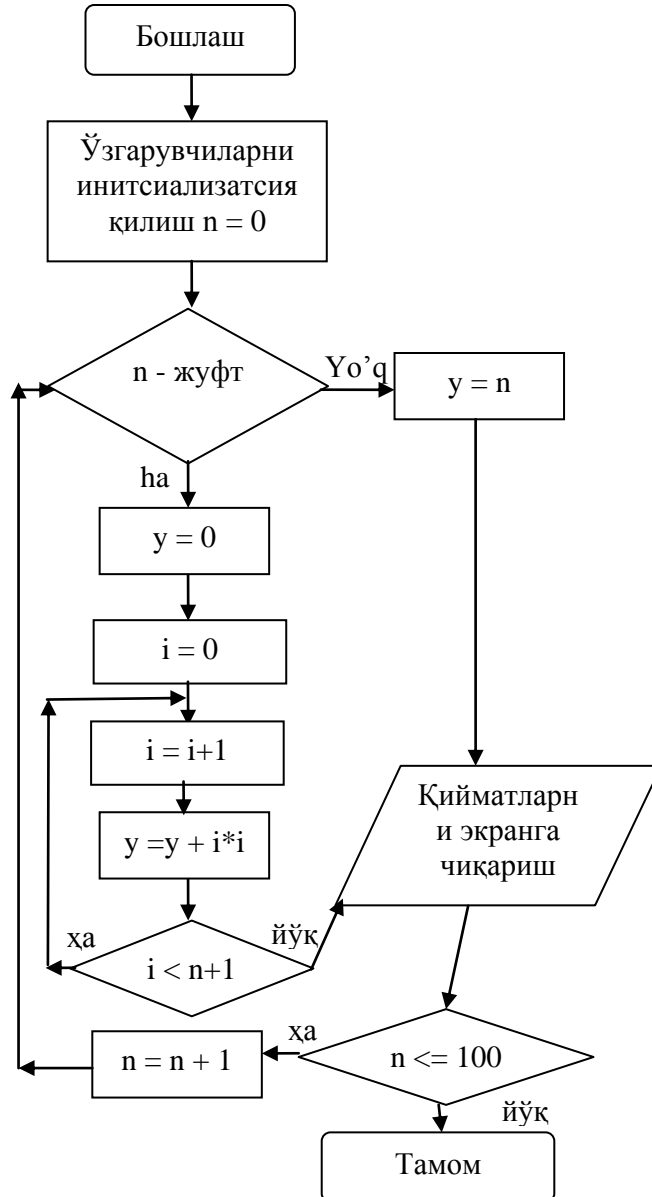
Ечиш

Шакл

Бу масалани ечиш учун зарур бўлган шакл олдинги масаланинг шаклига ўхшаш бўлади ва унда баъзи элементларнинг сарлавҳалари (Caption хоссаси) қатнашмайди холос.



Блок хема



Натурал сонлар квадратларининг йиғиндисини ҳисоблаш учун фор тсили ишлатилди.

Дастур коди

/* Глобал ўзгарувчилар*/

int n = 0, y = 0;

char T = 0;

void __fastcall TForm1::Button1Click(TObject *Sender)

{

if(n%2 == 0) // жуфтликка текшириш

{ //ha - juft

y = 0;

for(int i=0;i<n+1;i++)

{

y += i*i;

}

Label5->Caption = "ха";

}

else

{ //yo' = - toq

y = n;

Label5->Caption = "йўк";

}

Label4->Caption = IntToStr(n); // Шаклга чиқариш

Label6->Caption = IntToStr(y);

if(n<100){n++;} // номерни ошириш

}

//-----

void __fastcall TForm1::Button2Click(TObject *Sender)

{

T = !T;

if(!T) // Таймернинг Пуск/Пауза тугмаси

{Button2->Caption = "Пуск";}

else

{Button2->Caption = "Пауза";}

}

//-----

void __fastcall TForm1::Timer1Timer(TObject *Sender)

{

if(T){TForm1:Button1Click(Form1);} //Таймернинг тиги

}

Лаборатория ишини бажарилиш тартиби.

1. Ишнинг мақсадини тушиниш.
2. Масалани куйилиши ўрганиш.
3. Масалани ечиш кетма-кетлигини аниқлаш.
4. Масалани ечиш алгоритмини тузиш.
5. Масалани дастурини тузиш.
6. С++ муҳиtida дастур матнини киритиш.
7. Компьютерда натижани олиш.
8. Олинган натижани таҳлил қилиш.
9. Бажарилган иш бўйича ҳисобот тайёрлаш.

Ҳисоботнинг мазмуни

1. Ишнинг назарий асослари.
2. Масаланинг ечиш алгоритми.
3. Масалани ечиш дастури.

4. Ҳисоб натижалари ва олинган натижаларни таҳлили.

Топширик вариантлари:

Берилган қуш йиғинди, қуш купайтма, қуш йиғинди-купайтма ёки қуш купайтма-йиғиндиларнинг қийматининг ҳисоблаш алгоритми ва дастурини тузинг.

1. Лаборатория ишини бажариш учун топширик вариантлари.

Қўйдаги ифодалар қийматини ҳисоблаш алгоритми ва дастури тузилсин.

$$1. \quad S = \sum_{n=1}^{15} \frac{1}{(n+1)};$$

$$2. \quad S = \sum_{i=1}^N \frac{\sin k}{(i+1)};$$

$$3. \quad S = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + \frac{(-1)^{n+1}}{n};$$

$$4. \quad S = \frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \dots + \frac{1}{(n-1) \cdot n}, \quad n > 2;$$

$$5. \quad S = \left(1 + \frac{1}{1^2}\right) \left(1 + \frac{1}{2^2}\right) \dots \left(1 + \frac{1}{n^2}\right), \quad n = 5;$$

$$6. \quad P = \prod_{i,j=1}^{20} \frac{1}{i+j^2};$$

$$7. \quad P = \prod_{i=2}^7 \frac{i}{i+1};$$

$$8. \quad P = \prod_{k=1}^4 \prod_{n=2}^5 \frac{k(n+1)}{n(k+1)};$$

$$9. \quad S = \sum_{k=1}^3 \sum_{n=4}^9 \frac{\sqrt{n^k+1}}{\log k^n};$$

$$10. \quad y = \sum_{k=1}^n \frac{1}{(2k-1)(2k+1)};$$

$$11. \quad y = \sum_{k=1}^n \frac{2k+1}{(k+1)(k+2)};$$

$$12. \quad y = \sum_{k=1}^n \frac{2k+1}{(2k^2+1) \cdot k};$$

$$13. \quad y = \sum_{k=1}^n \frac{k+1}{k\sqrt{k^3+2}};$$

$$14. \quad y = \sum_{k=1}^n \frac{8k}{3k^2+5};$$

$$15. \quad y = \sum_{k=1}^n \frac{k^3}{2^k+3};$$

$$16. \quad y = \sum_{k=1}^n \frac{3(k+1)}{7k^3+9};$$

$$17. \quad y = \prod_{n=1}^6 \frac{n+3}{n^2+4n+1};$$

$$18. \quad y = \prod_{i=1}^7 \frac{i^4+i^2+3}{\sqrt{i^i+e^i}};$$

$$19. \quad y = \prod_{n=1}^5 \frac{3n+1}{n^2+5n+7};$$

$$20. \quad y = \prod_{n=1}^9 \frac{n^3+3n+1}{\sqrt[3]{n^2+7n+91}};$$

$$21. \quad y = \sum_{k=1}^n \frac{k+1}{\sin k + e^{k^2+1} + 1};$$

$$22. \quad y = \sum_{k=1}^n \frac{k^{k+1}}{a^{k+1} + (k+1)^a};$$

$$23. \quad y = \sum_{k=1}^n \frac{(100-k)}{\lg k + 5^{k^3}};$$

$$24. \quad y = \sum_{m=1}^n \frac{(-1)^m \sqrt{m}}{c^{2m}};$$

$$25. \quad y = \sum_{k=1}^n \frac{k^k + a^5}{\sqrt{3^k + k^3}};$$

$$26. \quad y = \prod_{n=1}^9 \frac{n+b}{n + \frac{1}{n}};$$

$$27. \quad y = \sum_{k=1}^{10} \frac{\arccos\left(\frac{k}{w}\right)}{k^k + \operatorname{tg}(k+1)};$$

$$28. \quad y = \prod_{g=1}^7 (-1)^g \frac{\cos(g^2 + 5)}{g^4 + |g - 71|};$$

$$29. \quad y = \prod_{n=1}^5 (-1)^{n+1} \frac{\operatorname{arctg} n}{n^{1.6} - \ln(n+13)};$$

30.

$$y = \prod_{i=1}^g \sum_{m=1}^2 (-1)^m \frac{(i^3 + m^4)}{\sqrt{\ln(i+m) + i^{\frac{1}{m}}}};$$

31.

$$y = \prod_{i=1}^4 \sum_{m=1}^5 \left(\frac{2 \cdot i^m + 4 \cdot i \cdot m + \operatorname{tgi}}{\sin m} - e^{-im} \right)$$

$$32. \quad y = \sum_{k=1}^7 \sum_{m=1}^4 \frac{\sqrt[4]{6k^m + m + 7k}}{\ln(m+k) + m \cdot k}$$

33.

$$y = \prod_{k=1}^8 \prod_{i=1}^4 (-1)^i \frac{\sqrt{5i^4 + e^k}}{\cos(i+k)^3 + k^i};$$

$$34. \quad y = \sum_{h=1}^7 \prod_{m=1}^5 \sqrt{\frac{k + m^3 + e^m}{\log_m^k + (mk)^3}};$$

$$35. \quad y = \prod_{i=1}^4 \sum_{m=1}^{21} \left(e^{\sqrt{i^2 + m^{i+1}}} + \frac{i^3}{m^4 + i^m} \right)$$

4- ТАЖРИБА ИШИ.

Мавзу: C++ дастурий мухитида массивлар билан ишлашни амалга ошириш

Ишнинг мақсади: C++ дастурий мухитида бир хил турдаги бир нечта ўзгарувчиларнинг тўпламидан ташкил топган ҳисоблаш жараёнларини мос алгоритм ва дастурий тизим бўйича талабаларда амалий билим ва кўникмалар ҳосил қилиш.

Масаланинг қўйилиши. Мазкур дастурнинг бош муаммоси экранга n та қаторни чиқариш заруратидан иборат. Уларнинг баъзиларининг узунликлари жуда катта бўлади. Бу муаммони ҳал қилиш учун стандарт панелнинг Мемо компоненти ишлатилган. У кўп қаторли матн майдон бўлиб, қўйилган масала учун энг мувофиқи ҳисобланади.

Қисқача назарий маълумотлар. Массив бир хил турдаги бир нечта ўзгарувчиларнинг тўпламидан ташкил топади (баъзи адабиётларда улар жадваллар деб ҳам номланади). Номи а бўлган LENGTH элементдан иборат TYPE туридаги массив қуйидагича эълон қилинади:

type a[length];

Бу буйруқда type турига тегишли махсус a[0], a[1], ..., a[length-1] номли ўзгарувчилар эълон қилинади. Массивнинг ҳар бир элементи ўз номери - индексга ега бўлади. Массивнинг x - элементига мурожат қилиш индексатсия амали ёрдамида амалга оширилади:

```
int x = ... ;           // Бутун қийматли индекс
TYPE value = a[x];     // x - элэмэнтни ўқиш
a[x] = value;         // x- элэмэнтга ёзиш
```

Индекс сифатида бутун турдаги қиймат берувчи ихтиёрий ифодани ишлатиш мумкин: char, short, int, long. С тилида массив элементларининг индекси 0 дан (1 дан емас) бошланади, LENGTH узунликдаги массивнинг охириги элементининг индекси еса LENGTH-1 (LENGTH эмас). Шунинг учун ҳам массивнинг барча элементлари бўйича цикл қуйидагича ёзилади:

```
TYPE a[LENGTH]; int indx;
for(indx=0; indx < LENGTH; indx++)
...a[indx]...;
```

Бу ерда index < LENGTH шарти index <= LENGTH-1 га тенг кучли. Массив чегарасидан чиқиш (мавжуд бўлмаган элементни ўқиш/ёзиш) қутилмаган натижаларга ва дастур ишида ҳам қутилмаган ҳолатларга олиб келиши мумкин. Бундай хатолар массивлар билан ишлашдаги энг кўп ёъл қўйиладиган хатолар ҳисобланади.

Статик массивларни унинг элементлари қийматларини {} ичида вергул билан ажратиб ёзиш, яъни инитсиализатсия қилиш йўли билан ҳам эълон қилиш мумкин. Агар массив узунлигидан кам элементлар берилган бўлса, у ҳолда қолган элементлари нол деб ҳисобланади:

```
int a10[10] = { 1, 2, 3, 4 };           // ва 6 та нол
```

Агар массивларни интсиализация қилишда унинг ўлчови берилмаса у комплятор томонидан ҳисобланади:

```
int a3[] = { 1, 2, 3 };           // Худди a3[3] каби.
```

«Паскал учбурчаги»

Дастур таснифи

Paskal учбурчаги қуйидаги жадвал кўринишида бўлади: биринчи қатор биринчи позитсияларда иккита бирдан ташкил топади, ҳар бир навбатдагиси еса биринчи позитсияда бир, бошқаларида еса олдинги қатордаги мазкур ва олдинги позитсиялардаги элементлар йиғиндиси ёрдамида ҳисобланади. Охириги элементи ҳам нол билан алмаштирилади. Шундай қилиб қуйидаги учбурчак ҳосил қилинади

```

1      1
1      2      1
1      3      3      1
1      4      6      4      1
1      5      10     10     5      1

```

Паскал учбурчаги Нютон биноми коэффитсиентларини осон ҳисоблашга ёрдам беради. Чунки Паскал учбурчаги қатори Нютон биноми ёйилмасининг қатор номерига мос коэффитсиентларидан ташкил топади.

Вазифа: Ёигирма бешинчи қаторгача Паскал учбурчаги тузилсин.

Муаммолар

Мазкур дастурнинг бош муаммоси экранга 25 та қаторни чиқариш заруратидан иборат. Уларнинг баъзиларининг узунликлари жуда катта бўлади. Бу муаммони ҳал қилиш учун стандарт панелнинг Мемо компоненти ишлатилган. У кўп қаторли матн майдон бўлиб, қўйилган масала учун енг мувофиқи ҳисобланади.

Мемо (Memo1->Lines->Add(AnsiString)) объектига тегишли Lines қисм объектининг Add протседураси матн охирига кўрсатилган қаторни қўшади.

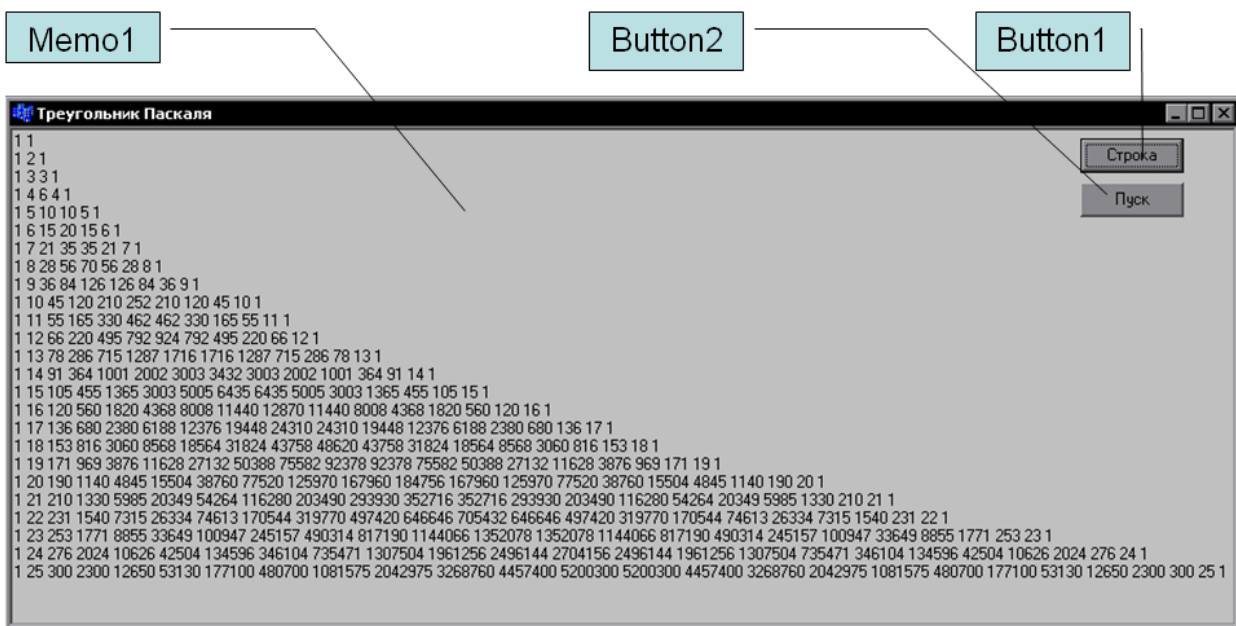
Зарур кўникмалар

Бу дастурни ёзиш учун массивлар билан бир турли маълумотлар мажмуи каби ишлашни билиш зарур. Ундан ташқари қаторлар билан ишлаш ва алоҳида ташкил етувчилардан қаторлар ҳосил қилишни ҳам билиши керак.

Ечиш

Шакл

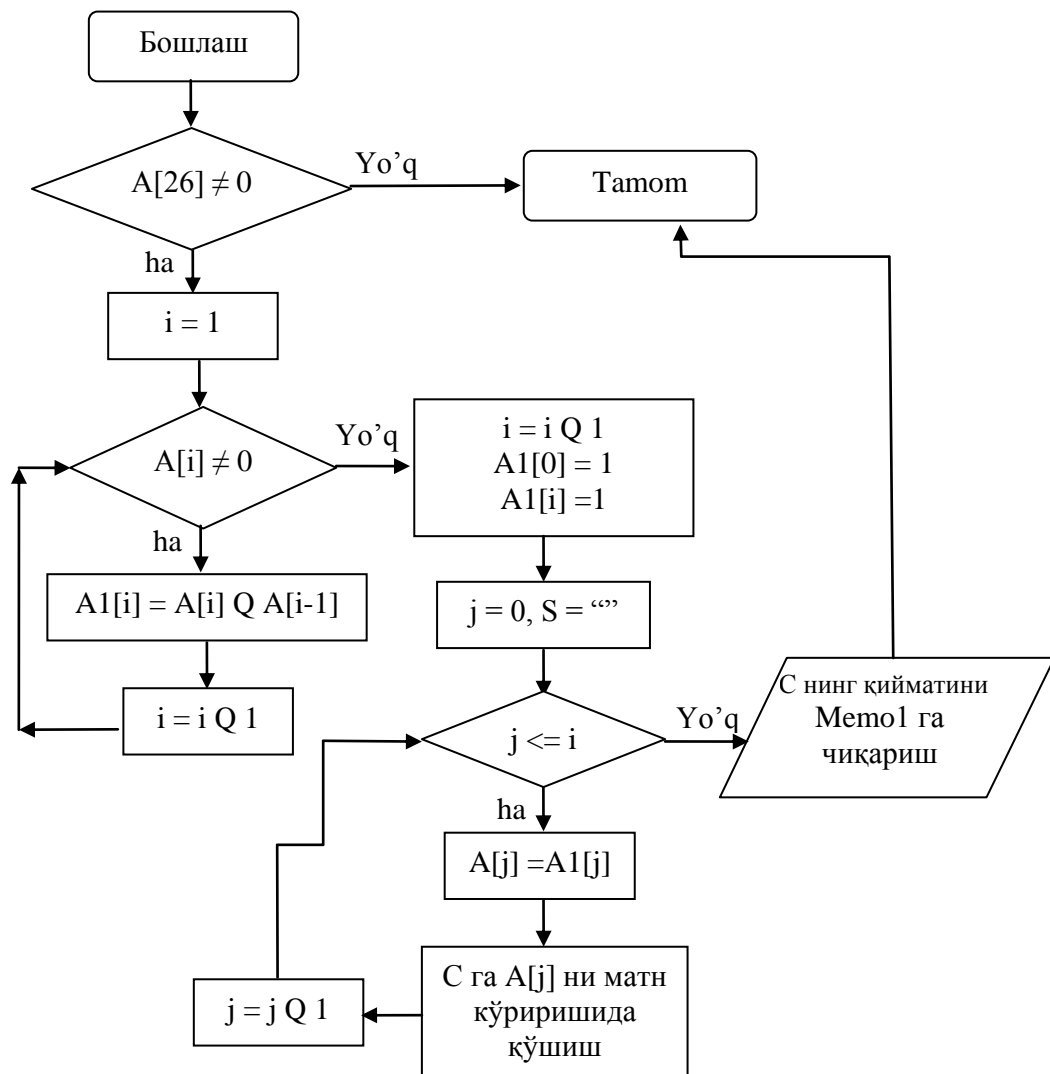
Мазкур масаланинг шакли ўлчамлари охириги қатор тўлиқ сиғадиган қилиб танланган матн майдонли битта Мемо объектдан ва иккита тугмачадан иборат. Бу тугмачалар олдинги мисолдаги ўхшаш тугмачаларнинг вазифаларини бажаради: «Строка» тугмачаси дастур алгоритмининг битта қадамини бажаради, «Пуск/Пауза» тугмачаси еса таймер ёрдамида алгоритмни бажариш учун ишга туширади.



Блок схема

Мазкур блок схема алгоритмининг битта қадамини амалга оширувчи протседурани ифодалайди. представляет протседуру, реализующую один шаг алгоритма. Тўлиқ натижага еришиш учун бу протседура 24 марта бажарилиши керак. Бунинг учун «Строка» тугмачаси кўп маротаба босилади ёки «Пуск/Пауза» тугмачаси ёрдамида ишга туширилувчи таймер алгоритмининг бажарилиши

билан амалга оширилади. Бу процедура бажарилганда А массив масаланинг биринчи каторини ҳосил қилади, яъни массивнинг биринчи ва иккинчи элементлари бир, бошқалари ноллар. Ундан ташқари дастур ўлчами билан А билан устма-уст тушадиган ёрдамчи А1 массивни ҳосил қилади.



Дастур коди

```
int A[26]; // массив
```

```
//-----
```

```
void __fastcall TForm1::FormCreate(TObject *Sender)
```

```
{
for(int i = 0; i<26; i++)
```

```
{
A[i] = 0;
```

```
}
```

```
A[0] = 1;
```

```
A[1] = 1; // массивни инициализация қилиш
```

```
}
```

```
//-----
```

```
void __fastcall TForm1::Button1Click(TObject *Sender)
```

```
{
if(A[25] == 0)
```

```
{
int A1[26];
AnsiString s = '';
```

```

int i = 1;
A1[0] = 1;
while(A[i] != 0)
{
    A1[i] = A[i] + A[i-1];
    i++;
}
A1[i] = 1;

for(int j = 0; j <= i; j++)
{
    A[j] = A1[j];
    s = s + IntToStr(A[j]) + ' ';
}
Memo1->Lines->Add(s);
}
//асосий процедура
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    if(Button2->Caption == "Пуск")
    {
        Button2->Caption = "Пауза";
    }
    else
    {
        Button2->Caption = "Пуск";
    } // taymerni aktivlashtirish
}
//-----
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    if(Button2->Caption == "Pauza")//активликка текшириш
    {
        Button1->Click(); // асосий процедурани чақириш
    }
}

```

Лаборатория ишини бажарилиш тартиби.

1. Ишнинг мақсадини тушиниш.
2. Масалани куйилиши ўрганиш.
3. Масалани ечиш кетма-кетлигини аниқлаш.
4. Масалани ечиш алгоритмини тузиш.
5. Масалани дастурини тузиш.
6. С++ муҳитида дастур матнини киритиш.
7. Компьютерда натижани олиш.
8. Олинган натижани таҳлил қилиш.
9. Бажарилган иш бўйича ҳисобот тайёрлаш.

Ҳисоботнинг мазмуни

1. Ишнинг назарий асослари.
2. Масаланинг ечиш алгоритми.
3. Масалани ечиш дастури.
4. Ҳисоб натижалари ва олинган натижаларни таҳлили.

Топширик вариантлари:

Берилган масаланинг шартига мос ҳисоблаш алгоритми ва дастурини тузинг.

1. Лаборатория ишини бажариш учун топширик вариантлари.

Қўйидаги ифодалар қийматини ҳисоблаш алгоритми ва дастури тузилсин.

Вариант рақами	М А С А Л А Ш А Р Т И
1.	<p>А) $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Унинг ҳамма манфий қийматларини квадратлари билан алмаштиринг.</p> <p>В) Клавиатурадан сатрлар бўйича икки ўлчовли массивни киритувчи ва унинг устунлар бўйича йиғиндисини топувчи дастур тузинг.</p>
2.	<p>А) $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Унинг нолмас элементлари миқдорини ҳисобланг.</p> <p>В) Иккинчи тартибли квадрат матрица аниқловчисини ҳисобловчи дастур ёзинг.</p>
3.	<p>А) $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Шу вектор элементларидан (c,d) оралиқда ётган элементлар йиғиндиси ва миқдорини топиш дастурини тузинг.</p> <p>В) Ҳақиқий қийматли $A=\{a_{ij}\}, (i,j=1,2,\dots,n)$ матрица берилган.</p> $S = \sum_{i=1}^n \sum_{j=1}^n a_{ij}^2$ <p>ни ҳисобланг.</p>
4.	<p>А) Клавиатурадан n та бутун сондан иборат бир ўлчовли массивни киритувчи ва ҳар бир элементни киритишдан олдин экранда тартиб рақами билан белги изоҳини чиқарувчи дастур тузинг.</p> <p>В) Ҳақиқий қийматли $A=\{a_{ij}\}, (i,j=1,2,\dots,n)$ матрица берилган, $c \leq a_{ij} \leq d$ шартни қаноатлантирувчи элементлар йиғиндисини топиш дастурини тузинг.</p>
5.	<p>А) Клавиатурадан киритилган n та бутун сонлар массивидан энг кичик қийматлисини топиш дастурини тузинг.</p> <p>В) Ҳақиқий қийматли $A=\{a_{ij}\}, (i,j=1,2,\dots,n)$ матрица берилган. Янги $B=\{b_{ij}\}=a_{ij}/S$ матрицани тузинг, бу ерда $S = \sum_{i=1}^n a_{ij}$.</p>
6.	<p>А) Куннинг ҳафталик ўртача ҳароратини ҳисобловчи дастурни ёзинг. Экрanga ҳафтанинг ҳар бир куни ҳароратини билдирувчи матн ҳам чиқсин.</p> <p>В) Ҳақиқий қийматли $A=\{a_{ij}\}, (i,j=1,2,\dots,n)$ матрица берилган. Ҳар бир i - элементи матрицанинг i - сатр элементлари йиғиндисидан иборат Z векторни ҳосил қилинг.</p>
7.	<p>А) Массив элементлари ичида клавиатурадан киритилган сон бор-йўқлигини текширувчи дастур тузинг. Агар бўлса, нечта?</p> <p>В) Ҳақиқий қийматли $A=\{a_{ij}\}, (i,j=1,2,\dots,n)$ матрица берилган. Ҳар бир i - элементи матрицанинг j - устун элементлари йиғиндисидан иборат Z векторни ҳосил қилинг.</p>
8.	<p>А) $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор ва X ҳақиқий сон берилган. Вектор элементларини тартибга солинг ва соннинг элементларнинг қайси оралиғида ётишини аниқланг.</p> <p>В) Ҳақиқий қийматли $A=\{a_{ij}\}, (i,j=1,2,\dots,n)$ матрица берилган. Ҳар бир i -</p>

	элементи матрицанинг i -диагонал элементлари йиғиндисидан иборат Z векторни ҳосил қилинг.
9.	А) Клавитурадан киритилган сон массивда нечанчи ўринда туришини аниқловчи дастур тузинг. Агар бундай сон бўлмаса, экранга тегишли ахборот чиқсин. Б) Ҳақиқий қийматли $A = \{a_{ij}\}$, ($i, j = 1, 2, \dots, n$) матрица берилган. Мусбат ва манфий элементлар миқдорини ҳисобланг.
10.	А) $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Шу векторнинг мусбат элементларини 0,5 га камайтириб боринг. Б) Ҳақиқий қийматли $A = \{a_{ij}\}$, ($i, j = 1, 2, \dots, n$) матрица берилган. Манфий ва мусбат элементлар йиғиндисини ҳисобланг.
11.	А) $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Шу векторда нечта мусбат, нечта манфий сон ва нечта ноль борлигини аниқланг. Б) Ҳақиқий қийматли $A = \{a_{ij}\}$, ($i, j = 1, 2, \dots, n$) матрица берилган. Ҳамма манфий элементларни уларнинг квадратлари, мусбат элементларни иккиланган қийматлари билан алмаштиринг.
12.	А) $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Массив элементлари ичидан энг катта қийматлисини топиш дастурини тузинг. Б) Ҳақиқий қийматли $A = \{a_{ij}\}$, ($i, j = 1, 2, \dots, n$) матрица берилган. Асосий диагоналдан юқорида жойлашган ва асосий диагоналдан пастда жойлашган элементлар йиғиндисини топинг.
13.	А) $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Векторнинг манфий элементигача бўлган элементлар миқдорини ҳисобланг. Б) Ҳақиқий қийматли $A = \{a_{ij}\}$, ($i, j = 1, 2, \dots, n$) матрица берилган. Ундан $b\{b_1, b_2, \dots, b_n\}$ векторни ҳосил қилинг. b_i асосий диагонал элементига тенг деб, агар $a_{ii} < 0$ бўлса, b_i шу манфий сон квадратига тенг деб олинади.
14.	А) $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. «Y» ўзгарувчига a векторнинг ҳамма мусбат элементлари йиғиндисининг иккиланган қийматини ўзлаштиринг. Б) Ҳақиқий қийматли $A = \{a_{ij}\}$, ($i, j = 1, 2, \dots, n$) матрица берилган. Асосий диагонал элементи манфий бўлган сатр элементлари йиғиндисини топинг.
15.	А) $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Вектор тоқ рақамли элементларини 5 га, жуфт рақамли элементларини 3 га кўпайтириб, уларнинг йиғиндисини топинг. Б) Ҳақиқий қийматли $A = \{a_{ij}\}$, ($i, j = 1, 2, \dots, n$) матрица берилган. Диагонал элементи манфий бўлган сатр элементлари кўпайтмаларини топинг.
16.	А) $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор берилган. Унинг манфий элементларини (векторда жойлашиш тартибини бузмасдан) биринчи, манфий бўлмаган элементларини эса кейинги ўринларда жойлаштириб, янги $b\{b_1, b_2, \dots, b_n\}$ вектор ҳосил қилинг. Б) Ҳақиқий қийматли $A = \{a_{ij}\}$, ($i, j = 1, 2, \dots, n$) матрица берилган. Асосий диагонал элементгача бўлган сатр ҳадларини шу сатр диагонал элементига кўшинг.
17.	А) Ўсиб бориш тарзида тартибга солинган 2 массивни бирлаштирувчи дастурни ёзинг. Б) Ҳақиқий қийматли $A = \{a_{ij}\}$, ($i, j = 1, 2, \dots, n$) матрица берилган. Агар матрица сатри манфий сон билан бошланса, шу сатр элементларини уларнинг квадратлари билан алмаштиринг.
18.	А) Гуруҳда бўйи ўртачадан баланд бўлган талабалар сонини аниқлаш дастурини тузинг. Б) A ва B тенг ўлчамли матрицалар берилган. Элементлари $c_{ij} = a_{ij} - b_{ij}$ формула билан ҳосил бўлувчи C матрицани тузинг.
19.	А) $a\{a_1, a_2, \dots, a_n\}$ ҳақиқий қийматли вектор ва X ҳақиқий сон берилган.

	<p>$a_i > X$ шартни каноатлантирувчи сонларнинг ўрта геометригини ҳисобланг.</p> <p>Б) A ва B тенг ўлчамли матрицалар берилган. Қуйидаги шарт бўйича C матрицани ҳосил қилинг: $c_{ij} = a_{ij} + b_{ij}$</p>
20.	<p>А) $a \{ a_1, a_2, \dots, a_n \}$ ҳақиқий қийматли вектор берилган. Массивнинг энг кичик элементи ва унинг тартиб рақамини топинг.</p> <p>Б) Ҳақиқий қийматли $A = \{a_{ij}\}$, $(i, j = 1, 2, \dots, n)$ матрица берилган. Энг катта қийматга эга бўлган сатрдан кейинги ҳамма манфий элементларни уларнинг квадратлари билан алмаштиринг.</p>
21.	<p>А) $a \{ a_1, a_2, \dots, a_n \}$ ҳақиқий қийматли вектор берилган. Даражага кўтариш амалидан фойдаланмасдан қуйидагини ҳисобланг: $Y = \sum_{i=1}^n ((-1)^i * a_i)$</p> <p>Б) Ҳақиқий қийматли $A = \{a_{ij}\}$, $(i, j = 1, 2, \dots, n)$ матрица берилган. Ҳар бир устуннинг энг катта элементини топинг ва диагонал элементларни у билан алмаштиринг.</p>
22.	<p>А) $a \{ a_1, a_2, \dots, a_n \}$ ҳақиқий қийматли вектор берилган. $S = \sum_{i=1}^n (a_i + a_i)$ ни ҳисобланг.</p> <p>Б) Ҳақиқий қийматли $A = \{a_{ij}\}$, $(i, j = 1, 2, \dots, n)$ матрица берилган. $v \{ v_1, v_2, \dots, v_n \}$ ҳақиқий қийматли векторни ҳосил қилинг. Агар $a_{ii} < 0$ бўлса, векторнинг элементи сифатида шу элементнинг квадратини қабул қилинг, агар $a_{ii} \geq 0$ бўлса, шу элементни ўзгаришсиз векторнинг элементи сифатида қабул қилинг.</p>
23.	<p>А) $a \{ a_1, a_2, \dots, a_n \}$ ҳақиқий қийматли вектор берилган. Ундан $b \{ a_n, a_{n-1}, \dots, a_1 \}$ ни ҳосил қилинг.</p> <p>Б) Ҳақиқий қийматли $A = \{a_{ij}\}$, $(i, j = 1, 2, \dots, n)$ матрица берилган. $Y = x_1 x_n + x_2 x_{n-1} + \dots + x_n x_1$ ни ҳосил қилинг. Бу ерда x_k A матрицанинг k-сатридаги энг катта элементи.</p>
24.	<p>А) $a \{ a_1, a_2, \dots, a_n \}$ ҳақиқий қийматли вектор ва X ҳақиқий сон берилган. Соннинг вектор элементлари орасида қайсиси (v, c) ораликда ётишини аниқланг.</p> <p>Б) Ҳақиқий қийматли $A = \{a_{ij}\}$, $(i, j = 1, 2, \dots, n)$ матрица берилган. Y ўзгарувчига матрицанинг энг катта элементи ётган сатр ва устуннинг скаляр кўпайтмаси қийматини беринг.</p>
25.	<p>А) $a \{ a_1, a_2, \dots, a_n \}$ ҳақиқий қийматли вектор ва X ҳақиқий қийматли сон берилган. Вектор элементларидан нечтаси шу сонга тенг бўлишини аниқланг.</p> <p>Б) Ҳақиқий қийматли $A = \{a_{ij}\}$, $(i, j = 1, 2, \dots, n)$ матрица берилган. Энг катта элемент ётган сатр ва устун тартиб рақамини топинг.</p>

5- ТАЖРИБА ИШИ.

Мавзу: C++ муҳитида файлларни юклаш ва сақлашларни амалга ошириш

Ишнинг мақсади. Borland C++ Builder дастурлаш тилида Windows операцион тизимида файлларни излаш учун файлларни сақлаш ва очиш универсал мулоқат ойналари назарда тутилган. Dialogs бандининг мос компоненталарини шаклда ўрнатиб улардан фойдаланишни ўрганиш.

Масаланинг қўйилиши: Матн файлларини ташкил қилувчи ва ўзгартирувчи дастур тузилсин.

Дискдаги файлларни очиш ва киритилган ўзгартиришларни сақлаш имкониятлари амалга оширилсин. Файлларни очиш/сақлаш стандарт мулоқатлардан фойдаланилсин.

Қисқача назарий маълумотлар: C++ да файллар билан ишлаш *fstream* кутубхонасининг баъзи синфлари ёрдамида амалга оширилади. *fstream* кутубхонаси файлларни ўқишга мўлжалланган *ifstream* ва маълумотларни файлга ёзишга мўлжалланган *ofstream* синфларига эга.

Бирор файлни ўқиш ёки ёзиш мақсадида очиш учун мос равишда *ofstream* ёки *ifstream* туридаги ўзгарувчини ҳосил қилиш керак бўлади. Бундай ўзгарувчини инициализатсия қилишда файл номи ўзгарувчининг номидан кейин қавс ичида белгилар массиви кўринишида берилади.

Масалан, C дискда жойлашган 'text.txt' файлини очиш керак. Бунинг учун қуйидаги фрагмент коди ишлатилади:

```
ifstream ifl ("C:\text.txt");           // o'qish uchun  
ofstream ofl ("C:\text.txt");           // yozish uchun
```

Бу ерда *ifl* ва *ofl* файл билан маълумотлар алмашувчи ўзгарувчиларнинг номлари. Агар файл бажарилувчи файл сақланаётган папкада жойлашган бўлса, у ҳолда файл номи тўлиқ кўрсатилмаслиги мумкин (унгача тўлиқ ёълсиз, фақат номи). Ундан ташқари, файл номини тўғридан-тўғри кўрсатишнинг ўрнига шу номни сақловчи белгилар массивини ишлатиш мумкин.

Мисол:

```
char s[20] = "C:\text.txt";  
ifstream ifl (s);
```

Маълумотларни файлга ёзиш учун пут буйруғидан фойдаланиш мумкин. У стандарт турдаги якка ўзгарувчини ёки бирор белгилар массивини узатади. Белгилар массивини узатиш зарурати туғилганда массивдаги белгилар сони ҳам узатилади.

```
ofstream ofl ("C:\text.txt");  
char s[9] = "The text";  
ofl.put(s,9);  
int i = 100;  
ofl.put(i);
```

пут функциясини чақиритишни "<<" буйруғини бажариш билан алмаштириш мумкин.

```
ofstream ofl ("C:\text.txt");  
ofl << "The text";  
int i = 100;  
ofl << i;
```

Бу буйруқни ҳар хил тур қийматларини ёки ҳар хил ўзгарувчилардан узатиб, коднинг бир қаторида кўп мартаба ишлатиш мумкин.

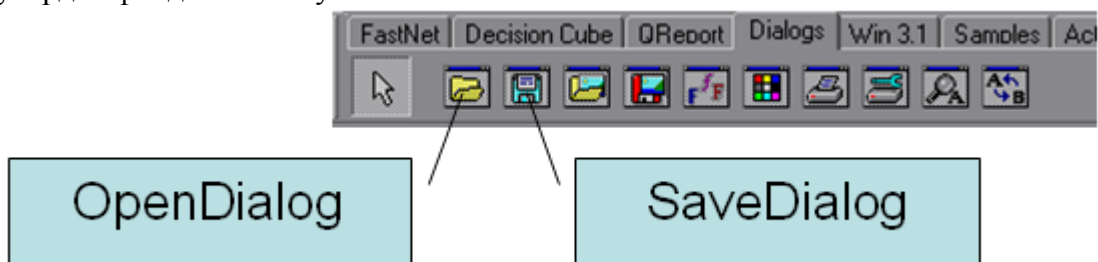
```
ofstream ofl ("C:\text.txt");  
char s[9] = "The text";  
int i = 100;  
ofl << "The text" << i << s << 200;
```

Сатр билан ишлаганда чиқариш сатр охири белгиси - 'ғн' пайдо бўлгунча давом еттирилади. Белги бўлмаган турга тегишли барча ўзгарувчилар олдиндан белги кўринишига ўтказилади.

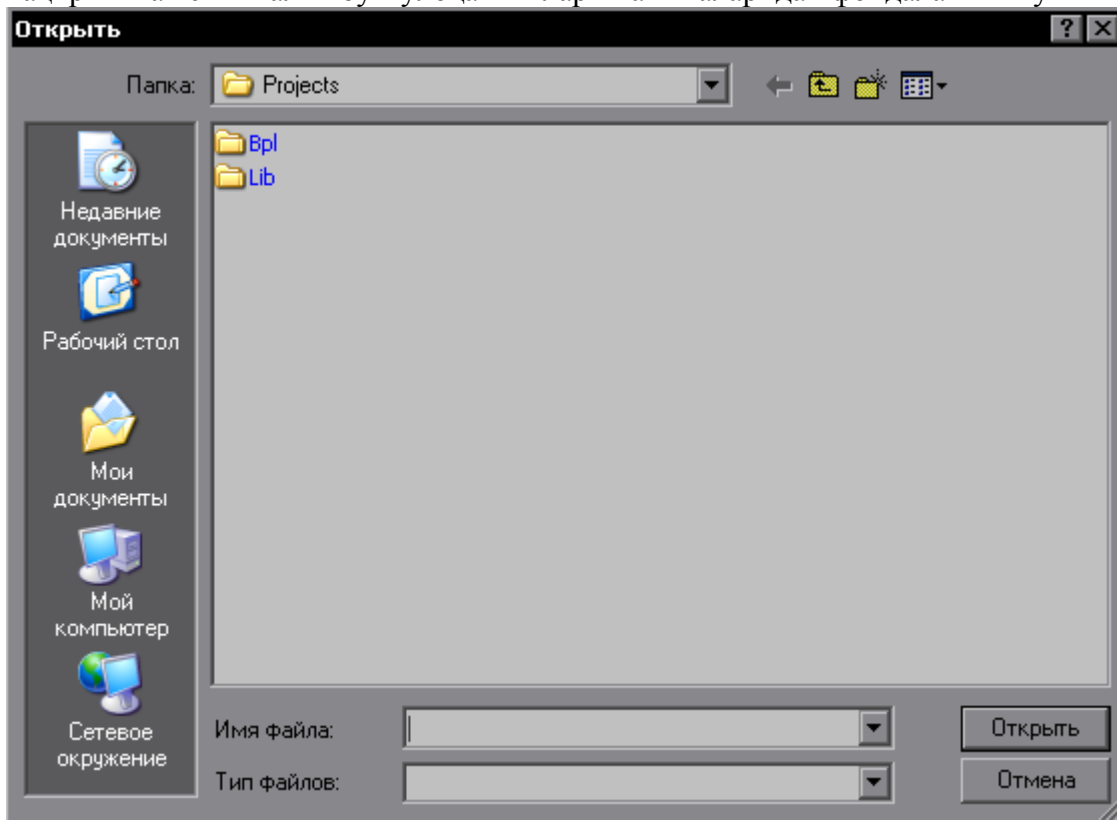
Маълумотларни файлдан ўқиш учун ">>" буйруғига эквивалент бўлган гет функцияси ишлатилади. пут функциясига ўхшаб гет функцияси ҳам ўзгарувчиларнинг ва белгилар массивининг ихтиёрий стандарт турлари билан ишлай олади. Ундан ташқари гетлине функцияси ҳам мавжуд. У гет функцияси билан сатр охирининг охириги белгисини қайтармаслиги билангина фарқ қилади.

Файлларни очиш сақлаш мулоқот ойнаси

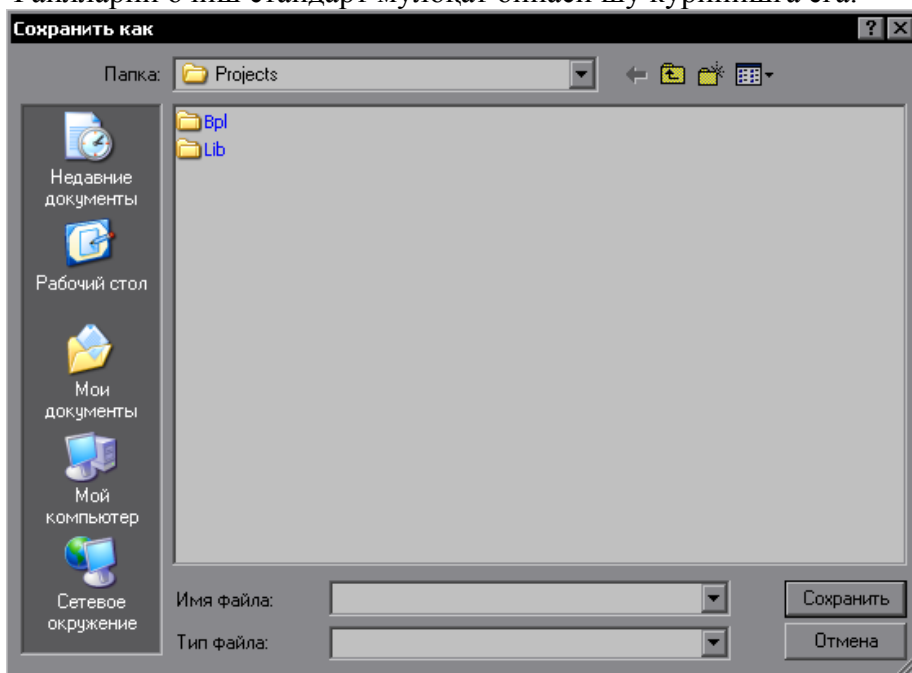
Windows операцион тизимида файлларни излаш учун файлларни сақлаш ва очиш универсал мулоқат ойналари назарда тутилган. Dialogs бандининг мос компоненталарини шаклда ўрнатиб улардан фойдаланиш мумкин.



Бу компоненталарни шаклда ўрнатиб файллар билан ишлаш стандарт мулоқат ойналарини чақириш ва кейинчалик бу мулоқат ишлари натижаларидан фойдаланиш мумкин.



Файлларни очиш стандарт мулоқат ойнаси шу кўринишга эга.



Файлларни сақлаш стандарт мулоқат ойнаси шу кўринишга ега.

«Сақлаш» - «Сохранит» ёки «Очиш» - «Открыть» тугмалари босилгандан сўнг, аммо ойнани ёпгунга қадар (яъни анлосе ҳолати пайтида), фойдаланувчи томонидан танланган файл номи AnsiString турига эга шакл объектга мос FileName хоссасида жойлашади.

«Матн муҳаррири»

Дастур таснифи

Вазифа: Матн файлларини ташкил қиувчи ва ўзгартирувчи дастур тузилсин. Дискдаги файлларни очиш ва критилган ўзгартиришларни сақлаш имкониятлари амалга оширилсин.

Файлларни излаш, ҳамда файлларни сақлаш жойини танлаш учун файлларни очиш/сақлаш стандарт мулоқатлардан фойдаланилсин.

Файлнинг матни Memo майдонида акс еттирилсин.

Муаммолар

ifstream ва ofstream синфларининг объектлари файлларни ҳосил қилиш ва улар билан ишлашда узатилаётган файл номини белгилар массиви сифатида ишлатади. Стандарт мулоқатлар эса «сатр» (AnsiString) туридаги кийматларни қайтаради. Яъни стандарт мулоқат ойналари қайтараётган кийматларни тўғридан-тўғри ifstream ёки ofstream объектларга узатишнинг имкони йўқ. Бу муаммони ечиш учун сатрни белгилар массивига алмаштирувчи процедура тузиш тавсия этилади.

Зарур кўникмалар

Мазкур дастурни ёзиш учун дастур яратиш муҳитининг стандарт компоненталари, файлларни излашга мўлжалланган мулоқат ойналари билан ишлашни билиш керак. Ундан ташқари матн ҳолатдаги файлларни ўқиш ва дискда сақлашни ҳам билиш зарур.

Ечиш

Мазкур дастурни ёзишда шаклда мос равишда файлларни очадиган ва сақлайдиган тугмачаларни жойлаштиришга тўғри келади. Яна мос мулоқат ойналарини ҳам ўрнатиш талаб этилади.

Тугмачаларни босиш ходисаларини қайта ишловчига мулоқат ойналарини чақириш ўрнатилади: SaveDialog1->Execute. Мулоқат ойнасининг OnCanClose ходисаларни қайта ишловчисига еса файллар билан ишлашни амалга оширувчи дастур коди ўрнатилади.

Файллар билан ишлаш мулоқат ойнасининг OnCanClose ходисаси содир бўлганда мос мулоқат ойнасининг FileName хоссасида танланган файлнинг номи пайдо бўлади. Айнан шу файл билан ишлаш керак бўлади.

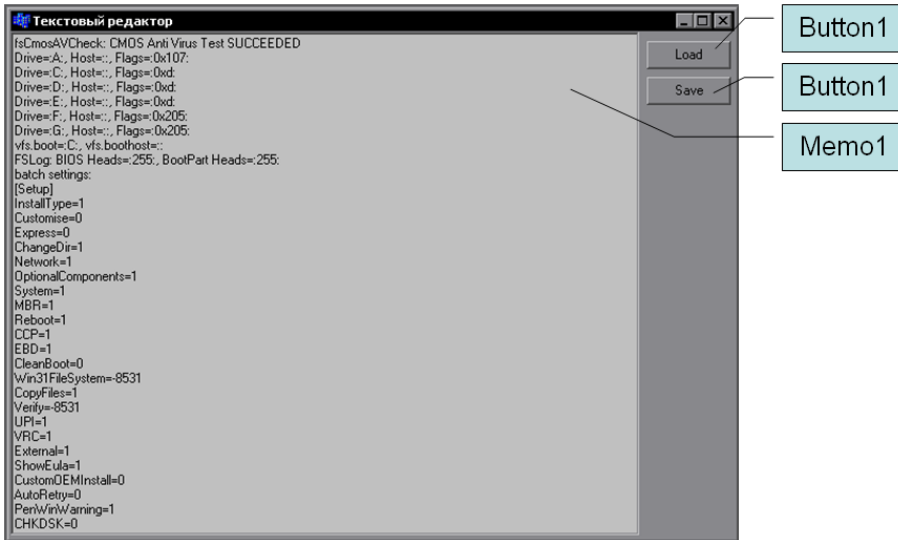
ifstream синф объектнинг сатрига ёзилган файл тўғрисидаги маълумотларни узатиш учун сатрни белгилар массивига алмаштиришга тўғри келади. Бу ишни массивнинг биринчи элементига мурожатни ва бевосита сатрни узатиш мўлжалланган процедура яратиб осонгина амалга ошириш мумкин. Бу процедура белгиларни сатрдан олиб, массивни кетма-кет, элементма-элемент тўлдиради. Бу процедуранинг ёрдамида барча керакли алмаштиришларни осонгина бажариш мумкин.

Файлнинг мазмунини Memo1 майдонига ёзиш учун сатрни ifstream синфининг getline() функцияси ёрдамида кетма-кет ўқиш ва унинг қисм объекти Lines (Memo1->Lines->Add(stroka);) ning Add() функцияси ёрдамида Memo1 майдонига ёзиш керак.

Маълумотларни файлда сақлаш учун файлга Memo1 объектини сатрини белгима-белги, сатр охири белгиси ('\n') ни қўшиб ва янги сатрдан бошлаб ёзиш зарур

Шакл

Шаклда кўринувчи компоненталардан ташқари Dialogs бандининг SaveDialog ва OpenFileDialog компоненталари ҳам қатнашади.



Дастур коди

```

#include <vcl.h>
#include <fstream.h>

#pragma hdrstop

#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void AnsiStringToCharPointer(char * c, AnsiString s, int n = 0)
{
int k = s.Length();
if(k > n && n != 0){ k = n; }
for(int i = 0; i < k; i++)
{
c[i] = s[i+1];
}
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
OpenDialog1->Execute();
}
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
SaveDialog1->Execute();
}
//-----
void __fastcall TForm1::OpenDialog1CanClose(TObject *Sender,

```

```

    bool &CanClose)
{
    AnsiString name = OpenFileDialog1->FileName;
    char c[100] = "";
    AnsiStringToCharPointer(c,name);
    ifstream in(c);
    Memo1->Text = "";
    while(in.good())
    {
        in.getline(c,100);
        Memo1->Lines->Add(c);
    }
    in.close();
}
//-----
void __fastcall TForm1::SaveDialog1CanClose(TObject *Sender,
    bool &CanClose)
{
    AnsiString name = SaveDialog1->FileName;
    char c[1000] = "";
    AnsiStringToCharPointer(c,name);
    ofstream out(c);

    for(int i = 1;i < Memo1->Lines->Count;i++)
    {
        AnsiString s = Memo1->Lines->operator [](i);
        for(int j = 1; j < s.Length()+1; j++)
        {
            out << s[j];
        }
        out << 'g'n';
    }
    out.close();
}

```

Лаборатория ишини бажарилиш тартиби.

1. Ишнинг мақсадини тушиниш.
2. Масалани куйилиши ўрганиш.
3. Масалани ечиш кетма-кетлигини аниқлаш.
4. Масалани ечиш алгоритмини тузиш.
5. Масалани дастурини тузиш.
6. С++ муҳитида дастур матнини киритиш.
7. Компьютерда натижани олиш.
8. Олинган натижани таҳлил қилиш.
9. Бажарилган иш бўйича ҳисобот тайёрлаш.

Ҳисоботнинг мазмуни

1. Ишнинг назарий асослари.
2. Масаланинг ечиш алгоритми.
3. Масалани ечиш дастури.
4. Ҳисоб натижалари ва олинган натижаларни таҳлили.
- 5.

Топширик вариантлари:

Берилган куш йиғинди, куш купайтма, куш йиғинди-купайтма ёки куш купайтма-йиғиндиларнинг қийматининг ҳисоблаш алгоритми ва дастурини тузинг.

1. Лабоаратория ишини бажариш учун топширик вариантлари.

Ишнинг бажариш учун тавсия қилинган адабиётлар.

1. Джесс Либерти. Освой самостоятельно С++ за 21 день.- СПб.:Питер.2003.815с.
2. Трехтомник Берна Страуструпа
3. Қораев Ф ва б. Программалашдан масалалар тўплами (методик қўлланма)-Тошкент -2008
4. Назиров Ш. Объектга мулжалланган дастурлаш тиллари - Тошкент- 2006