

**O'ZBEKISTON RESPUBLIKASI
OLIV VA O'RTA-MAXSUS TA'LIM VAZIRLIGI**

SAMARQAND DAVLAT UNIVERSITETI

**MEXANIKA-MATEMATIKA FAKULTETI
AMALIY MATEMATIKA VA INFORMATIKA BO'LIMI
5110700-INFORMATIKA O'QITISH METODIKASI YO'NALISHI**

„Sanoq sestemalari ustida amallar bajarish uchun interaktiv dasturiy ta'minot“

MAVZUSIDAN

KURS ISHI



206 guruh talabasi : Xasanov J.

Kurs ishi rahbari: Nazarov F.

SAMARQAND-2016

Mundarija

Kirish.

I. C++ tilida bir sanoq sestemadan boshqa sanoq sestemaga o'tuvchi dastur tuzishdair nazariy tushunchalar;

1.1 Turlar va C++ da o'zgaruvchilarni tavsiflash

1.2. Belgi va satrlar.

II. C++ tilida bir sanoq sestemadan boshqa sanoq sestemaga o'tuvchi dasturdasturiy ta'minoti;

2.1. Sanoq sistemalarni boshqasiga o'tkazishning matematik tahlili.

2.2. Borland C++ da bir sanoq sestemadan boshqa sanoq sestemaga o'tuvchi dastur dasturiy ta'minoti;

Xulosa.

Foydalanilgan adabiyotlar ro'yhati

Kirish.

Bizga ma'lumki, dasturlash tillarining yuzdan ortiq ko'rinishlari mavjud, lekin qo'llanilishi ko'lamiga qarab C/C++ va C# dasturlash tillari yuqori dasturlash sinfiga mansubdir. Keyingi yillarda amaliy dasturchilarga juda ko'p integratsion dastur tuzish muhitlari taklif etilmoqda. Bu muhitlar u yoki bu imkoniyatlari bilan bir-biridan farq qiladi.

Men ham ushbu kurs ishim davomida C++ tilida sanoq sistemalar bilan ishlovchi obektlarga mo'ljallangan dasturni tuzib chiqmoqchiman. Ushbu dastur kichik dastur bo'lsada kelajakda boshqa dasturlar tuzishim uchun fundament vazifasini bajarishga harakat qilaman.

Bu dasturni tuzib nafaqat hisoblashni amalga oshirish balki kichik dastur ko'rinishdagi hisoblash dasturi va sanoq sestemani tushinmaydigan foydalanuvchi ham bir ko'rinishda tushinadigan tushinarli, bejirm ko'rinishga ega dastur tuzmoqchiman.

Men bu dasturni tuzib obektlar yaratishni va ular orqali shunga o'xshash yana boshqa obektlar yaratmoqchiman bu dasturda ishlashimga yana bir sabab boshqa obektlarga mo'ljallanmagan dasturlash tillarida faqat masalani matematik dasturi tuzib natija olinadi bunda yani C++ builderda esa boshqa imkoniyatlar va kampanentalar bilan ham ishlash imkoni bor.

EHM - bu elektron raqamli qurilmadir. Elektron qurilma deyilishiga sabab har qanday ma'lumotlar EHM da elektr signallari orqali qayta ishlanadi. Raqamli deyilishiga sabab EHM da har qanday ma'lumot sonlar yordamida tasvirlanadi. Sonlarni yozish usuliga sanoq sistemasi deb ataladi. Sonlarni yozish uchun har bir sanoq sistemasida o'ziga xos turli belgilar to'plamidan foydalaniladi. Foydalanilgan to'plamdagi belgilar ularning soni, sanoq sistemasini xarakterlovchi asosiy kattaliklardir. Sanoq sistemasida foydalaniladigan belgilar soni sanoq sistemasining asosini tashkil etadi. Berilgan sanoq sistemasida sonlarni yozishdagi foydalanilgan belgilar soniga qarab, o'nlik, ikkilik, sakkizlik, o'n oltilik va boshqa sanoq sistemalarni kiritish mumkin.

C++ tilida bir sanoq sestemadan boshqa sanoq sestemaga o'tuvchi dastur tuzish doir nazariy tushunchalar;

1. Turlar va C++ da o'zgaruvchilarni tavsiflash

Har bir nom va har bir o'zgaruvchi ular ustida bajariluvchi amallar aniqlovchi turlarga ega bo'ladi. Masalan, `int i;` tavsiflash `i` o'zgaruvchi `int` turiga tegishli, ya'ni `i` butun o'zgaruvchi deb aniqlaydi. Tavsiflash - dasturga nom kirituvchi buyruqdir. Tavsiflash o'zgaruvchining turini aniqlaydi. Tur nom va ifodalardan to'g'ri foydalanishni aniqlaydi. Butun tur uchun quyidagi amallar aniqlangan: `+`, `-`, `*` va `/`. Asosiy turlar. Bevosita apparat ta'minotiga javob beradigan asosiy turlar quyidagilar: `char`; `short`; `int`; `long`; `float`; `double`. Birinchi to'rtta tur butun kattaliklarni, oxirgi ikkitasi suzuvchi nuqtali, ya'ni kasr sonlarni tasvirlash uchun ishlatiladi. `char` turidagi o'zgaruvchi mazkur kompyuterda belgilarni (odatda bayt) saqlash o'lchoviga ega, `int` turidagi o'zgaruvchi esa mazkur kompyuterdagi butun arifmetikaga mos o'lchovga ega (odatda so'z). Turlar bilan tasvirlangan butun sonlar diapazoni uning o'lchoviga bog'liq bo'ladi (uni `sizeof` buyrug'i yordamida hisoblash mumkin). C++ da o'lchovlar `char` turidagi kattaliklar o'lchovi birligida o'lchanadi. Asosiy turlar o'rtasidagi munosabatlarni quyidagicha yozish mumkin:

$$1 = \text{sizeof}(\text{char}) \leq \text{sizeof}(\text{short}) \leq \text{sizeof}(\text{int}) \leq \text{sizeof}(\text{long}) = \text{sizeof}(\text{float}) \leq \text{sizeof}(\text{double}).$$

Umuman, asosiy turlar xususida yana boshqa narsalarni faraz qilish ma'nosiz. Xususan, ko'rsatgichlarni saqlash uchun butun tur etarli, degan xulosa barcha kompyuterlar uchun to'g'ri emas. Asosiy turlarga `const` so'zini qo'shib tavsiflash mumkin. Bu boshlang'ich turga shu turning o'zini beradi, faqat bu holatda `const` turidagi o'zgaruvchilarning qiymatlari initsializatsiyadan so'ng o'zgarishi mumkin emas.

```
const float pi = 3.14;
```

```
const char plus = '+';
```

Bittalik qo'shtirnoqqa olingan belgilar belgi o'zgarmaslar hisoblanadi.

Shunga e'tibor berish lozimki, bu usulda tavsiflangan o'zgarmaslar xotirada joy egallamaydi. uning qiymati talab qilingan joyda bevosita ishlatiladi. O'zgarmaslar

initsializatsiya paytida tavsiflanishi shart. O'zgaruvchilar uchun initsializatsiya shartemas, ammo albatta tavsiya qilinadi. Lokal o'zgaruvchilarni initsializatsiyasiz kiritish asoslari juda ko'p. Bu turlarning ixtiyoriy kombinatsiyasiga quyidagi arifmetik amallar qo'llanilishi mumkin:

- + (plyus, unar va binar);
- (minus, unar va binar);
- * (ko'paytirish);
- / (bo'lish).

Hamda taqqoslash amallari:

- == (teng);
- != (teng emas);
- < (kichik);
- > (katta);
- <= (kichik yoki teng);
- >= (katta yoki teng).

Agar operandlar qo'yilgan shartni qanoatlantirsa, u holda taqqoslash amallari natijada 1 qiymatni beradi, aks holda esa 0 qiymatni beradi. Butunga bo'lish amali butun natijani beradi: $7/2 = 3$. Butun kattaliklar ustida % - qoldiqni hisoblash amali bajariladi: $7\%2 = 1$. O'zlashtirishda va arifmetik amallarda C++ ularni guruhlash uchun asosiy turlar o'rtasida barcha ma'noli almashtirishlarni bajaradi:

```
double d = 1;  
int i = 1;  
d = d + i;  
i = d + i;
```

Satriy turlar. C++ da belgilarning biron-bir ketma-ketligi (massivlar) dan iborat matn qatorlarini xotirada saqlash uchun maxsus AnsiString ma'lumotlar turi qo'llaniladi.

«Stroka» - «Satr» turidagi o'zgaruvchilar barcha boshqa o'zgaruvchilar kabi e'lon va initsializatsiya qilinadi. Kompilyatorga navbatdagi belgilar ketma-ketligi yangi o'zgaruvchining nomi emas, balki satr ekanligini bildirish uchun satrlar bittalik qo'shtirnoq ichiga olinadi. Misol:

```
AnsiString st = 'matn qatori';
```

Satr turidagi o'zgaruvchilar ustida boshqa satr o'zgaruvchilar bilan qo'shish amali bajarilishi mumkin. Bu amal ikkita satrni ularning kelish tartibida birlashtirish deb tushuniladi. Misol:

```
AnsiString s1 = 'qatori';
```

```
AnsiString s2 = 'matn';
```

```
AnsiString s = s1 + s2;
```

Natijada s o'zgaruvchi s1 va s2 o'zgaruvchilardan tashkil topgan 'stroka teksta' degan qiymatni qabul qiladi. Qo'shimcha turlar Borland C++ da butun qiymatli o'zgaruvchilarning turlarini qo'shimcha ajratish imkoni mavjud. Bu holda o'zgaruvchilarning barcha tur nomlari quyidagicha yoziladi - int X, bu erda X o'zgaruvchiining bitlardagi maydon o'lchami. X quyidagi qiymatlardan birini qabul qilishi mumkin: 8, 16, 32 va 64. Bu turdagi o'zgaruvchilardan foydalanish standart turda aniqlangan o'zgaruvchilardan foydalanishdan farq qilmaydi.

Quyidagi jadvalda bunday turlar bilan ishlash yaqqol ko'rsatilgan.

Tur nomi misol O'lcham

```
int8 c = 128; 8 bit
```

```
int16 s = 32767; 16 bit
```

```
int32 i = 123456789; 32 bit
```

```
int64 big = 12345654321; 64 bit
```

Turlarni o'zgartirish protseduralari. Standart turlarni o'zgartirish C++ ning ma'lumotlarning turlari ustida qattiq nazorati tufayli imkoni boricha qiymatlarni saqllovchi, turlarni o'zgartirish amallari kiritilgan. Boshqa o'zgaruvchidan ma'lum bir tur qiymatlarini olish uchun quyidagi konstruktsiya ishlatiladi: (yangi tur)o'zgaruvchi. Misol:

```
short S = 100;
```

```
int I = (int)S;
```

Bu misol ortiqcha buyruqlarga ega. C++ da ko'pgina tur o'zgaruvchilarining to'g'ridan-to'g'ri o'zlashtirilishi nazarda tutilgan, ammo ba'zi hollarda bu buyruqlar majburiy hisoblanadi (masalan, o'zgaruvchining qiymatini biror funksiyaga uzatishda).

Sonli qiymatlarni satrga almashtirish. C++ turlarning to'g'ridan-to'g'ri almashtirishda o'zgaruvchini uning o'nlik ko'rinishidan belgilar qatori ko'rinishiga yo'l qo'ymaydi, chunki, ular shakllarning ko'gina komponentalarida ishlatiladi. To'g'ridan-to'g'ri almashtirish faqatgina asosiy va qo'shimcha turlar uchun amalga oshiriladi. Massiv hisoblanadigan satr kattaliklar hosilaviy tur bo'lganligi sababli bunday almashtirishga yo'l qo'yilmaydi. Bunday almashtirishlar uchun quyidagi standart almashtirish funksiyalari ishlatiladi: IntToStr, StrToInt, FloatToStr va boshqalar. Ko'pchilik ma'lumotlar turlari uchun shu kabi satrga va teskari o'tkazish funksiyalari mavjud. Misol:

```
char S[10]; // belgilar massivi
int I = 100; // butun qiymatli o'zgaruvchi
S = IntToStr(I); // o'tkazish
```

Shartli buyruq. Dasturda tarmoqlanishni amalga oshirish, ya'ni ba'zi faktorlarga bog'liq holda turli amallar bajarilishi uchun if buyrug'i ishlatiladi. Buyruq quyidagi formatga ega:

```
if (ifoda){ 1 - operator;} [else { 2 - operator;}]
```

if-buyrug'ining bajarilishi ifodaning qiymatini hisoblashdan boshlanadi. So'ngra ish quyidagi sxema asosida amalga oshiriladi:

agar ifoda rost bo'lsa (ya'ni 0 dan farqli), u holda 1 - operator bajariladi.

agar ifoda yolg'on bo'lsa (ya'ni 0 ga teng), u holda 2 - operator bajariladi.

agar ifoda yolg'on va 2 - operator yo'q bo'lsa (kvadrat qavsga zarur bo'lmagan konstruktsiya kiritiladi), u holda if dan keyingi buyruq bajariladi. Misol:

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

}

Bu misol 1 - operatorning o'rnida ham, 2 - operatorning o'rnida ham murakkab konstruktsiya qatnashishi mumkinligini bildiradi. Ichma-ich if buyrug'ini ishlatish imkoniyati ham mavjud. if buyrug'i boshqa if buyrug'ining if yoki else konstruktsiyalari ichida qatnashishi ham mumkin.

2.Belgi va satrlar: Standart C++ tili ikki xildagi belgilar majmuasini qo'llab quvvatlaydi. Birinchi toifaga, an'anaviy, «tor» belgilar deb nomlanuvchi 8-bitli belgilar majmuasi kiradi, ikkinchisiga 16-bitli «keng» belgilar kiradi. Til kutubxonasida har bir guruh belgilari uchun maxsus funksiyalar to'plami aniqlangan.

C++ tilida satr uchun maxsus tur aniqlanmagan. Satr char turidagi belgilar massivi sifatida qaraladi va bu belgilar ketma-ketligi satr terminatori deb nomlanuvchi 0 kodli belgi bilan tugaydi ('\0'). Odatda, nol-terminator bilan tugaydigan satrlarni ASCIIZ-satlar deyiladi.

Quyidagi jadvalda C++ tilida belgi sifatida ishlatilishi mumkin bo'lgan o'zgarmaslar to'plami keltirilgan.

C++ tilidagi belgi o'zgarmaslar

Belgilar sinflari	Belgi o'zgarmaslar
Katta harflar	'A'...'Z', 'A'...'Я'
Kichik qarflar	'a'...'z', 'a'...'я'
Raqamlar	'0'...'9'
Bo'sh joy	gorizontal tabulyatsiya (ASCII kodi 9),satrni o'tkazish (ASCII kodi 10), vertikal tabulyatsiya (ASCII kodi 11), formani o'tkazish (ASCII kodi 12) karetkani qaytarish (ASCII kodi 13)
Punktuasiya belgilari (ajratuvchilar)	! " # \$ % & ' () * + - , . / : ; < = > ? @ [\] ^ _ { } ~
Boshqaruv belgilari	ASCII kodi 0...1Fh oralig'ida va 7Fh bo'lgan belgilar
Probel	ASCII kodi 32 bo'lgan belgi
O'n oltilik raqamlar	'0'...'9', 'A'...'F', 'a'...'f'

Satr massivi e'lon qilinishida, satr oxiriga terminator qo'yilishi va natijada satrga qo'shimcha bitta bayt bo'lishini inobatga olinishi kerak:

```
char satr[10];
```

Ushbu e'londa satr satri uchun jami 10 bayt ajratiladi - 9 satr hosil qiluvchi belgilar uchun va 1 bayt terminator uchun.

Satr o'zgaruvchilar e'lon qilinishida boshlang'ich qiymatlarni qabul qilishi mumkin. Bu holda kompilyator avtomatik ravishda satr uzunligini hisoblaydi va satr oxiriga terminatorni qo'shib qo'yadi

Satrqiymatinio'qishdaoqimlio'qishoperatori ">>" o'rnigagetline() funksiyasini ishlatganma'qulhisoblanadi, chunkioqimlio'qishdaprobellarinkorqilinadi (garchiularsatrbelgisihisoblansaham) o'qilayotganbelgilarketma-ketligisatrdan «oshib» ketgandahambelgilarnikiritishdavometishimumkin.

Natijadasatro'zigaajratilgano'lchamdanortiqbelgilarni «qabul» qiladi. Shusababli, getline() funksiyasiikkita parametrga egabo'lib,

birinchiparametro'qishamalgaoshirilayotgansatrgako'rsatkich,

ikkinchiparametrdaesao'qilishikerakbo'lganbelgilarsoniko'rsatiladi.

Programmadaishlatilgansatrsatri 5 tabelginiqabulqilishimumkin, ortiqchalaritashlabyuboriladi. getline()

funksiyasigamurojaatdaikkinchiparametrqiymatio'qilayotgansatr uzunligidankattabo'lmas ligikerak.

Satr bilan ishlaydigan funksiyalarning aksariyati «string.h» kutubxonasida jamlangan. Nisbatan ko'p ishlatiladigan funksiyalarning tavsifini keltiramiz.

Satr uzunligini aniqlash funksiyalari: Satrlar bilan ishlashda, aksariyat hollarda satr uzunligini bilish zarur bo'ladi. Buning uchun «string.h» kutubxonasida strlen() funksiyasi aniqlangan bo'lib, uning sintaksisi quyidagicha bo'ladi:

```
size_t strlen(const char* string )
```

Bu funksiya uzunligi hisoblanishi kerak bo'lgan satr boshiga ko'rsatkich bo'l-gan yagona parametrga ega va u natija sifatida ishorasiz butun sonni qaytaradi. strlen() funksiyasi satrning real uzunligidan bitta kam qiymat qaytaradi, ya'ni nol-terminator o'zni hisobga olinmaydi.

Xuddi shu maqsadda sizeof() funksiyasidan ham foydalanish mumkin va u strlen() funksiyasidan farqli ravishda satrning real uzunligini qaytaradi.

Odatda sizeof() funksiyasidan getline() funksiyasining ikkinchi argumenti si-fati ishlatiladi va satr uzunligini yaqqol ko'rsatmaslik imkonini beradi:

Satrlarni nusxalash: Satr qiymatini biridan ikkinchisiga nusxalash mumkin. Bu maqsadda bir qator standart funksiyalar aniqlangan bo'lib, ularni ayrimlarining tavsiflarini keltiramiz.

strcpy() funksiyasi prototipi

```
char* strcpy(char* str1, const char* str2);
```

ko'rinishga ega va bu funksiya str2 satrdagi belgilarni str1 satrga baytma-bayt nusxalaydi. Nusxalash str2 ko'rsatib turgan satrdagi nol-terminal uchraguncha davom etadi. Shu sababli, str2 satr uzunligi str1 satr uzunligidan katta emas-ligiga ishonch hosil qilish kerak, aks hol berilgan sohasida (segmentda) str1 satrdan keyin joylashgan berilganlar «ustita» str2 satrning «ortib qolgan» qismi yozilishi mumkin.

Navbatdagi programma qismi “Satrni nusxalash!” satrini Str satrga nusxalaydi:

```
char Str[20];
```

```
strcpy(Str, “Satrni nusxalash!”);
```

Zarur bo'lganda satrning qaysidir joyidan boshlab, oxirigacha nusxalash mumkin.

strncpy() funksiyasining strcpy() funksiyasidan farqli joyi shundaki, unda bir satrdan ikkinchisiga nusxalanadigan belgilar soni ko'rsatiladi. Uning sintaksisi quyidagi ko'rinishga ega:

```
char* strncpy(char* str1, const char* str2, size_t num)
```

Agar str1 satr uzunligi str2 satr uzunligidan kichik bo'lsa, ortiqcha belgilar «kesib» tashlanadi.

strdup() funksiyasiga yagona parametr sifatida satr-manbaga ko'rsatkich uzatiladi. Funksiya, satrga mos xotiradan joy ajratadi, unga satrni nusxalaydi va yuzaga kelgan satr-nusxa adresini javob sifatida qaytaradi, strdup() funksiya sintaksisi:

```
char* strdup(const char* source)
```

Quyidagi programma bo'lagida satr1 satrining nusxasi xotiraning satr2

ko'rsatgan joyida paydo bo'ladi:

```
char* satr1="Satr nusxasini olish."; char* satr2;  
satr2=strdup(satr1);
```

Satrlarni ulash: Satrlarni ulash (konkatenatsiya) amali yangi satrlarni hosil qilishda keng ko'llaniladi. Bu maqsadda «string.h» kutubxonasida strcat() va strncat() funksiyalari aniqlangan.

strcat() funksiyasi sintaksisi quyidagi ko'rinishga ega:

```
char* strcat(char* str1, const char* str2)
```

Funksiya ishlashi natijasida str2 satr, funksiya qaytaruvchi satr str1 satr oxiriga ulanadi. Funksiyani chaqirishdan oldin str1 satr uzunligi, unga str2 satri ulanishi uchun yetarli bo'lishi hisobga olingan bo'lishi kerak.

Quyida keltirilgan amallar ketma-ketligining bajarilishi natijasida satr satriga qo'shimcha satr ulanishi ko'rsatilgan:

```
char satr[80];  
strcpy(satr,"Bu satrga");  
strcat(satr,"satr osti ulandi.");
```

Amallar ketma-ketligini bajarilishi natijasida ko'rsatayotgan joyda "Bu satrga satr osti ulandi." satri paydo bo'ladi.

strncat() funksiyasi strcat() funksiyadan farqli ravishda str1 satrga str2 satr-ning ko'rsatilgan uzunlikdagi satr qismini ulaydi. Ulanadigan satr qismi uzunligi funksiyaning uchinchi parametri sifatida beriladi.

Garchi bu masala ostilarining yechimlari C++ tilning standart kutubxonalarida funksiyalar ko'rinishida mavjud bo'lsa ham, ular kodini qayta yozish foydalanuv-chiga bu amallarning ichki mohiyatini tushunishga imkon beradi.

II.C++ tilida bir sanoq sestemadan boshqa sanoq sestemaga o'tuvchi dastur dasturiy ta'minoti;

1.Sanoq sistemalarni boshqasiga o'tkazishning matematik tahlili.

EHM - bu elektron raqamli qurilmadir. Elektron qurilma deyilishiga sabab har qanday ma'lumotlar EHM da elektr signallari orqali qayta ishlanadi. Raqamli deyilishiga sabab EHM da har qanday ma'lumot sonlar yordamida tasvirlanadi. Sonlarni yozish usuliga sanoq sistemasi deb ataladi. Sonlarni yozish uchun har bir sanoq sistemasida o'ziga xos turli belgilar to'plamidan foydalaniladi. Foydalanilgan to'plamdagi belgilar ularning soni, sanoq sistemasini xarakterlovchi asosiy kattaliklardir. Sanoq sistemasida foydalaniladigan belgilar soni sanoq sistemasining asosini tashkil etadi. Berilgan sanoq sistemasida sonlarni yozishdagi foydalanilgan belgilar soniga qarab, o'nlik, ikkilik, sakkizlik, o'n oltilik va boshqa sanoq sistemalarni kiritish mumkin. Shu bilan birga sanoq sistemalarini *pozision* va *nopozision* turlarga ajratish mumkin. Pozitsion sanoq sistemasida berilgan sonning qiymati sonni tasvirlovchi raqamlarning egallagan o'rniga bog'liq bo'ladi. Misol sifatida, 0,1,2,3,. . . ,9 arab raqamlaridan tashkil topgan o'nlik sanoq sistemani qarash mumkin. Nopozitsion sanoq sistemalarida, belgining qiymati uning egallagan o'rniga bog'liq emas. Misol sifatida rim raqamlari sanoq sistemasini keltirish mumkin. Masalan, XX sonida X raqami, qayerda joylashganiga qaramasdan o'nlik sanoq sistemasidagi 10 qiymatini anglatadi. Quyidagi jadvalda o'nlik sanoq sistemasida berilgan 1 dan 16 gacha sonlarning ikkilik, sakkizlik va o'n oltilik sanoq sistemalaridagi ko'rinishi keltirilgan.

SANOQ SISTEMALARI							
2	3	4	5	6	8	10	16
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1

10	2	2	2	2	2	2	2
11	10	3	3	3	3	3	3
100	11	10	4	4	4	4	4
101	12	11	10	5	5	5	5
110	20	12	11	10	6	6	6
111	21	13	12	11	7	7	7
1000	22	20	13	12	10	8	8
1001	100	21	14	13	11	9	9
1010	101	22	20	14	12	10	?
1011	102	23	21	15	13	11	?
1100	110	30	22	20	14	12	?
1101	111	31	23	21	15	13	?
1110	112	32	24	22	16	14	?
1111	120	33	30	23	17	15	F
10000	121	100	31	24	20	16	10

Bu jadval bo'yicha bir sanoq sistamasidan ikkinchisiga o'tish masalasini ko'rib o'taylik.
Masalan: 10 lik sanoq sistemasidagi 13 soniga 8 lik sanoq sistemasida 15 soni mos keladi

va u 13 ni 8 ga bo'linganda hosil bo'lgan butun son 1 va qoldiq 5 lardan tashkil topgan. Xuddi shuningdek 13 ni 6 ga bo'lganda hosil bo'luvchi butun son 2 va qoldiq 1 lar 21 sonini hosil qiladi. Bu son 13 sonining 6 lik sanoq sistemasidagi qiymatidir. Odatda biror X sonining qaysi sanoq sistemasiga tegishliligini ko'rsatish uchun uning pastida indeks sifatida zarur sanoq sistemasining asosi ko'rsatiladi. Masalan, X_6 – X sonining 6 lik sanoq sistemasiga tegishli ekanligini ko'rsatadi.

$X_{10}=13$ sonining X2-ikkilik sanoq sistemasidagi ko'rinishini topaylik. Yuqoridagidek, 13 ni ketma-ket 2 ga bo'lamiz va bo'lishni to butun qismida nol hosil bo'lguncha davom ettiramiz.

$$\begin{array}{r}
 13 \mid 2 \\
 \hline
 12 \mid 6 \mid 2 \\
 \hline
 \textcircled{1} \mid 6 \mid 3 \mid 2 \\
 \hline
 \mid 0 \mid 1 \mid 2 \\
 \hline
 \mid 1 \mid 0 \mid 0 \\
 \hline
 \mid \mid 1 \mid 0
 \end{array}$$

O'ngdan chapga tartibida yozilgan qoldiqlar, ya'ni 1101 soni $X_{10}=13_{10}$ sonining ikkilik sanoq sistemasidagi ko'rinishi bo'ladi. Endi 8 lik sanoq sistemasidan 10 lik sanoq sistemasiga bo'lish yo'li bilan o'tishga doirmisollar ko'raylik. Masalan, jadval bo'yicha 15_8 ga 13_{10} mos keladi. Endi uni topib kuraylik, buning uchun 15_8 ni 10 lik sanoq sistemasining asosi–10 ning 8 lik sanoq sistemasidagi ko'rinish – 12 ga bo'lish kerak bo'ladi. 15_8 ni 12 ga bo'lsa butun qismida 1 va qoldiqda 3, ya'ni 13_{10} – hosil bo'ladi. Bunga jadval orqali ishonch hosil qilish ham mumkin.

Ikkinchi *misol*: 175_8 sonini 10 lik sanoq sistemasidagi ko'rinishini topish talab qilingan bo'lsin. Xuddi yuqoridagidek 175_8 ni 12 ga ketma-ket bo'lamiz. Eslatib o'tamiz, bo'lish amali 8 sonlik sanoq sistemasida olib boriladi. (Jadvalga qaralsin)

$$\begin{array}{r|l}
 175 & 12 \\
 - 12 & 14 \quad 12 \\
 \hline
 55 & 12 \quad (1) \\
 - 50 & (2) \\
 \hline
 (5) &
 \end{array}$$

R sanoq sistemasida berilgan sonni Q sanoq sistemasiga o'tkazish uchun, R sanoq sistemasidagi X soni Q sanoq sistemasining asosiga, ya'ni Q ga ketma-ket, to butun qismida 0 hosil bo'lguncha davom ettirish kerak. Qoldiqlar o'ngdan chapga karab ketma-ket yozilsa, R sanoq sistemasida berilgan X_r sonining Q sanoq sistemasidagi X_q ko'rinishi hosil bo'ladi. Bo'lish amali berilgan R sanoq sistemasida amalga oshiriladi. Ba'zi bir sanoq sistemalaridan ikkinchisiga qulayroq, osonroq holda o'tish imkoniyatlari mavjud. Xususiyl holda, 2 ga karrali sonlarning biridan 2 ikkinchisiga o'tish qoidasini ko'rib o'tamiz.

Masalan, 8 lik sanoq sistemasida berilgan $X_8=5361$ sonidan X_2 ga bo'lish uchun, X_8 ning har bir raqamini 2 likdagi ko'rinishi-triadalar ($2^3=8$) bilan almashtirib chiqamiz:

$$X_2 = \begin{array}{cccc}
 \underline{101} & \underline{011} & \underline{110} & \underline{001} \\
 5 & 3 & 6 & 1
 \end{array}$$

$D8A2_{16}$ ni 2 lik sanoq sistemasiga o'tkazish uchun uning har bir raqamini 2 lik sanoq sistemasidagi to'rtliklar-tetradalar bilan

almashtiramiz:

$$X_2 = \begin{array}{cccc}
 \underline{1101} & \underline{1000} & \underline{1010} & \underline{0010} \\
 D & 8 & A & 2
 \end{array}$$

Ikkilik sanoq sistemasida berilgan sondan 8 lik sanoq sistemasiga o'tish uchun, uning o'ng tomonidan boshlab har bir uchliklarni (triadalar) 8 likdagi mos raqamlar bilan almashtiramiz. Masalan

$$X_2 = \begin{array}{cccc} 001010 & 010 & 101 & \\ 1 & 2 & 2 & 5 \end{array} = 1225_3$$

Yuqoridagi X_2 sonini 16 lik sanoq sistemasiga o'tkazish uchun X_2 ni o'ng tomondan boshlab to'rtliklar (tetradalar) bilan almashtiramiz.

$$X_2 = \begin{array}{ccc} 0010 & 1001 & 0101 \\ 2 & 9 & 5 \end{array} = 295_{16}$$

Endi, ixtiyoriy sanoq sistemasidan o'nlik sanoq sistemasiga o'tishning xususiy qoidasini ko'rib o'tamiz. Sakkizlik sanoq sistemasida berilgan sonning 175_8 o'nlik sanoq sistemasidagi ko'rinishini X_{10} topish talab etilsin. Buning uchun

berilgan sonning 8 lik sanoq sistemasidagi yoyilmasini yozib olamiz.

$$X_8 = 175_8 = (1 \cdot 10^2 + 7 \cdot 10^1 + 5 \cdot 10^0)_8$$

va 8 lik sanoq sistemasida $10_8 = 8$ ekanligini hisobga olib topamiz.

$$X_8 = (1 \cdot 8^2 + 7 \cdot 8^1 + 5 \cdot 8^0) = (64 + 56 + 5)_{10} = 125_{10}$$

Xuddi yuqoridagilardek, quyidagi *misollarni* ham qurish mumkin:

$$\begin{aligned} X_{16} = AB_{16} &= (A \cdot 10^1 + B \cdot 10^0)_{16} = (10 \cdot 16^1 + 11 \cdot 16^0)_{10} = (160 + 11)_{10} = 171_{10} \\ X_6 = 154_6 &= (1 \cdot 10^2 + 5 \cdot 10^1 + 4 \cdot 10^0)_6 = (1 \cdot 6^2 + 5 \cdot 6^1 + 4 \cdot 6^0)_{10} = 70_{10} \end{aligned}$$

Shu paytgacha biz butun sonlarni bir sanoq sistemasidan boshqasiga o'tkazish bilan shug'ullandik. Kasr sonlarni bir sanoq sistemasidan ikkinchisiga o'tkazish uchun, uning butun qismi yuqorida keltirilgan qoida, ya'ni bo'lish asosida amalga oshiriladi. Kasr qismini R sanoq sistemasidan Q sanoq sistemasiga o'tkazish uchun kasr sonni Q ga

ketma-ket ko'paytirishda hosil bo'lgan sonning butun kislmlari ketma-ketligi, berilgan son kasr qismining Q sanoq sistemasidagi ko'rinishini hosil qiladi. Misol sifatida o'nlik sanoq sistemasida berilgan $X_{10}=25,205$ sonini 8 lik sanoq sistemasiga o'tkazaylik. Berilgan sonning butun qismi- 25_{10} sakkizlik sanoq sistemasida 41_8 ga teng. Endi kasr qismi 0,205 ni 8 lik sanoq sistemasiga o'tkazamiz. Buning uchun uni ketma-ket 8 ga ko'paytiramiz va hosil bo'lgan butun qismini chiziqning chap tomoniga o'tkazamiz.

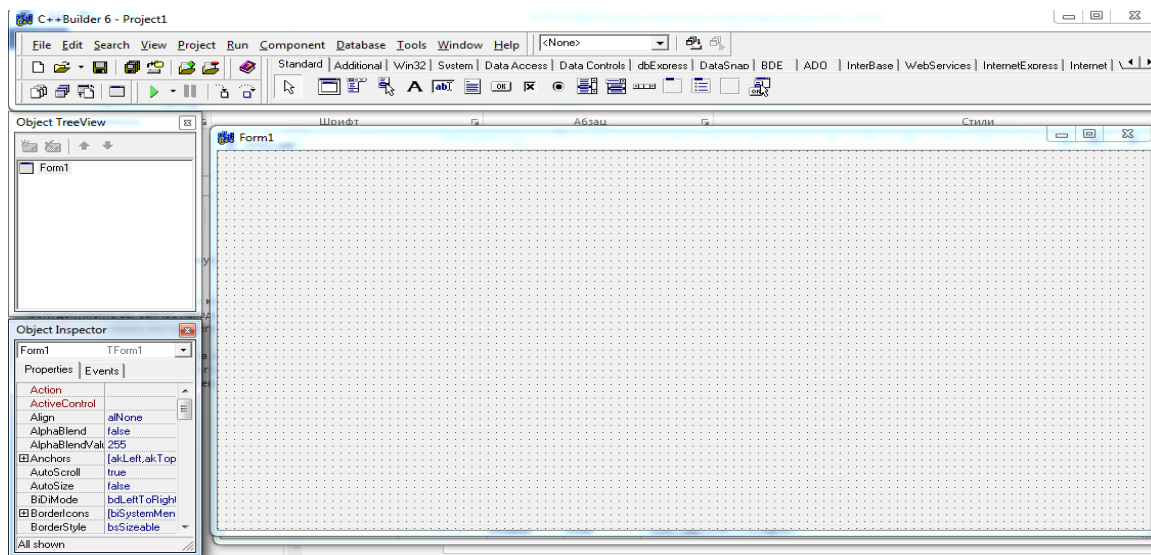
0	0.205
	8
1	0,640
	8
5	0,040
	8
0	0,320
	8
2	0,560

0,205 ni 8 ga ko'paytirganimizda 1,640 hosil bo'ladi va uning butun qismini chiziqning chap tomoniga o'tkazamiz. Keyin 0,640 yana 8 ga ko'paytiramiz va hosil bo'lgan 5,040 sonining butun qismini chiziqning chap tomoniga o'tkazamiz. Ko'paytirishni shu tarzda davom ettiramiz natijada 0,15028 sonini hosil qilamiz va butun qismini 41_8 ni hisobga olib, berilgan $X_{10}=25,205$ sonini 8 lik sanoq sistemasidagi ko'rinishini topamiz:

2. Borland C++ da bir sanoq sestemadan boshqasiga o'tkazuvchi dastur.

Dasturni tuzish jarayoni.

Dastlab C++ Builder dasturlash tilini ishga tayyorlab olamiz. Bu dasturni uchun quyidagi ketma- ketlikda C++ Builder 6 dasturiga kiramiz: **Пуск ->Всепрограмми -> Borland C++ Builder 6 -> C++ Builder 6 .dastur** ishga tushirgach quydagi oyna hosil bo'ladi



Form oynasi hosil bo'lgach unga mos ravishda 2 ta edit, 2 label, 5 ta radiobutton, 5ta button kampanentalarini form oynasiga tashlaymiz. Keyin mos ravishda bu kampanentalarni xossalarni dastur ishlashiga to'g'irlaymiz quydagicha:

Edit1 va edit2 larning text xossasini bo'sh holatga aylantiramiz.

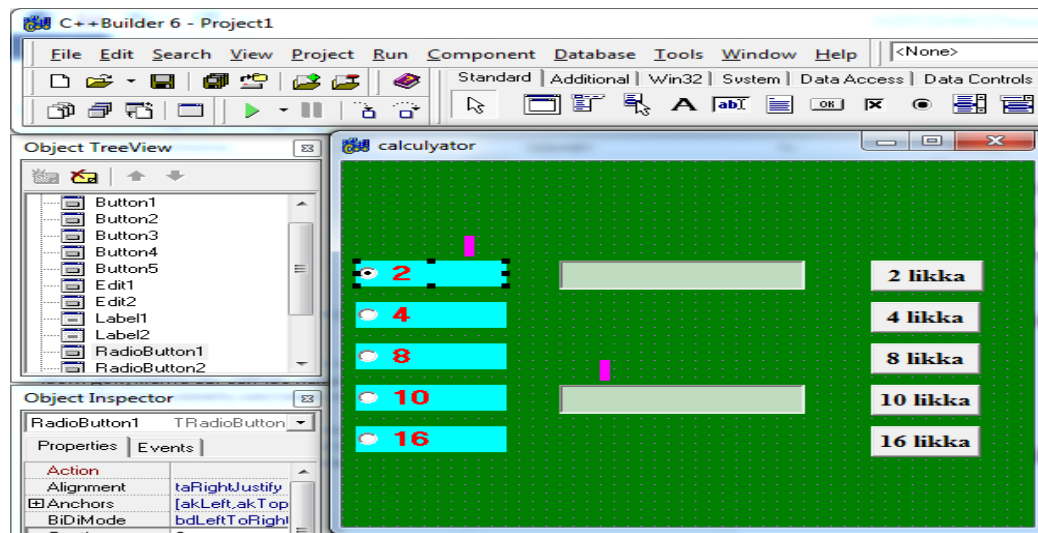
Label1 va label2 larning caption xossasini bo'sh holatga aylantiramiz..

Form1 ni tanlab uning xam vaption xossasini o'zgartirish mumkin va color xossasidan o'zimizga yoqqan rang ni tanlaymiz.

Radiobuttonlarning caption xossasiga mos ravishda ikkilik, to'rtlik, sakkizlik, o'nlik, o'noltilik deb yozib chiqamizva font xossasidan o'zimiz xoxlagan rangni, shrift o'lchamini tanlaymiz. Checked xossasini ham ture (rost) qilamiz bunday qilishga sabab dastur ishlagan vaqtda radiobuttonlarni tanlash mumkin bo'ladi.

Buttonlarni xamcaption xossasiga mos ravishda ikkilikga, to'rtlikga, sakkizlikga, o'nlikga, o'noltilikga deb yozib chiqamiz.

Form oynasiga kerakli kampanentalarni tashlab ularning xossalarini o'zgartirgandan so'ng form oynasi quydagi ko'rinishni egallaydi.



Formani shakillantirib bo'lgach “ikkilik” yani button1 ning kod oynasiga quydagilarni kiritamiz .

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

```
{ Label1->Caption ="ikkilikka" ;
```

```
String u;
```

```
u=Edit1->Text ;
```

```
if (RadioButton1->Checked ==true) Edit2->Text =u;
```

```
String a,s,c ;
```

```
int i,n,k,z,q,m,j;
```

```
if (RadioButton2->Checked==true)
```

```
{ u=Edit1->Text ;
```

```
n=u.Length();
```

```
m=0;
```

```
for (i=1; i<=n; i++ )
```

```
{
```

```
q=StrToInt(u[i]);
```

```
z=1;
```

```
for (k=1; k<=n-i; k++)
```

```
z=z*4;
```

```
m=m+q*z; }
```

```
s=""; k=0;
```

```
while (m!=0) {
```

```
j= m % 2;
```

```
c= IntToStr(j);
```

```
k=k+1;
```

```
s+=c;
```

```
m=m/2; }
```

```

Edit2->Text = "";
for (i=k; i>=1; i--)
Edit2->Text=Edit2->Text+s[i];
}
if (RadioButton3->Checked==true)
{ u=Edit1->Text ;
n=u.Length();
m=0;
for (i=1; i<=n; i++ )
{
q=StrToInt(u[i]);
z=1;
for (k=1; k<=n-i; k++)
z=z*8;
m=m+q*z;
}

s=""; k=0;
while (m!=0)
{
j= m % 2;
c= IntToStr(j);
k=k+1;
s+=c;
m=m/2;
}
Edit2->Text = "";
for (i=k; i>=1; i--)
Edit2->Text=Edit2->Text+s[i];
}
if (RadioButton4->Checked==true)
{ u=Edit1->Text ;
n=u.Length();
m=0;
for (i=1; i<=n; i++ )
{
q=StrToInt(u[i]);
z=1;

```

```

        for (k=1; k<=n-i; k++)
            z=z*10;
        m=m+q*z;}
s=""; k=0;
        while (m!=0      {
            j= m % 2;
            c= IntToStr(j);
            k=k+1;
            s+=c;
            m=m/2; }
Edit2->Text = "";
for (i=k; i>=1; i--)
    Edit2->Text=Edit2->Text+s[i];;}
if (RadioButton5->Checked==true)
{ u=Edit1->Text ;
n=u.Length();
m=0;
for (i=1; i<=n; i++ )
{if (u[i]==char(65)) q=10; else
if (u[i]==char(66)) q=11; else
if (u[i]==char(67)) q=12 ; else
if (u[i]==char(68)) q=13; else
if (u[i]==char(69)) q=14; else
if (u[i]==char(70)) q=15; else
q=StrToInt(u[i]);
z=1;
for (k=1; k<=n-i; k++)
    z=z*16;
    m=m+q*z;}
s=""; k=0;
        while (m!=0){
            j= m % 2;
            c= IntToStr(j);
            k=k+1;
            s+=c;
            m=m/2;
        }
Edit2->Text = "";

```

```

for (i=k; i>=1; i--)
Edit2->Text=Edit2->Text+s[i];;
}

```

“To’rtlik” yani button2 ning kod oynasiga quydagilarni kiritamiz .

```

void __fastcall TForm1::Button2Click(TObject *Sender)

```

```

{ Label1->Caption ="to'rtlikka" ;

```

```

String u;

```

```

u=Edit1->Text ;

```

```

if (RadioButton2->Checked ==true) Edit2->Text =u;

```

```

String a,s,c ;

```

```

int i,n,k,z,q,m,j;

```

```

if (RadioButton1->Checked==true)

```

```

{ u=Edit1->Text ;

```

```

n=u.Length();

```

```

m=0;

```

```

for (i=1; i<=n; i++){

```

```

q=StrToInt(u[i]);

```

```

z=1;

```

```

for (k=1; k<=n-i; k++)

```

```

z=z*2;

```

```

m=m+q*z;}

```

```

s=""; k=0;

```

```

while (m!=0){

```

```

j= m % 4;

```

```

c= IntToStr(j);

```

```

k=k+1;

```

```

s+=c;

```

```

m=m/4;}

```

```

Edit2->Text ="";

```

```

for (i=k; i>=1; i--)

```

```

Edit2->Text=Edit2->Text+s[i];;}

```

```

if (RadioButton3->Checked==true)

```

```

{ u=Edit1->Text ;

```

```

n=u.Length();

```

```

m=0;

```

```

for (i=1; i<=n; i++){

```

```

q=StrToInt(u[i]);

```

```

z=1;

```

```

        for (k=1; k<=n-i; k++)
            z=z*8;
        m=m+q*z;}
s=""; k=0;
    while (m!=0){
        j= m % 4;
        c= IntToStr(j);
        k=k+1;
        s+=c;
        m=m/4;}
    Edit2->Text = "";
    for (i=k; i>=1; i--)
        Edit2->Text=Edit2->Text+s[i];;}
    if (RadioButton4->Checked==true)
    { u=Edit1->Text ;
        n=u.Length();
        m=0;
        for (i=1; i<=n; i++){
            q=StrToInt(u[i]);
            z=1;
            for (k=1; k<=n-i; k++)
                z=z*10;
            m=m+q*z;}
s=""; k=0;
    while (m!=0){
        j= m % 4;
        c= IntToStr(j);
        k=k+1;
        s+=c;
        m=m/4;        }
    Edit2->Text = "";
    for (i=k; i>=1; i--)
        Edit2->Text=Edit2->Text+s[i];;}
    if (RadioButton5->Checked==true)
    { u=Edit1->Text ;
        n=u.Length();
        m=0;
        for (i=1; i<=n; i++ )

```

```

        {if (u[i]==char(65)) q=10; else
        if (u[i]==char(66)) q=11; else
        if (u[i]==char(67)) q=12 ; else
        if (u[i]==char(68)) q=13; else
        if (u[i]==char(69)) q=14; else
        if (u[i]==char(70)) q=15; else
        q=StrToInt(u[i]);
        z=1;
        for (k=1; k<=n-i; k++)
        z=z*16;
        m=m+q*z;}
s=""; k=0;
    while (m!=0){
        j= m % 4;
        c= IntToStr(j);
        k=k+1;
        s+=c;
        m=m/4;}
    Edit2->Text ="";
    for (i=k; i>=1; i--)
    Edit2->Text=Edit2->Text+s[i];;
}

```

“Sakkizlik” yani button3 ning kod oynasiga quydagilarni kiritamiz .

```

void __fastcall TForm1::Button3Click(TObject *Sender)
{ Label1->Caption ="sakkizlikka" ;
String u;
    u=Edit1->Text ;
    if (RadioButton3->Checked ==true) Edit2->Text =u;
    String a,s,c ;
    int i,n,k,z,q,m,j;
if (RadioButton1->Checked==true)
    { u=Edit1->Text ;
    n=u.Length();
    m=0;
    for (i=1; i<=n; i++){
        q=StrToInt(u[i]);
        z=1;

```



```

        for (k=1; k<=n-i; k++)
            z=z*2;
        m=m+q*z;          }
s=""; k=0;
    while (m!=0){
        j= m % 8;
        c= IntToStr(j);
        k=k+1;
        s+=c;
        m=m/8;}
    Edit2->Text = "";
    for (i=k; i>=1; i--)
        Edit2->Text
            while (m!=0){
                j= m % 8;
                c= IntToStr(j);
                k=k+1;
                s+=c;
                m=m/8;}
        Edit2->Text = "";
        for (i=k; i>=1; i--)
            Edit2->Text=Edit2->Text+s[i];;}
    if (RadioButton4->Checked==true)
        { u=Edit1->Text ;
          n=u.Length();
          m=0;
          for (i=1; i<=n; i++){
              q=StrToInt(u[i]);
              z=1;
              for (k=1; k<=n-i; k++)
                  z=z*10;
              m=m+q*z;}
          s=""; k=0;
          while (m!=0){
              j= m % 8;
              c= IntToStr(j);
              k=k+1;
              s+=c;

```

```

m=m/8;
}
Edit2->Text = "";
for (i=k; i>=1; i--)
Edit2->Text=Edit2->Text+s[i];
}
if (RadioButton5->Checked==true)
{ u=Edit1->Text ;
n=u.Length();
m=0;
for (i=1; i<=n; i++ )
{if (u[i]==char(65)) q=10; else
if (u[i]==char(66)) q=11; else
if (u[i]==char(67)) q=12 ; else
if (u[i]==char(68)) q=13; else
if (u[i]==char(69)) q=14; else
if (u[i]==char(70)) q=15; else
q=StrToInt(u[i]);
z=1;
for (k=1; k<=n-i; k++)
z=z*16;
m=m+q*z;}
s=""; k=0;
while (m!=0){
j= m % 8;
c= IntToStr(j);
k=k+1;
s+=c;
m=m/8;}
Edit2->Text = "";
for (i=k; i>=1; i--)
Edit2->Text=Edit2->Text+s[i];
}

```

“O’nlik” yani button1 ning kod oynasiga quydagilarni kiritamiz .

```

void __fastcall TForm1::Button4Click(TObject *Sender)
{ Label1->Caption ="o'nlikka" ;
String a,s,c,u,jj ;
int i,n,k,z,q,m,j;

```

```

if (RadioButton1->Checked==true)
{ u=Edit1->Text ;
n=u.Length();
m=0;
for (i=1; i<=n; i++){
q=StrToInt(u[i]);
z=1;
for (k=1; k<=n-i; k++)
z=z*2;
m=m+q*z;
Edit2->Text =IntToStr(m);
}} else if (RadioButton2->Checked==true)
{ u=Edit1->Text ;
n=u.Length();
m=0;
for (i=1; i<=n; i++){
q=StrToInt(u[i]);
z=1;
for (k=1; k<=n-i; k++)
z=z*4;
m=m+q*z;
Edit2->Text =IntToStr(m);
}}if (RadioButton3->Checked==true)
{ u=Edit1->Text ;
n=u.Length();
m=0;
for (i=1; i<=n; i++){
q=StrToInt(u[i]);
z=1;
for (k=1; k<=n-i; k++)
z=z*8;
m=m+q*z;
Edit2->Text =IntToStr(m);
}} else if (RadioButton5->Checked==true)
{ u=Edit1->Text ;
n=u.Length();
m=0;
for (i=1; i<=n; i++ )

```

```

{ if (u[i]==char(65)) q=10; else
if (u[i]==char(66)) q=11; else
if (u[i]==char(67)) q=12 ; else
if (u[i]==char(68)) q=13; else
if (u[i]==char(69)) q=14; else
if (u[i]==char(70)) q=15; else
q=StrToInt(u[i]);
z=1;
for (k=1; k<=n-i; k++)
z=z*16;
m=m+q*z;
Edit2->Text =IntToStr(m);
}} if (RadioButton4->Checked==true) Edit2->Text =Edit1->Text ;}
void __fastcall TForm1::Button5Click(TObject *Sender)
{ Label1->Caption ="o'noltilikka" ;
String u;
u=Edit1->Text ;

```

“O’ noltilik” yani button5 ning kod oynasiga quydagilarni kiritamiz .

```

if (RadioButton5->Checked ==true) Edit2->Text =u;
String a,s,c ;
int i,n,k,z,q,m,j;
if (RadioButton1->Checked==true)
{ u=Edit1->Text ;
n=u.Length();
m=0;
for (i=1; i<=n; i++ )      {
q=StrToInt(u[i]);
z=1;
for (k=1; k<=n-i; k++)
z=z*2;
m=m+q*z;                }
s=""; k=0;
while (m!=0)             {
j= m % 16;
if (j==10) c="A"; else
if (j==11) c="B"; else
if (j==12) c="C"; else

```

```

if (j==13) c="D"; else
if (j==14) c="E"; else
if (j==15) c="F"; else
c= IntToStr(j);
k=k+1;
s+=c;
m=m/16;          }
Edit2->Text = "";
for (i=k; i>=1; i--)
Edit2->Text=Edit2->Text+s[i];;}
if (RadioButton2->Checked==true)
{ u=Edit1->Text ;
n=u.Length();
m=0;
for (i=1; i<=n; i++ )      {
q=StrToInt(u[i]);
z=1;
for (k=1; k<=n-i; k++)
z=z*4;
m=m+q*z;          }
s=""; k=0;
while (m!=0)          {
j= m % 16;
if (j==10) c="A"; else
if (j==11) c="B"; else
if (j==12) c="C"; else
if (j==13) c="D"; else
if (j==14) c="E"; else
if (j==15) c="F"; else
c= IntToStr(j);
k=k+1;
s+=c;
m=m/16;          }
Edit2->Text = "";
for (i=k; i>=1; i--)
Edit2->Text=Edit2->Text+s[i];;}
if (RadioButton3->Checked==true)
{ u=Edit1->Text ;

```

```

n=u.Length();
m=0;
for (i=1; i<=n; i++ )
{
q=StrToInt(u[i]);
z=1;
for (k=1; k<=n-i; k++)
z=z*8;
m=m+q*z;      }
s=""; k=0;
while (m!=0)      {
j= m % 16;
if (j==10) c="A"; else
if (j==11) c="B"; else
if (j==12) c="C"; else
if (j==13) c="D"; else
if (j==14) c="E"; else
if (j==15) c="F"; else
c= IntToStr(j);
k=k+1;
s+=c;
m=m/16;
}
Edit2->Text = "";
for (i=k; i>=1; i--)
Edit2->Text=Edit2->Text+s[i];
}
if (RadioButton4->Checked==true)
{ u=Edit1->Text ;
n=u.Length();
m=0;
for (i=1; i<=n; i++ )
{
q=StrToInt(u[i]);
z=1;
for (k=1; k<=n-i; k++)
z=z*10;
m=m+q*z;      }

```

```

s=""; k=0;
while (m!=0)      {
j= m % 16;
if (j==10) c="A"; else
if (j==11) c="B"; else
if (j==12) c="C"; else
if (j==13) c="D"; else
if (j==14) c="E"; else
if (j==15) c="F"; else
c= IntToStr(j);
k=k+1;
s+=c;
m=m/16;
}
Edit2->Text ="";
for (i=k; i>=1; i--)
Edit2->Text=Edit2->Text+s[i];
}

```

Qaysi sanoq sestemada edit1 ga raqam kiritish va qaysi sanoq sestemani tanlashimizga qarab qaysi raqamlar kiritilishi label2 ning captionda xossasida chiqishi uchun radiobuttonning click xossasiga quydagilarni kiritamiz

```

void __fastcall TForm1::RadioButton1Click(TObject *Sender)
{
Label2->Caption ="0 dan 1 gacha raqamlar kiriting";
}
//-----
void __fastcall TForm1::RadioButton2Click(TObject *Sender)
{
Label2->Caption ="0 dan 3 gacha raqamlar kiriting";
}
//-----
void __fastcall TForm1::RadioButton3Click(TObject *Sender)
{
Label2->Caption ="0 dan 7 gacha raqamlar kiriting";
}
//-----
void __fastcall TForm1::RadioButton4Click(TObject *Sender)

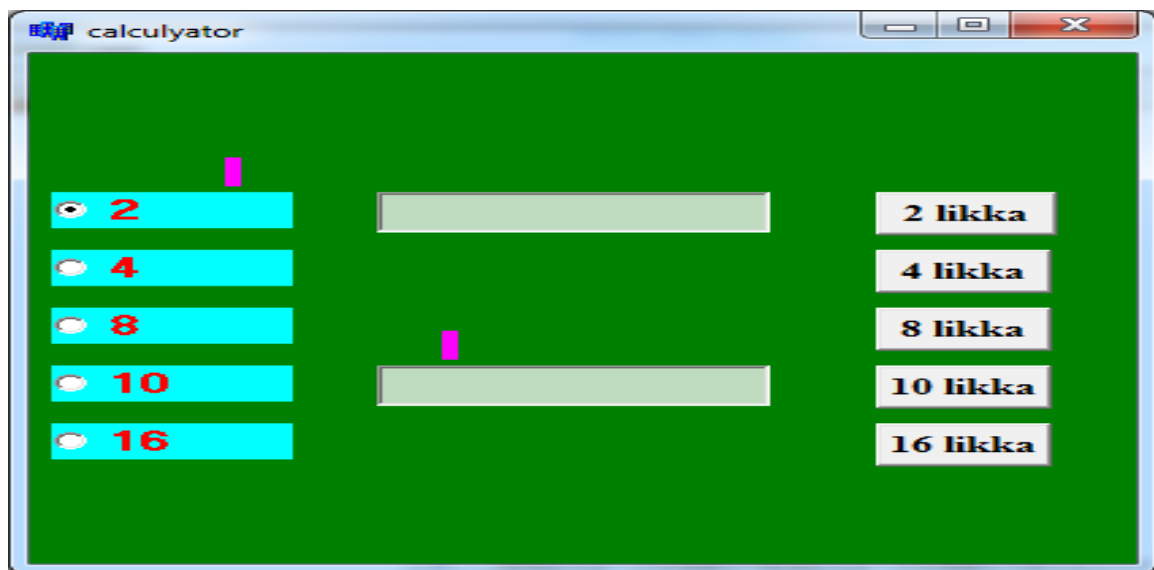
```

```

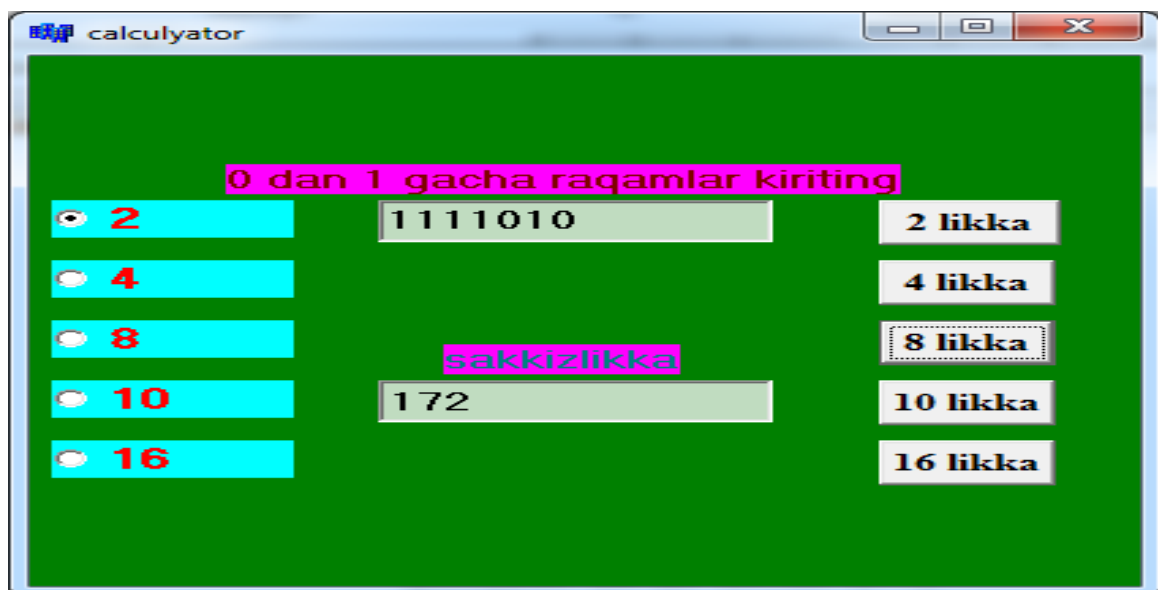
{
Label2->Caption ="0 dan 9 gacha raqamlar kiriting";
}
//-----
void __fastcall TForm1::RadioButton5Click(TObject *Sender)
{
Label2->Caption ="0 dan 9 gacha va A,B,C,D,E,F larni kiriting ";
}

```

Barcha kodlarni kod oynasiga kiritgach tuzgan dasturimizni ishga tushiramiz



Dasturni ishga tushurgach dastur tog'ri ishlashini sinab ko'ramiz ikkilikni tanlab ikkilikda son kiritamiz va javobni skkizlikda chiqaramiz



Ushbu dasturi taminot yordamida ikkilik, to'rtlik, sakkizlik, o'nlik, o'noltilik sanoq sestemadagi sonlarni boshqasiga o'tkazishga moslashgan.

Xulosa.

Men ushbu dasturni ishlab chiqish jarayonida C++ builder obektlar bilan ishlash muhitida turli kampanentalar bilan ishlashni o'rgandim. C++ builder obektlar bilan ishlash muhitida nafaqat berilgan masalani yechish dasturini tuzishni balki bu yerda xuddi hisoblash qurilmasida ishlayotgandek obektlar bilan ishladim, bu obektga o'zim xoxlagandek dizayn berib, obekt yaratdim.

Bu dasturni tuzib obektlar yaratishni va ular orqali shunga o'xshash yana boshqa obektlar yaratmoqchiman bu dasturda ishlashimga yana bir sabab boshqa obektlarga mo'ljallanmagan dasturlash tillarida faqat masalani matematik dasturi tuzib natija olinadi bunda yani C++ builderda esa boshqa imkoniyatlar va kampanentalar bilan ham ishlash imkoni o'rgandim.

Foydalanilganadabiyotlarro'yhati

1. СтрауструпБ. Язык программирования С++. Третье издание, М.: Бином, 1999.
2. Шмидский Я.К.Программирование на языке С++: Самоучитель. Учебное пособие. Диалектика. 361 стр, 2004 г.
3. Q.Abduraximov С++ Dasturlash asoslari 2014
4. Ашарина Н.А. Основы программирования на языках Си,С++. Учебный курс.М.: 2002 г.
5. Подбельский В.В. Язык С++ М.: Финансы и статистика, 1996.
6. www.tammi.uz
7. www.ziyonet.uz
8. www.google.uz qidiruv sayti