

O'ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA
KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI

MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT AXBOROT
TEXNOLOGIYALARI UNIVERSITETI
FARG'ONA FILIALI

KOMPYUTER INJINIRINGI FAKULTETI

"Ma'lumotlar tuzilmasi"

fanidan

Laboratoriya ishlari bo'yicha

USLUBIY KO'RSATMA

5330500 – "Kompyuter injiniringi" ("Kompyuter injiniringi", "Axborot xavfsizligi",
"AT-xizmat", "Multimediya texnologiyalari") bakalavr yo'nalishidagi talabalar
uchun.

Farg'ona- 2017

1



Uslubiy ko`rsatma O`zbekiston Respublikasi Oliy ta`lim va o`rta – maxsus ta`lim vazirligi tomonidan tasdiqlangan na`munaviy o`quv rejaga binoan tuzilgan. Buyruq № _____” _____” _____yil.

Tuzuvchilar:

Dildhodov A – TATUFF “Kompyuter tizimlari” kafedra kata o`qituvchisi.

Akbarov N – TATUFF “Kompyuter tizimlari” kafedra asistenti.

Taqrizchilar:

Abduqodirov A – TATUFF “Axborot texnologiyalari” kafedra mudiri.

Uslubiy ko`rsatma “Kompyuter tizimlari” kafedra majlisida tasdiqlangan.
Qaydnoma № ____ 2017 yil.

Kafedra mudiri _____ Djalilov M.L

Uslubiy ko`rsatma “Kompyuter injiniringi” fakultetining ilmiy – uslubiy kengashi tomonidan tasdiqlangan.

Qaydnoma № _____” _____” _____2017 yil

Fakultet kengashi raisi _____ Tojiboyev I

MUNDARIJA:

1-LABORATORIYA ISHI.....	6
2-LABORATORIYA ISHI.....	11
3-LABORATORIYA ISHI.....	15
4-LABORATORIYA ISHI.....	20
5-LABORATORIYA ISHI.....	25
6-LABORATORIYA ISHI.....	34
7-LABORATORIYA ISHI.....	37
8-LABORATORIYA ISHI.....	40
9-LABORATORIYA ISHI.....	44
FOYDALANILGAN ADABIYOTLAR.....	48

1-LABORATORIYA ISHI.

MAVZU: Ma'lumotlarning oddiy sozlangan turlari.

Ishdan maqsad: Ma'lumotlarning oddiy sozlangan, unar va binar amallar, sikl va takrorlash operatorlarini o'rganish va ularni tadqiq qilish.

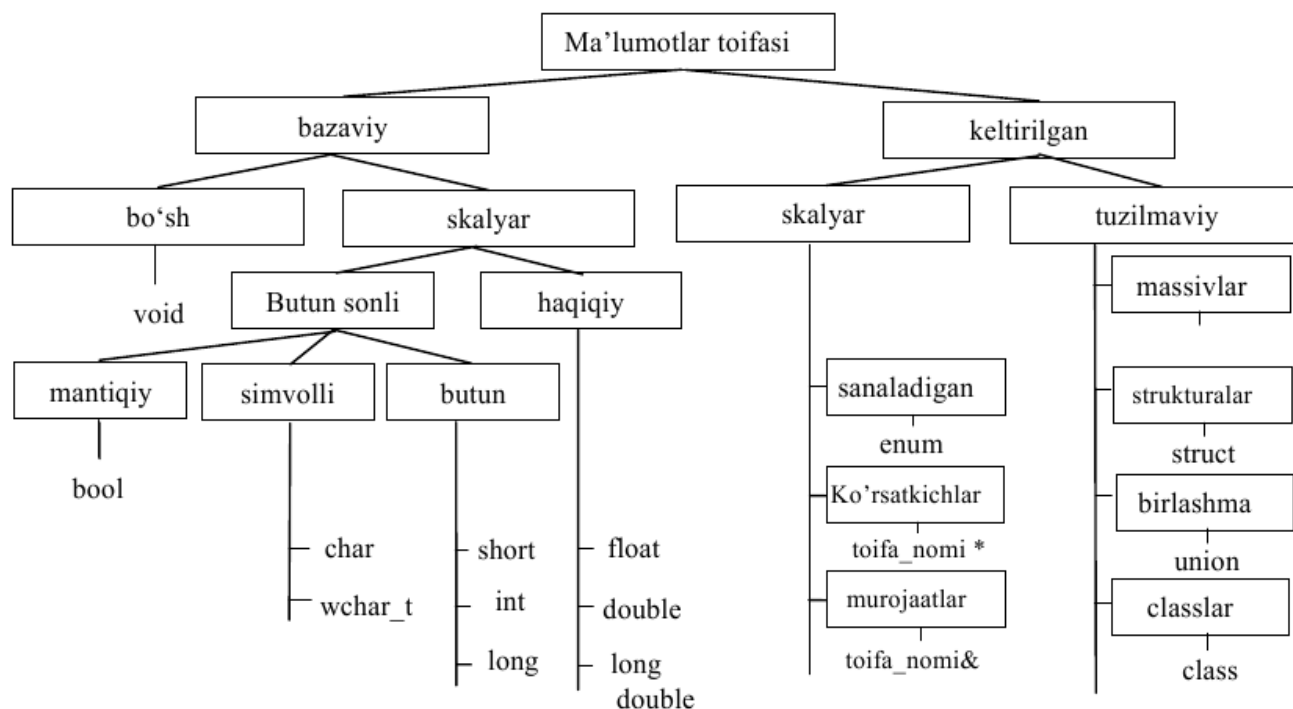
Qo'yilgan masala: C++ tilida butun, haqiqiy, belgili, mantiqiy toifadagi ma'lumotlarni e'lon qilish, unar va binar amallarni, sikl va takrorlash operatorlarini qo'llash va ularga doir misollarning dasturini ishlab chiqish.

Ish tartibi:

- Tajriba ishi nazariy ma'lumotlarini o'rganish;
- Berilgan topshiriqning algoritmini ishlab chiqish;
- C++ dasturlash muhitida dasturni yaratish;
- Natijalarni tekshirish;
- Hisobotni tayyorlash va topshirish.

Ma'lumotlar toifalari

Ko'plab dasturlash tillarida ma'lumotlar bazaviy va keltirilgan toifalarga ajratiladi. Ma'lumotlarning toifalarini 1.1-rasmdagidek klassifikatsiyalash mumkin.



1.1-rasm. Toifalarni klassifikatsiyasi.

Ma'lumotlarning ixtiyoriy toifasi qiymatlar sohasi va ular ustida bajarilishi mumkin bo'lgan amallar orqali tavsiflanadi. void kalit so'zi hech qanday toifaga ega emaslikni anglatadi. Bunday toifadagi funksiyalar hech qanday qiymatni qaytarmaydi. Lekin asosiy dastur tanasi, ya'ni main() funksiyasi void toifasiga ega bo'lolmaydi, u int toifasida bo'lishi kerak.

Butun toifa - int

Mazkur toifa butun sonlar to'plamining qandaydir qism to'plami bo'lib, uning o'lchami mashina, ya'ni kompyuter konfiguratsiyasiga bog'liq ravishda o'zgarib turadi. Mazkur toifaga kiruvchi sonlar ikkiga bo'linadi: ishorali (signed) va ishorasiz (unsigned). Sonlarni xotirada tasvirlashda eng chapdagi bit ishora uchun belgilanadi. Toifalarni signed (ishorali), unsigned (ishorasiz) kalit so'zlari bilan modifikatsiyalash mumkin. Bunda ishorali toifa uchun ajratilgan joyning eng chap biti ishora uchun, qolgan bitlar qiymatlarni saqlash uchun ishlatiladi, ya'ni 0 - plus, 1 - minus. Ishorasiz toifalarda esa barcha bitlar qiymatlarni saqlash uchun ishlatiladi. Ularning har biri uchun mos ravishda qiymat qabul qilish oralig'i mavjud:

1. Ishorasiz sonlar uchun ($0 \dots 2^n - 1$); 2. Ishoralilar uchun ($-2^{n-1} \dots 2^{n-1} - 1$).

Butun sonlar ustida turli matematik (+, -, /, *) va solishtirish amallarini bajarish mumkin, ya'ni ==, !=, <, <=, >, >= operatorlar bilan binar amallarni bajarish mumkin. Ammo bu operatsiyalarning natijalari int toifasiga kirmaydi, ular bool toifasiga kiradi.

Butun qiymat qabul qiluvchi o'zgaruvchilarni e'lon qilish uchun int, short int, long int xizmatchi so'zlaridan foydalanish mumkin. Butun qiymatli toifalarning barchasi 1.1-jadvalda keltirilgan:

Toifa ko'rinishi	Mazkur toifadagi o'zgaruvchining qabul qiladigan qiymatlar oralig'i	O'zgaruvchining kompyuter xotirasidan egallaydigan joyi
short int	signed: -32768 ... 32767 unsigned: 0 ... 65535	2 bayt
int	signed: -2147483648 ... 2147483647 unsigned: 0 ... 4294967295	4 bayt
long int	signed: -2147483648 ... 2147483647 unsigned: 0 ... 4294967295	4 bayt

1.1-jadval

Bu sanab o'tilgan toifalar o'zlarining qiymatlar qabul qilish oralig'i va xotiradan egallagan joyining katta yoki kichikligi bilan farqlanadi. Shuning uchun, o'zgaruvchilarning qabul qiladigan qiymatlarini katta yoki kichikligiga qarab, yuqoridagi toifalardan mosini tanlash maqsadga muvofiqdir.

Haqiqiy toifa

Haqiqiy toifaga kasr qismlari bor chekli sonlar to'plami kiradi. Haqiqiy sonlar ustida turli matematik amallarni bajarish mumkin. Bu amallarning natijalari ham haqiqiy toifaga kiradi. Bu yerda ham binar amallarga nisbatan masalaning yechimlari mantiqiy toifaga tegishli bo'ladi.

Kompyuter xotirasida haqiqiy sonlar asosan qo'zg'aluvchan nuqta formatida saqlanadi.

1) $937,56 = 93756 * 10^{-2} = 0,93756 * 10^3 = 0,93756E+3$

2) $0,002355 = 2,355 * 10^{-3} = 2,355E-3$

Haqiqiy (kasr) qiymatli toifaga tegishli o'zgaruvchilarni e'lon qilish uchun float, double, long double xizmatchi so'zlaridan foydalanish mumkin.

Haqiqiy toifa shakllari

Toifa ko'rinishi	Mazkur toifadagi o'zgaruvchining qabul qiladigan qiymat oralig'i	O'zgaruvchining kompyuter xotirasidan egallaydigan joyi
Float	$\pm 3.4E-38 \dots \pm 3.4E+38$	4 bayt
Double	$\pm 1.7E-308 \dots \pm 1.7E+308$	8 bayt
long double	$\pm 1.7E-308 \dots \pm 1.7E+308$	1 bayt

1.2-jadval.

Amallar odatda **unar** ya'ni bitta operandga qo'llaniladigan amallarga va **binar** ya'ni ikki operandga qo'llaniladigan amallarga ajratiladi.

Binar amallar additiv ya'ni qo'shuv $+$ va ayirish $-$ amallariga, hamda multiplikativ ya'ni ko'paytirish $*$, bo'lish $/$ va modul olish $\%$ amallariga ajratiladi. Additiv amallarining ustuvorligi multiplikativ amallarining ustuvorligidan pastroqdir. Butun sonni butun songa bo'lganda natija butun songacha yaxlitlanadi. Masalan, $10/3=3$, $(-10)/3=-3$, $10/(-3)=-3$.

Modul amali butun sonni butun songa bo'lishdan hosil bo'ladigan qoldiqqa tengdir. Agar modul amali musbat operandlarga qo'llanilsa, natija ham musbat bo'ladi, aks holda natija ishorasi kompilyatorga bog'liqdir.

Binar arifmetik amallar bajarilganda tiplarni keltirish quyidagi qoidalar asosida amalga oshiriladi:

- short va char tiplari int tipiga keltiriladi;
- agar operandlar biri long tipiga tegishli bo'lsa ikkinchi operand ham long tipiga keltiriladi va natija ham long tipiga tegishli bo'ladi;
- agar operandlar biri float tipiga tegishli bo'lsa ikkinchi operand ham float tipiga keltiriladi va natija ham float tipiga tegishli bo'ladi;
- agar operandlar biri double tipiga tegishli bo'lsa ikkinchi operand ham double tipiga keltiriladi va natija ham double tipiga tegishli bo'ladi;
- agar operandlar biri long double tipiga tegishli bo'lsa ikkinchi operand ham long double tipiga keltiriladi va natija ham long double tipiga tegishli bo'ladi;

Unar amallarga ishorani o'zgartiruvchi unar minus $-$ va unar plus $+$ amallari kiradi. Bundan tashqari $++$ va $--$ amallari ham unar amallarga kiradi.

$++$ unar amali qiymatni 1 ga oshirishni ko'rsatadi. Amalni prefiks ya'ni $++i$ ko'rinishda ishlatish oldin o'zgaruvchi qiymatini oshirib so'ngra foydalanish lozimligini, postfiks ya'ni $i++$ ko'rinishda ishlatishdan oldin o'zgaruvchi qiymatidan foydalanib, so'ngra oshirish kerakligini ko'rsatadi. Masalan, i qiymati 2 ga teng bo'lsin, u holda $3+(++i)$ ifoda qiymati 6 ga, $3+i++$ ifoda qiymati 5 ga teng bo'ladi. Ikkala holda ham i qiymati 3 ga teng bo'ladi.

$--$ unar amali qiymatni 1 ga kamaytirishni ko'rsatadi. Bu amal ham prefiks va postfiks ko'rinishda ishlatilishi mumkin. Masalan, i qiymati 2 ga teng bo'lsin, u holda $--i$ ifoda qiymati 1 ga, $i--$ ifoda qiymati 2 ga teng bo'ladi. Ikkala holda ham i qiymati 1 ga teng bo'ladi.

Masala: Ushbu masalalarni C++ dasturlash tilida `int a=100, b=10, c=5`; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga `a=4` va `b=810` natija chiqsin

Dasturning ko'di:

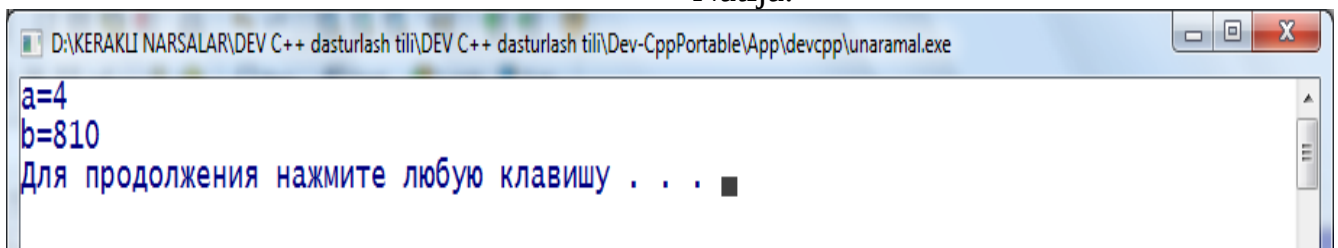
```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int a=100,b=10,c=5;
    for(int i=0; i<2; i++)
    {
        a/=b-c;
    }
    for(int i=0; i<2; i++)
    {
        b*=a+c;
    }
    cout<<"a="<<a<<endl;
    cout<<"b="<<b<<endl;

    system("PAUSE");
}
```

Natija:



1.1-rasm.

Nazorat savollari:

- 1 C++ tiliga amallar necha gurunga bo`linadi?
- 2 Arifmetik amallarning ba`zi xususiyatlarni ayting ?
- 3 Amallar bilan ishlaganda qavslarning o`rni qanday ?
- 4 Unary amalga misol keltiring ?
- 5 ++33 qanday amalni bajaradi ?
- 6 33++ qanday amalni bajaradi ?
- 7 Short va char tiplarini qaysi tipga keltirib bo`ladi ?
- 8 Shartli amaldan qachon foydalanish mumkin?

Ushbu laboratoriya ishini bajarish davomida talabalar quyidagi masalalarni o'zining jurnalidagi tartib raqami bo'yicha yechishi lozim:

1. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=4 va b=810 natija chiqsin.
2. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=400 va b=64000 natija chiqsin.
3. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=400 va b=170 natija chiqsin.
4. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=104 va b=50 natija chiqsin.
5. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=104, b=50 va c=313 natija chiqsin.
6. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=104, b=50 va c=113 natija chiqsin.
7. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=104, b=50 va c=14580 natija chiqsin.
8. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=4 natija chiqsin.
9. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=2500 natija chiqsin.
10. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=90 natija chiqsin.
11. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=110 natija chiqsin.
12. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=115 natija chiqsin.
13. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=75 natija chiqsin.
14. C++ dasturlash tilida int a=100, b=10; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=80 natija chiqsin.
15. C++ dasturlash tilida int a=100, b=10; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=120 natija chiqsin.
16. C++ dasturlash tilida int a=100, b=10; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=10000 natija chiqsin.
17. C++ dasturlash tilida int a=100, b=10; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=10 natija chiqsin.
18. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=22500 natija chiqsin.
19. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=400 natija chiqsin.
20. C++ dasturlash tilida int a=100, b=10, c=5; qiymatlar berilgan. Sikl, unar va arifmetik amallar yordamida ekranga a=250000 natija chiqsin.

2-LABORATORIYA ISHI.

MAVZU: Yarimstatik turdagi ma'lumotlar tuzilmasi

Ishdan maqsad: Ma'lumotlarning mantiqiy va belgili turlarini, sikl va massivni, gets() va strlen() funksiyalarini o'rganish va ularni tadqiq qilish.

Qo'yilgan masala: C++ tilida belgili, mantiqiy turdagi ma'lumotlarni e'lon qilish, sikl, massivlar, gets() va strlen() funksiyasini qo'llash va ularga doir misollarning dasturini ishlab chiqish.

Ish tartibi:

- Tajriba ishi nazariy ma'lumotlarini o'rganish;
- Berilgan topshiriqning algoritmini ishlab chiqish;
- C++ dasturlash muhitida dasturni yaratish;
- Natijalarni tekshirish;
- Hisobotni tayyorlash va topshirish.

Mantiqiy tur.

Mazkur tur mantiqiy mulohazalarning to'g'riligini aniqlash uchun, turli xil dasturlash tillarida turlicha ifodalaniladigan ifodalarni 2 ta ko'rinishda aniqlaydi. Mantiqiy ma'lumotlar ustida quyidagi mantiqiy operatsiyalarni bajarish mumkin: konyunktsiya (va), dizyunktsiya (yoki) va inkor (yo'q), hamda qiyinroq bo'lgan ekvivalentlik, implikatsiya, chiqarib tashlash va boshqa operatsiyalar. Yuqorida keltirilgan ixtiyoriy operatsiyaning natijasi - mantiqiy qiymatga ega bo'ladi. Mantiqiy qiymatni xotirada saqlash uchun bitta bit yetarli.

Asosiy mantiqiy funksiyalarning chinlik jadvali

A	B	not A	A or B	A and B
1	1	0	1	1
1	0	0	1	0
0	1	1	1	0
0	0	1	0	0

1.1-jadval.

Mantiqiy tur tavsifi

Tur ko'rinishi	Mazkur turdagi o'zgaruvchining qabul qiladigan qiymat oralig'i	O'zgaruvchining kompyuter xotirasidan egallaydigan joyi
Bool	true , false	1 bayt

1.2-jadval.

C++ da and mantiqiy amalining yana bir yozilish shakli &&, or yoki , not yoki ! va “inkor-yoki” amali xor kabi yozilishi mumkin. bool turiga bitta misol ko’rib chiqamiz.

```
#include <iostream>
using namespace std; int main()

{
    bool b=true;
bool s=false;
    bool d1 =not b || s;
    bool d2=b && s; bool d3=b xor s;
    cout<<d1<<" "<<d2<<" "<<d3;
    system ("PA USE ");
}
Natija: 0 0 1
```

Belgili tur

Belgili turga belgilarning chekli to’plami yoki liter, ularga lotin alifbosidagi harflar va unda yo’q kirill harflar, o’nlik raqamlar, matematik va maxsus belgilar kiradi. Belgili ma’lumotlar hisoblash texnikasi bilan inson o’rtasidagi aloqani o’rnatishda katta ahamiyatga ega. Belgili turdagi o’zgaruvchilar ustida turli matematik amallarni bajarish mumkin. Bunda amallar belgilarning ASCII kodlari ustida bajariladi. Shu sababli, belgili turlarni taqqoslash ham mumkin va taqqoslashlarning natijalari bool tursiga kiradi. C++ tilida belgili turlarning qiymatlari qo’shtirnoq ichida beriladi va u bitta belgidan iborat bo’lishi mumkin.

Belgili tur shakllari

Tur ko’rinishi	Mazkur turdagi o’zgaruvchining qabul qiladigan qiymat oralig’i	O’zgaruvchining kompyuter xotirasidan egallaydigan joyi
Char(signedchar)	-128...127	1 bayt
Unsigned char	0...255	1 bayt
Wchar_t	0...65535	2 bayt

1.3-jadval.

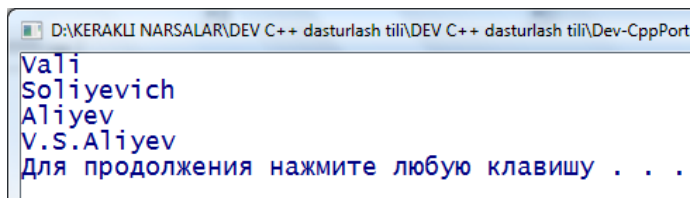
Satr (qator) - bu qandaydir belgilar ketma-ketligi bo’lib, satr bitta, bo’sh yoki bir nechta belgilar birlashmasidan iborat bo’lishi mumkin. C++ tilida satrlarni e’lon qilish belgilar massivi shaklida amalga oshiriladi. Bu haqda keyinroq batafsil to’xtalamiz.

Belgili turdagi o’zgaruvchilar ustida o’zlashtirish, taqqoslash va turli matematik amallarni bajarish mumkin. Bunda agar belgili turlar ustida matematik amallar bajariladigan bo’lsa, belgilarning ASCII kodlari olinadi.

Belgilar va qatorlarga doir quyidagi sodda dasturni keltiramiz:
1-misol. Belgili turda massivni qo'llash.

```
#include <string>
{
    char i[30], o[30], f[30];
    w: gets (i);
    gets (o);
    gets (f);
    cout<<i[0]<<"."<<o[0]<<"."<<f<<endl;
    system("PAUSE");
    goto w;
}
```

Natija:

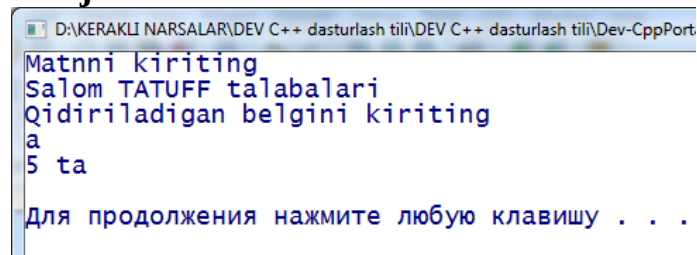


```
D:\KERAKLI NARSALAR\DEV C++ dasturlash tili\DEV C++ dasturlash tili\Dev-CppPort
Vali
Soliyevich
Aliyev
V.S.Aliyev
Для продолжения нажмите любую клавишу . . .
```

2-misol. Kiritilgan matndan ixtiyoriy belgini qidirish.

```
char a[30];
char x;
    cout<<"Matnni kiriting "<<endl;
    gets (a);
    cout<<"Qidiriladigan belgini kiriting "<<endl;
    cin>>x;
    int b=0;
    for (int i=0; i<strlen(a); i++)
    {    if (a[i]==x)
    {    b++;    }
    }
    cout<<b<<" ta"<<endl;
    cout<<endl;
```

Natija:

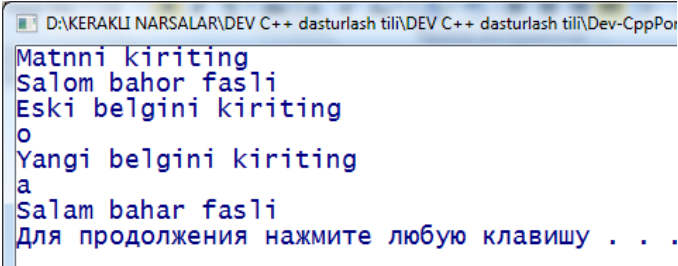


```
D:\KERAKLI NARSALAR\DEV C++ dasturlash tili\DEV C++ dasturlash tili\Dev-CppPort
Matnni kiriting
Salom TATUFF talabarlari
Qidiriladigan belgini kiriting
a
5 ta
Для продолжения нажмите любую клавишу . . .
```

3-misol. Kiritilgan matndan ixtiyoriy belgini almashtirish.

```
char a[30];
char b, c;
cout<<"Matnni kiriting "<<endl;
gets (a);
cout<<"Eski belgini kiriting "<<endl;
cin>>b;
cout<<"Yangi belgini kiriting "<<endl;
cin>>c;
for (int i=0; i<strlen(a); i++)
{
if (a[i]==b)
{
a[i]=c;
} }
cout<<a<<endl;
```

Natija:



```
D:\KERAKLI NARSALAR\DEV C++ dasturlash tili\DEV C++ dasturlash tili\Dev-CppPor
Matnni kiriting
Salom bahor fasli
Eski belgini kiriting
o
Yangi belgini kiriting
a
Salam bahar fasli
Для продолжения нажмите любую клавишу . . .
```

2-laboratoriya ishida talabalar quyidagi topshiriqlarni bajarishlari lozim.

1. Har bir talaba 1-misolimizdagi kabi o'zlarining familiya, ismi va otasining ismidan foydalanib quyidagicha natija chiqarishi kerak. (Ibragimov Ahadjon Ahamadullo o'g'li deb kiritilsa, A.A.Ibragimov holatiga o'tishi kerak).
2. Har bir talaba o'zining ismini kiritib teskarisini topishi kerak.(Ahadjon deb kiritilsa, nojdahA holatiga o'tishi kerak).
3. Har bir talaba 2-misolimizdagi kabi o'zlarining familiya, ismi va otasining ismidan foydalanib ixtiyoriy birorta belgilar sonini topishlari kerak(Ibragimov Ahadjon Ahamadullo o'g'li deb kiritilsa, va belgi "o" kiritilsa 4 ta "o" belgisi bor deb natija chiqishi lozim).

3-LABORATORIYA ISHI.

MAVZU: Dinamik ma'lumotlar tuzilmasi.

Ishdan maqsad: Dinamik ma'lumotlar tuzilmalari bilan tanishish. Bir o'lchovli va ikki o'lchovli massivlar haqida tushunchalar berish va ular ustida turli amallar bajarish.

Qo'yilgan masala: C++ dasturlash tilida massivning initsializatsiya qilingan va initsializatsiya qilinmagan holda e'lon qilishni o'rganish, bir o'lchovli va ikki o'lchovli massiv elementlariga murojaat qilish, massiv elementlarini o'sish va kamayish tartibida saralashlarni o'rganish va massivga doir turli misollarni ishlab chiqish.

Ish tartibi:

- Tajriba ishi nazariy ma'lumotlarini o'rganish;
- Berilgan topshiriqning algoritmini ishlab chiqish;
- C++ dasturlash muhitida dasturni yaratish;
- Natijalarni tekshirish;
- Hisobotni tayyorlash va topshirish.

Bir o'lchovli massivlar.

Massiv bu bir tipli nomerlangan ma'lumotlar jamlanmasidir. Massiv indeksli o'zgaruvchi tushunchasiga mos keladi. Massiv ta'riflanganda tipi, nomi va indekslar chegarasi ko'rsatiladi. Masalan, `long int a[5]; char w[200]; double f[4][5][7]; char[7][200]`. Massiv indekslar har doim 0 dan boshlanadi.

C++ dasturlash tili standarti bo'yicha indekslar soni 31 tagacha bo'lishi mumkin, lekin amalda bir o'lchovli va ikki o'lchovli massivlar qo'llaniladi. Bir o'lchovli massivlarga matematikada vektor tushunchasi mos keladi. Massivning `int z[3]` shakldagi ta'rif, `int` tipiga tegishli `z[0], z[1], z[2]` elementlardan iborat massivni aniqlaydi.

Massivlar ta'riflanganda initsializatsiya qilinishi, ya'ni boshlang'ich qiymatlarlari ko'rsatilishi mumkin. Masalan, `float C[]={1,-1,2,10,-12.5};` Bu misolda massiv chegarasi avtomatik aniqlanadi. Agar massiv initsializatsiya qilinganda elementlar chegarasi ko'rsatilgan bo'lsa, ro'yhatdagi elementlar soni bu chegaradan kam bo'lishi mumkin, lekin ortiq bo'lishi mumkin emas. Masalan, `int A[5]={2,-2}`. Bu holda `a[0]` va `a[1]` qiymatlari aniqlangan bo'lib, mos holda 2 va -2 ga teng.

Bir o'lchovli massivga quyidagi misolni ko'rib chiqaylik.

Berilgan `x[]={-1,2,5,-4,8,9}` ushbu massiv ichidan musbat elementlar soni, yig'indisini va ko'paytmasi topilsin.

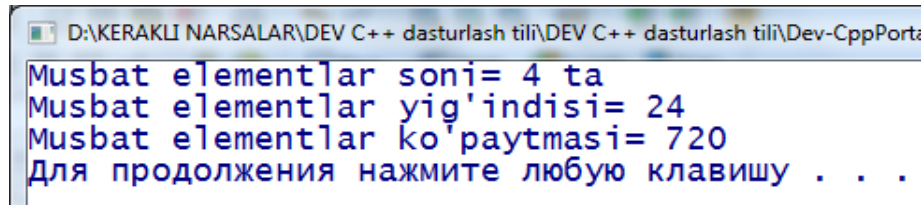
```
int x[]={-1,2,5,-4,8,9};
int s=0, p=1, k=0;
for ( int I=0; I<6; I++)
{ if (x[I]>0)
{ k++; } }
cout<<"Musbat elementlar soni=" <<k<<" ta"<<endl;
int t=0;
```

```

for ( int I=0; I<6; I++)
{ if (x[I]>0)
{ s+=x[I];
  p*=x[I]; }}
cout<<"Musbat elementlar yig'indisi= "<<s<<endl;
cout<<"Musbat elementlar ko'paytmasidisi= "<<p<<endl;

```

NATIJA:



1-rasm.

Ikki o'lchovli massivlar.

Ikki o'lchovli massivlar matematikada matritsa yoki jadval tushunchasiga mos keladi. Jadvallarning initsializatsiya qilish qoidasi, ikki o'lchovli massivning elementlari massivlardan iborat bo'lgan bir o'lchovli massiv ta'rifiga asoslangandir. Masalan, ikki qator va uch ustundan iborat bo'lgan haqiqiy tipga tegishli d massiv boshlang'ich qiymatlari quyidagicha ko'rsatilishi mumkin:

```
float d[2][3]={(1,-2.5,10),(-5.3,2,14)};
```

Bu yozuv quyidagi qiymat berish operatorlariga mosdir:

```
d[0][0]=1;d[0][1]=-2.5;d[0][2]=10;d[1][0]=-5.3;d[1][1]=2;d[1][2]=14;
```

Bu qiymatlarni bitta ro'yxat bilan hosil qilish mumkin:

```
float d[2][3]={1,-2.5,10,-5.3,2,14};
```

Initsializatsiya yordamida boshlang'ich qiymatlar aniqlanganda massivning hamma elementlariga qiymat berish shart emas. Masalan,

```
int x[3][3]={(1,-2,3),(1,2),(-4)}
```

Bu yozuv quyidagi qiymat berish operatorlariga mosdir:

```
x[0][0]=1;x[0][1]=-2;x[0][2]=3;x[1][0]=-1;x[1][1]=2;x[2][0]=-4;
```

Initsializatsiya yordamida boshlang'ich qiymatlar aniqlanganda massivning birinchi indeksi chegarasi ko'rsatilishi shart emas, lekin qolgan indekslar chegaralari ko'rsatilishi shart. Masalan,

```
Double x[][2]={(1.1,1.5),(-1.6,2.5),(3,-4)}
```

Ikki o'lchovli massivga quyidagi misolni ko'rib chiqaylik.

Berilgan int a[2][3]={2,-5,9,4,7,-2} ushbu ikki o'lchovli massiv ekranda matritsa ko'rinishida hosil qilsin va elementlar yig'indisi aniqlansin

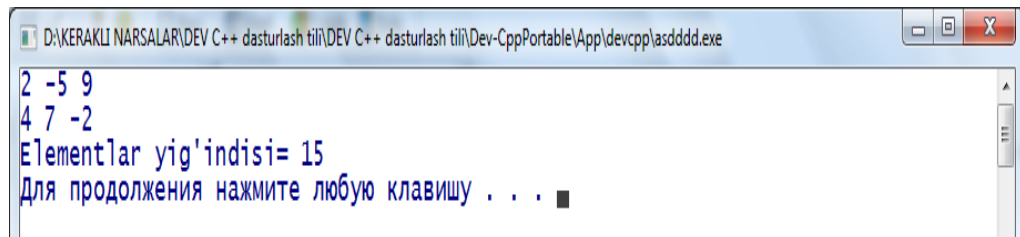
```

int a[2][3]={2,-5,9,4,7,-2};
int s=0;
for(int i=0; i<2; i++)
{for(int j=0; j<3; j++)
{cout<<a[i][j]<<" ";}
cout<<endl;}

```

```
for(int i=0; i<2; i++)
for(int j=0; j<3; j++)
s+=a[i][j];
cout<<"Elementlar yig'indisi= "<<s<<endl;
```

NATIJA:



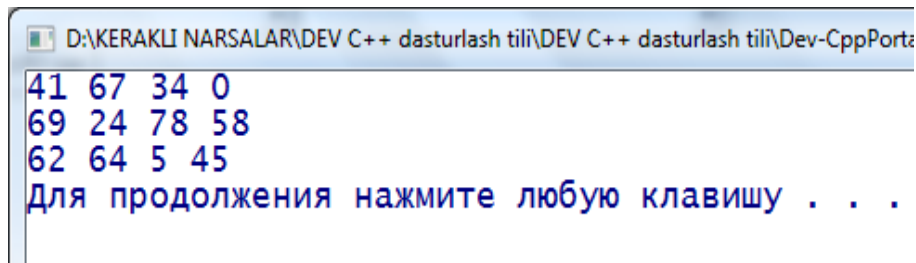
```
D:\KERAKLI NARSALAR\DEV C++ dasturlash tili\DEV C++ dasturlash tili\Dev-CppPortable\App\devcpp\asddddd.exe
2 -5 9
4 7 -2
Elementlar yig'indisi= 15
Для продолжения нажмите любую клавишу . . .
```

2-rasm.

Ushbu dasturimizda ikki o'lcho'vli massivlar uchun rand() funksiyasidan foydalanish misol qilib keltirilgan. rand() funksiyasi 0 dan 100 gacha sonlar ichidan ixtiyoriy matritsa ko'rinishida hosil qiladi. Ushbu misolimizda 0 dan 100 gacha sonlardan 3 ga 4 matritsa hosil qilingan.

```
int a[3][4];
for(int i=0; i<3; i++)
{for(int j=0; j<4; j++)
{a[i][j]=rand()%100;
cout<<a[i][j]<<" "; }
cout<<endl;}
```

NATIJA:



```
D:\KERAKLI NARSALAR\DEV C++ dasturlash tili\DEV C++ dasturlash tili\Dev-CppPorta
41 67 34 0
69 24 78 58
62 64 5 45
Для продолжения нажмите любую клавишу . . .
```

3-rasm.

Nazorat savollari:

1. Massiv deganda nimani tushunasiz ?
2. int S[25]; C++ dasturlash tilida nimani bildiradi ?
3. int x={1,2,3,4} ; necha o'lchovli massiv ?
4. Ikki o'lchovli massi deganda nimani tushunasiz va qanday e'lon qilamiz ?
5. rand() funksiyasining vazifasi nimadan iborat?

Topshiriqlar:

1. Berilgan $x[] = \{-21, 12, -5, 7, -9, 3\}$ ushbu massiv ichidan manfiy elementlar soni, yig'indisini va ko'paytmasi topilsin va manfiy element yo'q bo'lsa, "manfiy element yo'q" yozuvi ekranga chiqsin.
2. Berilgan n ta elementi bor massiv ichidan 8 karrali sonlarining nechtaligini, yig'indisini va ko'paytmasini aniqlovchi dastur tuzing.
3. Berilgan $\text{int } a[2][3] = \{6, -5, 10, 14, 7, -12\}$ ushbu ikki o'lchovli massiv ekranda matritsa ko'rinishida hosil qiluvchi, elementlar yig'indisi va ko'paytmasini aniqlovchi dastur tuzing.
4. Berilgan $x[] = \{-21, 12, -5, 17, -9, 23\}$ ushbu massiv ichidan musbat elementlar soni, yig'indisini va ko'paytmasi topilsin va musbat element yo'q bo'lsa, "musbat element yo'q" yozuvi ekranga chiqsin.
5. Berilgan $x[] = \{-21, -12, -5, -17, -9, 7, -23\}$ ushbu massiv ichidan manfiy elementlarni kvadratga oshiruvchi dastur tuzing.
6. Berilgan $x[] = \{11, 2, -5, 8, -9, 14, 7\}$ ushbu massiv elementlarining eng kichigini, eng kattasini va o'rta arifmetigini toping.
7. Berilgan $x[] = \{11, 3, -51, 28, 9, -4, 77\}$ ushbu massiv elementlarining kamayish tartibda saralovchi dastur tuzing.
8. Berilgan n ta elementi bor massiv elementlari ichidan musbat elementlarini kubni va manfiy elementlarini kvadratni topadigan dastur tuzing.
9. Berilgan $x[] = \{-21, -12, -5, -17, -9, 0, -23\}$ ushbu massiv ichidan manfiy elementlar soni, yig'indisini va ko'paytmasi topilsin va musbat element yo'q bo'lsa, "musbat element yo'q" yozuvi ekranga chiqsin.
10. Berilgan $x[] = \{-21, 12, -5, 8, -9, 22\}$ ushbu massiv ichidan juft elementlarni kvadratga oshiruvchi dastur tuzing.
11. Berilgan $\text{int } a[2][3] = \{3, -15, 9, 1, 17, -32\}$ ushbu ikki o'lchovli massiv ekranda matritsa ko'rinishida hosil qiluvchi, elementlar yig'indisi va ko'paytmasini aniqlovchi dastur tuzing.
12. Berilgan $x[] = \{-21, 12, -5, 8, -9, 3\}$ ushbu massiv ichidan toq elementlar soni, yig'indisini va ko'paytmasi topilsin va toq element yo'q bo'lsa, "toq element yo'q" yozuvi ekranga chiqsin.
13. Berilgan $x[] = \{11, 3, -51, 28, 9, -4, 77\}$ ushbu massiv elementlarining o'shish tartibda saralovchi dastur tuzing.
14. Ikki o'lcho'vli massivlar uchun $\text{rand}()$ funksiyasidan foydalanib 0 dan 100 gacha bo'lgan sonlar ichidan 4 ga 5 matritsa yaratuvchi dastur tuzing.
15. Berilgan n ta elementi bor massiv ichidan 7 karrali sonlarining nechtaligini va kvadratini aniqlovchi dastur tuzing.
16. Berilgan $\text{int } a[2][3] = \{9, 12, 0, -24, 2, -1\}$ ushbu ikki o'lchovli massiv ekranda matritsa ko'rinishida hosil qiluvchi, elementlar yig'indisi va ko'paytmasini aniqlovchi dastur tuzing.
17. Berilgan $x[] = \{-21, 12, -5, 8, -9, 22\}$ ushbu massiv ichidan juft elementlar soni, yig'indisini va ko'paytmasi topilsin va juft element yo'q bo'lsa, "juft element yo'q" yozuvi ekranga chiqsin.
18. Berilgan n ta elementi bor massiv elementlarini o'sish tartibda saralovchi

dastur tuzing.

19. Berilgan n ta elementi bor massiv elementlari ichidan juft elementlarini kubni va toq elementlarini kvadratni topadigan dastur tuzing.

20. Berilgan n ta elementi bor massiv ichidan 5 karrali sonlarining nechtaligini, yig'indisini va ko'paytmasini aniqlovchi dastur tuzing.

21. Berilgan n ta elementi bor massiv elementlarini o'rta arifmetigini topadigan dastur tuzing.

22. Ikki o'lcho'vli massivlar uchun `rand()` funksiyasidan foydalanib 0 dan 100 gacha bo'lgan sonlar ichidan 2 ga 3 matritsa yaratuvchi dastur tuzing.

23. Berilgan $x[] = \{-21, 12, -5, 17, -9, 23\}$ ushbu massiv ichidan musbat elementlarni kvadratga oshiruvchi dastur tuzing.

24. Berilgan n ta elementi bor massiv ichidan 9 karrali sonlarining nechtaligini va kvadratlarni aniqlovchi dastur tuzing.

25. Berilgan $x[] = \{-21, 12, -5, 8, -9, 3\}$ ushbu massiv ichidan toq elementlarni kvadratga oshiruvchi dastur tuzing.

26. Berilgan n ta elementi bor massiv elementlarini kamayish tartibda saralovchi dastur tuzing.

4-LABORATORIYA ISHI.

MAVZU: Rekursiv ma'lumotlar tuzilmasi.

Ishdan maqsad: Rekursiv ma'lumotlar tuzilmasi bilan tanishish. Vektorlar haqida tushunchalar berish va ular ustida turli amallar bajarish.

Qo'yilgan masala: C++ dasturlash tilida vektorlarni e'lon qilishni o'rganish, "vector" va "algorithm" kutubxonasining funksiyalari amalda qo'llash, vector oxiriga element qo'shish va o'chirish, vector elementlarini o'sish va kamayish tartibida saralashlarni o'rganish va vektorga doir turli misollarni ishlab chiqish.

Ish tartibi:

- Tajriba ishi nazariy ma'lumotlarini o'rganish;
- Berilgan topshiriqning algoritmini ishlab chiqish;
- C++ dasturlash muhitida dasturni yaratish;
- Natijalarni tekshirish;
- Hisobotni tayyorlash va topshirish.

Vektorlar.

C++ dasturlash muhitida ma'lumotlarni massivdan tashqari sal boshqacharoq usulda saqlashning yana bir turi mavjud, ya'ni vektorlar. Vektor elementlari ustida massiv elementlari ustida bajariladigan amallarni bajarish mumkin. Ma'lumotlarni massivda saqlashda elementlar soni oldindan ma'lum bo'lishi kerak. Ayrim paytlarda massivga nechta element kiritilishi ma'lum bo'lmaydi va o'shanda dinamik dasturlashdan foydalanish kerak bo'ladi, ya'ni massivga qo'shiladigan elementga xotira ajratishga to'g'ri keladi. Shunday hollarda vector klassidan foydalanish mumkin. Vector klassi o'zgaruvchan uzunlikdagi massiv yaratishga yordam beradi. Vektor bu elementlari soni oldindan ma'lum bo'lmagan bir xil toifadagi elementlar ketma-ketligidir. Vektorning massivdan farqi, vector uzunligi oldindan berilmaydi va u dastur bajarilishi mobaynida o'zgarib turadi.

Vektor yaratish uchun <vector> kutubxonasiga ulanish kerak, ya'ni dastur boshida #include<vector> qatori bo'lishi kerak va vektorni e'lon qilishning 2 ta usuli mavjud - vektor uzunligini ko'rsatib va bo'sh vektor ko'rinishida.

```
vector<toifa_nomi> o'zgaruvchinomi;
```

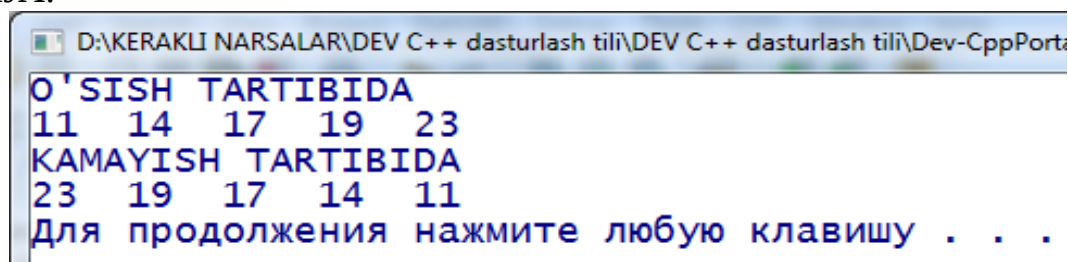
Masalan, vector <int> test; bu yerda int toifali, test nomli bo'sh vektor yaratildi. Vektor elementlariga indeks orqali murojaat qilib bo'ladi, lekin bu ko'rinishda vektor yaratilganda vektor elementiga indeks bilan murojaat qilib qiymat berib bo'lmaydi, ya'ni quyidagi dastur kodi noto'g'ri:

```
vector<int> vek;  
vek[0]=123;  
vek[1]=234;
```

1-misol: Butun turda 5 ta elementi bor w nomli vektor yarating va vektor elementlarini o'shish va kamayish tartibida saralang.

Buning uchun bizga vector va algorithm kutubxonalri kerak bo'ladi.

```
vector<int>w(5);
w[0]=19;
w[1]=11;
w[2]=23;
w[3]=14;
w[4]=17;
cout<<"O'SISH TARTIBIDA"<<endl;
sort(w.begin(),w.end());
for(int i=0;i<w.size();i++) cout<<w[i]<<" ";
cout<<endl;
cout<<"KAMAYISH TARTIBIDA"<<endl;
sort(w.begin(),w.end());
for(int i=(w.size()-1); i>=0; i--) cout<<w[i]<<" ";
cout<<endl;
NATIJA:
```

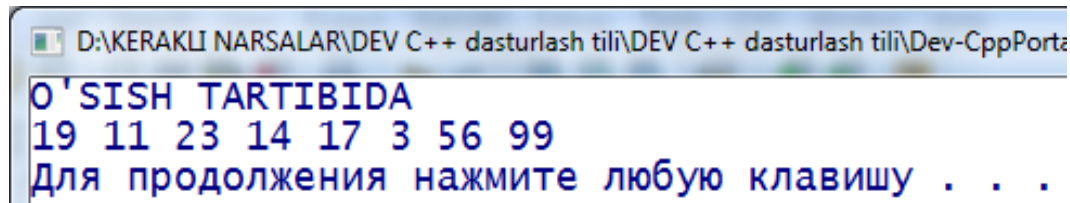


1-rasm.

2-misol: Butun turda 5 ta elementi bor w nomli vektor yarating, vektor oxiriga 3 ta element qo'shing va vektor elementlaridan faqat oxiringi 3 ta elementini o'shish tartibida saralang.

```
vector<int>w(5);
w[0]=19;
w[1]=11;
w[2]=23;
w[3]=14;
w[4]=17;
w.push_back(99);
w.push_back(3);
w.push_back(56);
cout<<"O'SISH TARTIBIDA"<<endl;
sort(w.begin()+5,w.end());
for(int i=0;i<w.size();i++) cout<<w[i]<<" ";
cout<<endl;
```

NATIJA:



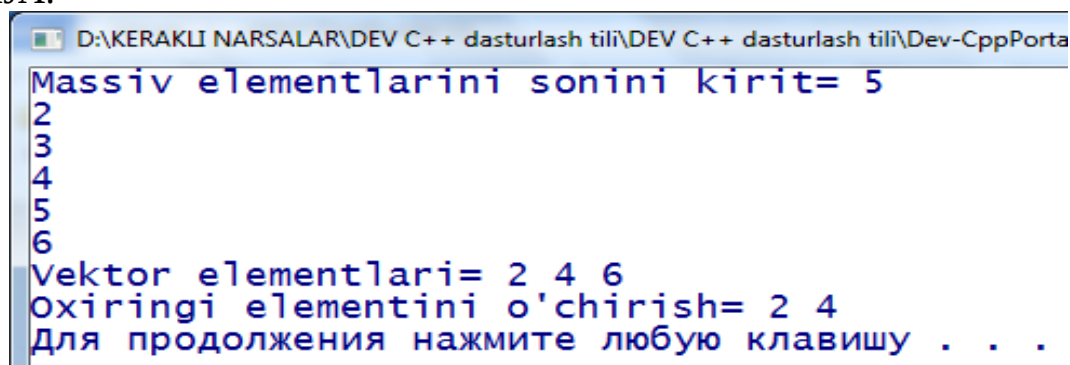
```
O'SISH TARTIBIDA
19 11 23 14 17 3 56 99
Для продолжения нажмите любую клавишу . . .
```

2-rasm.

3-misol: Butun turda ixtiyoriy elementi bor massiv ichidan juft elementlarini vektorga o'zlashtiring, ularni ekranga chiqaring va vektorning oxirigi elementini o'chirib ekranga chiqaring.

```
vector<int> v;
int n; cout<<"Massiv elementlarini sonini kirit= ";
cin>>n; int a[n];
for (int i=0;i<n;i++)
{
    cin>>a[i];
    if(a[i]%2==0)
        v.push_back(a[i]);
}
cout<<"Vektor elementlari= ";
for (int i=0;i<v.size();i++)
    cout<<v[i]<<" "; cout<<endl;
v.pop_back();
cout<<"Oxiringi elementini o'chirish= ";
for (int i=0;i<v.size();i++)
    cout<<v[i]<<" "; cout<<endl;
```

NATIJA:



```
Massiv elementlarini sonini kirit= 5
2
3
4
5
6
Vektor elementlari= 2 4 6
Oxiringi elementini o'chirish= 2 4
Для продолжения нажмите любую клавишу . . .
```

3-rasm.

Nazorat savollari:

1. Vektor bilan massiv qanday farqlanadi?
2. Vektor ustida qanday amallar bajarish mumkin?
3. Vektor elementlari ustida massiv elementlari ustida bajariladigan amallarni bajarish mumkinmi?
4. Vektorning oxiriga element qo'shish yoki o'chirish mumkinmi?

Topshiriqlar:

1. Butun turda ixtiyoriy elementi bor massiv ichidan toq elementlarini vektorga o'zlashtiring, ularni ekranga chiqaring va vektorning oxirigi elementini o'chirib ekranga chiqaring.
2. Butun turda 5 ta elementi bor "vek" nomli vektor hosil qiling, vektor oxiriga 4 ta element qo'shing, vektorning barcha elementlarini kamayish tartibida saralang va oxirigi elementi o'chirib ekranga chiqaring.
3. Butun turda 8 ta elementi bor "vek" nomli vektor hosil qiling, vektor oxiriga 1 ta element qo'shing, vektorning barcha elementlarini kamayish tartibida saralang va oxirigi elementi o'chirib ekranga chiqaring..
4. Butun turda ixtiyoriy elementi bor massiv ichidan musbat elementlarini vektorga o'zlashtiring, ularni ekranga chiqaring va vektorning oxirigi elementini o'chirib ekranga chiqaring.
5. Butun turda ixtiyoriy elementi bor massiv ichidan manfiy elementlarini vektorga o'zlashtiring, ularni ekranga chiqaring va vektorning oxirigi elementini o'chirib ekranga chiqaring.
6. Butun turda ixtiyoriy elementi bor massiv ichidan 5 karrali elementlarini vektorga o'zlashtiring, ularni ekranga chiqaring va vektorning oxirigi elementini o'chirib ekranga chiqaring.
7. Butun turda ixtiyoriy elementi bor massiv ichidan 9 karrali elementlarini vektorga o'zlashtiring, ularni ekranga chiqaring va vektorning oxirigi elementini o'chirib ekranga chiqaring.
8. Ushbu vector<int>w(5) vektorni hosil qiling va elementlarini kiriting, vektor oxiriga 2 ta element qo'shing va vektor elementlaridan faqat oxirigi 2 ta elementini o'shish tartibida saralang.
9. Ushbu vector<int>w(7) vektorni hosil qiling va elementlarini kiriting, vektor oxiriga 3 ta element qo'shing va vektor elementlaridan faqat oxirigi 5 ta elementini o'shish tartibida saralang.
10. Ushbu vector<int>w(8) vektorni hosil qiling va elementlarini kiriting, vektor oxiriga 2 ta element qo'shing va vektor elementlaridan faqat oxirigi 2 ta elementini kamayish tartibida saralang.
11. Ushbu vector<int>w(7) vektorni hosil qiling va elementlarini kiriting, vektor oxiriga 3 ta element qo'shing va vektor elementlaridan faqat oxirigi 5 ta elementini kamayish tartibida saralang.
12. Butun turda 5 ta elementi bor "vek" nomli vektor hosil qiling, vektor oxiriga 4 ta element qo'shing va vektorning barcha elementlarini o'shish va kamayish tartibida saralang.
13. Butun turda 8 ta elementi bor "vek" nomli vektor hosil qiling, vektor oxiriga 1 ta element qo'shing va vektorning barcha elementlarini o'shish va kamayish tartibida saralang.
14. Ushbu vector<int>w(7) vektorni hosil qiling, birinchi elementiga 5 ni qo'shing va oxirigi elementini o'chirib ekranga chiqaring.
15. Ushbu vector<int>w(4) vektorni hosil qiling, birinchi va ikkinchi elementlariga 8 ni qo'shing va oxirigi elementini o'chirib ekranga chiqaring.

16. Ushbu 17, 9, 78, 4,89 sonlardan iborat vektorni hosil qiling, birinchi va ikkinchi elementlariga 8 ni qo'shing va oxirigi elementini o'chirib ekranga chiqaring.

17. Ushbu 12, -5, 6, -45, 23 sonlardan iborat massiv ichidan manfiy elementlarini vektorga o'zlashtiring, ularni ekranga chiqaring va vektorning oxirigi elementini o'chirib ekranga chiqaring.

18. Ushbu 12, 35, 16, 45, 21 sonlardan iborat massiv ichidan 7 karrali elementlarini vektorga o'zlashtiring, ularni ekranga chiqaring va vektorning oxirigi elementini o'chirib ekranga chiqaring.

19. Ushbu 18, 35, 63, 45, 27 sonlardan iborat massiv ichidan 9 karrali elementlarini vektorga o'zlashtiring, ularni ekranga chiqaring va vektorning oxirigi elementini o'chirib ekranga chiqaring.

20. Ushbu 18, 35, 63, 45, 27 sonlardan iborat vektorni hosil qiling va vektor oxiriga 2 ta element qo'shing va vektor elementlaridan faqat oxirigi 3 ta elementini o'shish tartibida saralang.

21. Ushbu 12, 35, 66, 45, 31 sonlardan iborat vektorni hosil qiling va vektor oxiriga 3 ta element qo'shing va vektor elementlaridan faqat oxirigi 2 ta elementini o'shish tartibida saralang.

22. Ushbu 32, 5, 16, 45, 24 sonlardan iborat vektorni hosil qiling va vektor oxiriga 2 ta element qo'shing va vektor elementlaridan faqat oxirigi 4 ta elementini kamayish tartibida saralang.

23. Ushbu 13, 35, 36, 45, 99 sonlardan iborat vektorni hosil qiling va vektor oxiriga 3 ta element qo'shing va vektor elementlaridan faqat oxirigi 5 ta elementini kamayish tartibida saralang.

24. Ushbu 56, 35, 66, 45, 67 sonlardan iborat vektorni hosil qiling va vektor oxiriga 4 ta element qo'shing va vektorning barcha elementlarini o'shish va kamayish tartibida saralang.

25. Ushbu 44, 55, 66, 44, 77 sonlardan iborat vektorni hosil qiling va vektor oxiriga 1 ta element qo'shing va vektorning barcha elementlarini o'shish va kamayish tartibida saralang.

5-LABORATORIYA ISHI.

MAVZU: Qidiruv usullarini tadqiq qilish.

Ishdan maqsad: Qidiruv usullarini bilan tanishish. Yozuv va jadval ko'rinishidagi ma'lumotlar tuzilmalari haqida tushunchalar berish va ular ustida turli amallar bajarish.

Qo'yilgan masala: C++ dasturlash tilida yozuv va jadvallarni e'lon qilishni o'rganish, natijalarni ekranga chiqarish uchun funksiyalar hosil qilish, kalit so'z bo'yicha ma'lumotlarni qidirish va ixtiyoriy turlar bilan ishlashni o'rganish va yozuv va jadvallarga doir turli misollarni ishlab chiqish.

Ish tartibi:

- Tajriba ishi nazariy ma'lumotlarini o'rganish;
- Berilgan topshiriqning algoritmini ishlab chiqish;
- C++ dasturlash muhitida dasturni yaratish;
- Natijalarni tekshirish;
- Hisobotni tayyorlash va topshirish.

Yozuv(struktura).

Yozuv – bu turli tipdagi ma'lumotlarning biror nom ostida birlashtirilgan, dasturchi tomonidan beriladigan yangi tipdir.

Yozuvning massivdan farqi shundan iborat-ki, uning elementlari turli turlarga tegishli bo'la di, massiv elementlari esa bir xil turga tegishli bo'ladi.

Yozuvni e'lon qilish: Yozuvni e'lon qilish uchun struct kalit so'zi ishlatiladi. Undan keyin turlarga nom beriladi va {} qavs ichida maydonlar turlari va nomlari e'lon qilinadi. Yozuvda ma'lumot elementlarini yozuv maydonlari deb ham ataladi

Strukturali tip kiritilishi bu tip uchun xotiradan joy ajratilishiga olib kelmaydi. Har bir konkret struktura (ob'ekt) ta'riflanganda, shu ob'ekt uchun elementlar tiplariga qarab xotiradan joy ajratiladi. xotiradan joy zich ajratilganda struktura uchun ajratilgan joy hajmi har bir element uchun zarur bo'lgan xotira hajmlari yig'indisiga teng bo'ladi. Shu bilan birga xotiradan joy zich ajratilmasligi ham mumkin ya'ni elementlar orasida bo'sh joylar ham qolishi mumkin. Bu bo'sh joy keyingi elementni xotira qismlarining qabul qilingan chegaralari bo'yicha tekislash uchun qoldiriladi. Misol uchun butun tipdagi elementlar juft adreslardan boshlansa, bu elementlarga murojaat tezroq amalga oshiriladi.

Konkret strukturalarni joylashuviga ba'zi kompilyatorlarda #PRAGMA PACK(N) preprotssessor direktivasi yordamida ta'sir o'tkazish mumkin. Bu direktivadan quyidagi shaklda foydalanish mumkin:

Bu yerda n qiymati 1,2 eki 4 ga teng bo'lishi mumkin.

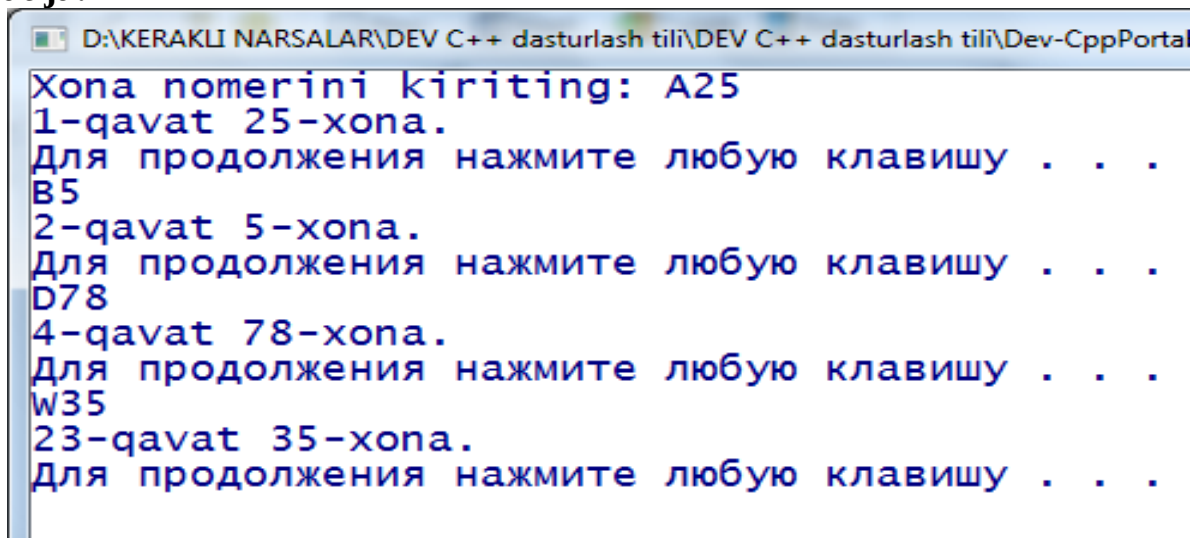
- Pack(1) – elementlarni bayt chegara- lari bo'yicha tekislash;
- Pack(2) – elementlarni so'zlar chegaralariga qarab tekislash;
- Pack(4) – elementlarni ikkilangan yuzlar chegaralariga qarab tekislash.

1-misol. Binoning qavatlari va xonalari bilan bog'liq Bino nomli yozuv va qavat, xona nomli maydon yaratining va A23 nomli xona kiritilsa 1-qavat 23-xona so'zi ekranga chiqsin.

Dastur qismi:

```
struct Bino
{
    char qavat;
    int xona;
};
int main(int argc, char *argv[])
{
    Bino b;
    cout<<"Xona nomerini kiriting: ";
    cin>>b.qavat>>b.xona;
    cout<<int(b.qavat)-64<<"-qavat "<<b.xona<<"-xona.";
```

Natija:



1-rasm.

2-misol. Talaba nomli yozuv va familiya, ism, tug'ilgan yil va baho nomli maydon yaratining va faqat 4 bahoga o'zlashtirgan talabalarni ekranga chiqaruvchi va 4 bahoga o'zlashtirgan talaba yo'q bo'lsa "topilmadi" so'zi ekranga chiqsin.

Dastur qismi:

```
struct talaba
{
    string fam,ism;
    int tyil;
    int baho;
};
void print_t(talaba a)
{ cout<<a.fam<<" "<<a.ism<<" "<<a.tyil<<" "<<a.baho<<endl;}
int main(int argc, char *argv[])
{
```

```

int n;
talaba b[10];
cout<<"Talaba soni kirit= ";
cin>>n;
for(int i=1; i<=n; i++)
{
cout<<i<<"-fam kit= "; cin>>b[i].fam;
cout<<i<<"-ism kit= "; cin>>b[i].ism;
cout<<i<<"-tyil kit= "; cin>>b[i].tyil;
cout<<i<<"-baho kit= "; cin>>b[i].baho; }
int k=0;
for(int i=1; i<=n; i++)
{
if(b[i].baho==4)
{print_t(b[i]);}
else k++;
if (k==n)
cout<<"topilmadi"<<endl;
}
Natija:

```

```

D:\KERAKLI NARSALAR\DEV C++ dasturlash tili\DEV C++ dasturlash tili\Dev-CppPorta
Talaba soni kirit= 3
1-fam kit= Aliyev
1-ism kit= Vali
1-tyil kit= 1990
1-baho kit= 5
2-fam kit= Ahmedova
2-ism kit= Lola
2-tyil kit= 1980
2-baho kit= 3
3-fam kit= Qodirov
3-ism kit= Botir
3-tyil kit= 1984
3-baho kit= 4
Aliyev Vali 1990 5
Qodirov Botir 1984 4
Для продолжения нажмите любую клавишу . . .

```

2-rasm.

Nazorat savollari:

5. Yozuv bilan jadvalning qanday farqlanadi?
6. Yozuv va jadval ustida qanday amallar bajarish mumkin?
7. Yozuv va jadval ko'rinishidagi ma'lumotlarni xotiraga joylashtirishni qanday usullarini bilasiz?
8. **PRAGMA PACK()** funksiyasini vazifasi nimadan iborat.

Topshiriqlar:

1-topshiriq.

Quyidagi maydonlardan STUDENT nomi bilan yozuv hosil qiling:

- Familiya va ismi;
- Gurux raqami;
- Bahosi;
- Manzili;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 10 ta STUDENT toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- Bahosi 3 va 4 bo'lgan talabalar ro'yxati ekranga chiqarilsin.
- Agar bunday bahoga ega talabalar bo'lmasa, u xolda ekranga "bahosi 3 va 4 bo'lgan talabalar topilmadi" xabari matni chiqarilsin.

2-topshiriq.

Quyidagi maydonlardan WORKER nomi bilan yozuv hosil qiling:

- Familiya va ismi;
- Ish joyi;
- Lavozimi;
- Ishga kirgan vaqti;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 7 ta WORKER toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- TATUFF ishchilari ro'yxati ekranga chiqarilsin.
- Agar bunday ishchilar bo'lmasa, u xolda ekranga "TATUFF ishchilari topilmadi" xabari matni chiqarilsin.

3-topshiriq.

Quyidagi maydonlardan PRICE nomi bilan yozuv hosil qiling:

- Mahsulot nomi;
- Narxi;
- Do'kon nomi;
- Ishlab chiqarilgan yili;
- Yaroqlilik muddati;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 11 ta PRICE toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- Narxi 40000 so'm bo'lgan mahsulotlar ro'yxati ekranga chiqarilsin.
- Agar bunday mahsulot yo'q bo'lmasa, u xolda ekranga "Mahsulot topilmadi" xabari matni chiqarilsin.

4-topshiriq.

Quyidagi maydonlardan MARKETING nomi bilan yozuv hosil qiling:

- Familiya va ismi;
- Guruxi;
- Kursi;

- Qarzdorlik summasi;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 8 ta MARKETING toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- Qarzdorligi 150000 so'mdan yuqori talabalar ro'yxati ekranga chiqarilsin.
- Agar bunday talaba yo'q bo'lmasa, u xolda ekranga "Qarzdorlar yo'q" xabari matni chiqarilsin.

5-topshiriq.

Quyidagi maydonlardan PRICE nomi bilan yozuv hosil qiling:

- Mahsulot nomi;
- Narxi;
- Do'kon nomi;
- Ishlab chiqarilgan yili;
- Yaroqlilik muddati;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 7 ta PRICE toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- Yaroqlilik muddat 2 yildan past bo'lgan mahsulotlar ro'yxati ekranga chiqarilsin.
- Agar bunday mahsulot yo'q bo'lmasa, u xolda ekranga "Mahsulot topilmadi" xabari matni chiqarilsin.

6-topshiriq.

Quyidagi maydonlardan WORKER nomi bilan yozuv hosil qiling:

- Familiya va ismi;
- Ish joyi;
- Lavozimi;
- Ishchini manzili;
- Ishga kirgan yili;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 11 ta WORKER toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- Andijonlik bo'lgan ishchilar ro'yxati ekranga chiqarilsin.
- Agar bunday ishchilar yo'q bo'lmasa, u xolda ekranga "Andijonlik ishchilar topilmadi" xabari matni chiqarilsin.

7-topshiriq.

Quyidagi maydonlardan STUDENT nomi bilan yozuv hosil qiling:

- Familiya va ismi;
- Gurux raqami;
- Fakultet;
- Bahosi;
- Manzili;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 10 ta STUDENT toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;

- Namanganlik talabalar ro'yxati ekranga chiqarilsin.
- Agar bunday bahoga ega talabalar bo'lmasa, u xolda ekranga "Namanganlik talabalar topilmadi" xabari matni chiqarilsin.

8-topshiriq.

Quyidagi maydonlardan WORKER nomi bilan yozuv hosil qiling:

- Familiya va ismi;
- Ish joyi;
- Lavozimi;
- Tug'ilgan yili;
- Ishga kirgan vaqti;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 8 ta WORKER toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- 1990 yilda tug'ilgan ishchilar ro'yxati ekranga chiqarilsin.
- Agar bunday ishchilar yo'q bo'lmasa, u xolda ekranga "1990 yilda tug'ilgan ishchilar topilmadi" xabari matni chiqarilsin.

9-topshiriq.

Quyidagi maydonlardan PRICE nomi bilan yozuv hosil qiling:

- Mahsulot nomi;
- Narxi;
- Do'kon nomi;
- Ishlab chiqarilgan yili;
- Yaroqlilik muddati;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 10 ta PRICE toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- Narxi 40000 so'mdan yuqori bo'lgan mahsulotlar ro'yxati ekranga chiqarilsin.
- Agar bunday mahsulot yo'q bo'lmasa, u xolda ekranga "Mahsulot topilmadi" xabari matni chiqarilsin.

10-topshiriq.

Quyidagi maydonlardan MARKETING nomi bilan yozuv hosil qiling:

- Familiya va ismi;
- Guruxi;
- Kursi;
- Qarzdorlik summasi;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 7 ta MARKETING toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- Qarzdorligi 1000000 so'mdan past talabalar ro'yxati ekranga chiqarilsin.
- Agar bunday talaba yo'q bo'lmasa, u xolda ekranga "Topilmadi" xabari matni chiqarilsin.

11-topshiriq.

Quyidagi maydonlardan PRICE nomi bilan yozuv hosil qiling:

- Mahsulot nomi;

- Narxi;
- Do'kon nomi;
- Ishlab chiqarilgan yili;
- Yaroqlilik muddati;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 8 ta PRICE toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- Yaroqlilik muddat 2 yildan yuqori bo'lgan mahsulotlar ro'yxati ekranga chiqarilsin.
- Agar bunday mahsulot yo'q bo'lmasa, u xolda ekranga "Mahsulot topilmadi" xabari matni chiqarilsin.

12-topshiriq.

Quyidagi maydonlardan WORKER nomi bilan yozuv hosil qiling:

- Familiya va ismi;
- Ish joyi;
- Lavozimi;
- Ishchini manzili;
- Ishga kirgan yili;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 8 ta WORKER toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- Namanganlik bo'lgan ishchilar ro'yxati ekranga chiqarilsin.
- Agar bunday ishchilar yo'q bo'lmasa, u xolda ekranga "Namanganlik ishchilar topilmadi" xabari matni chiqarilsin.

13-topshiriq.

Quyidagi maydonlardan STUDENT nomi bilan yozuv hosil qiling:

- Familiya va ismi;
- Gurux raqami;
- Fakultet;
- Bahosi;
- Manzili;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 9 ta STUDENT toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- Andijonlik talabalar ro'yxati ekranga chiqarilsin.
- Agar bunday bahoga ega talabalar bo'lmasa, u xolda ekranga "Andijonlik talabalar topilmadi" xabari matni chiqarilsin.

14-topshiriq.

Quyidagi maydonlardan WORKER nomi bilan yozuv hosil qiling:

- Familiya va ismi;
- Ish joyi;
- Lavozimi;
- Tug'ilgan yili;
- Ishga kirgan vaqti;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 10 ta WORKER toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- Ishchilaridan assistentlarni ro'yxati ekranga chiqarilsin.
- Agar assistent yo'q bo'lmasa, u xolda ekranga "Assistent topilmadi" xabari matni chiqarilsin.

15-topshiriq.

Quyidagi maydonlardan PRICE nomi bilan yozuv hosil qiling:

- Mahsulot nomi;
- Narxi;
- Do'kon nomi;
- Ishlab chiqarilgan yili;
- Yaroqlilik muddati;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 9 ta PRICE toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- Narxi 40000 so'mdan arzon bo'lgan mahsulotlar ro'yxati ekranga chiqarilsin.
- Agar bunday mahsulot yo'q bo'lmasa, u xolda ekranga "Mahsulot topilmadi" xabari matni chiqarilsin.

16-topshiriq.

Quyidagi maydonlardan MARKETING nomi bilan yozuv hosil qiling:

- Familiya va ismi;
- Guruxi;
- Kursi;
- Qarzdorlik summasi;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 10 ta MARKETING toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- Qarzdorligi yo'q talabalar ro'yxati ekranga chiqarilsin.
- Agar bunday talaba yo'q bo'lmasa, u xolda ekranga "Topilmadi" xabari matni chiqarilsin.

17-topshiriq.

Quyidagi maydonlardan PRICE nomi bilan yozuv hosil qiling:

- Mahsulot nomi;
- Narxi;
- Do'kon nomi;
- Ishlab chiqarilgan sanasi;
- Yaroqlilik muddati;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 9 ta PRICE toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- 01.01.2015 kuni ishlab chiqarilgan mahsulotlar ro'yxati ekranga chiqarilsin.
- Agar bunday mahsulot yo'q bo'lmasa, u xolda ekranga "Mahsulot topilmadi" xabari matni chiqarilsin.

18-topshiriq.

Quyidagi maydonlardan WORKER nomi bilan yozuv hosil qiling:

- Familiya va ismi;
- Ish joyi;
- Lavozimi;
- Ishchini manzili;
- Ishga kirgan yili;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 6 ta WORKER toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- Farg'onalik bo'lgan ishchilar ro'yxati ekranga chiqarilsin.
- Agar bunday ishchilar yo'q bo'lmasa, u xolda ekranga "Farg'onalik ishchilar topilmadi" xabari matni chiqarilsin.

19-topshiriq.

Quyidagi maydonlardan STUDENT nomi bilan yozuv hosil qiling:

- Familiya va ismi;
- Gurux raqami;
- Fakultet;
- Bahosi;
- Manzili;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 6 ta STUDENT toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- Farg'onalik talabalar ro'yxati ekranga chiqarilsin.
- Agar bunday bahoga ega talabalar bo'lmasa, u xolda ekranga "Farg'onalik talabalar topilmadi" xabari matni chiqarilsin.

20-topshiriq.

Quyidagi maydonlardan WORKER nomi bilan yozuv hosil qiling:

- Familiya va ismi;
- Ish joyi;
- Lavozimi;
- Tug'ilgan yili;
- Ishga kirgan vaqti;

Navbatdagi amallarni bajaruvchi dasturni tuzing:

- 7 ta WORKER toifasidagi yozuv klaviatura orqali massiv ma'lumotlarini kiriting;
- 1980 yildan 1991 yilgacha tug'ilgan ishchilar ro'yxati ekranga chiqarilsin.
- Agar bunday ishchilar yo'q bo'lmasa, u xolda ekranga "Bunday ishchilar topilmadi" xabari matni chiqarilsin.

6-LABORATORIYA ISHI.

MAVZU: Saralash usullarini tadqiq qilish.

Ishdan maqsad: Ushbu laboratoriya ishining maqsadi talabalar qanday saralash usullari va algoritmlari mavjudligini va ularning samaradorliklarini baholashni o'rganishlari kerak. Shu asosda saralash usullarini qiyosiy tahlil qilishlari va ularga oid dasturlar tuzishni o'zlashtirishlari kerak.

Qo'yilgan masala: Talabalar topshiriq variantiga mos saralash usuli yordamida masalani yechish dasturini yaratish ko'nikmasiga ega bo'lishlari kerak.

Ish tartibi:

- Tajriba ishi nazariy ma'lumotlarini o'rganish;
- Berilgan topshiriqning algoritmini ishlab chiqish;
- C++ dasturlash muhitida dasturni yaratish;
- Natijalarni tekshirish;
- Hisobotni tayyorlash va topshirish.

Tuzilma elementlarini saralash.

Ma'lumotlarni kompyuterda qayta ishlashda elementning informatsion maydoni va uning mashina xotirasida joylashishini bilish zarur. Shu maqsadda ma'lumotlarni saralash amalga oshiriladi. Demak, saralash - bu ma'lumotlarni kalitlari bo'yicha doimiy ko'rinishda mashina xotirasida joylashtirishdan iborat. Bu yerda doimiylik ma'lumotlarni massivda kalitlari bo'yicha o'sishi tartibida berilishi tushuniladi.

Ma'lumotlarga qayta ishlov berilayotganda ma'lumotning informatsion maydonini hamda uning mashinada joylashishini (adresini) bilish zarur.

Saralashning ikkita turi mavjud: **ichki** va **tashqi**:

- ichki saralash bu operativ xotiradagi saralash;
- tashqi saralash - tashqi xotirada saralash.

Agar saralanayotgan yozuvlar xotirada katta hajmni egallasa, u holda ularni almashtirishlar katta sarf (vaqt va xotira ma'nosida) talab qiladi. Ushbu sarfni kamaytirish maqsadida, saralash kalitlar adresi jadvalida amalga oshiriladi. Bunda faqatgina ma'lumot ko'rsatkichlari almashtirilib, massiv o'z joyida qoladi. Bu usul adreslar jadvalini saralash usuli deyiladi.

Saralanayotganda bir xil kalitlar uchrashi mumkin, bu holda saralangandan keyin bir xil kalitlilar boshlang'ich tartibda qanday joylashgan bo'lsa, shu tartibda qoldirilishi maqsadga muvofiq bo'ladi (Bir xil kalitlilar o'zlariga nisbatan). Bunday usulga turg'un saralash deyiladi.

Saralash samaradorligini bir necha mezonlar bo'yicha baholash mumkin:

- saralashga ketgan vaqt;
- saralash uchun talab qilingan operativ xotira;
- dasturni ishlab chiqishga ketgan vaqt.

To'g'ridan-to'g'ri qo'shish usuli bilan saralash algoritmi.

Bunday usul karta o'yinida keng qo'llaniladi. Elementlar (kartalar) hayolan "tayyor" $a(1), \dots, a(i-1)$ va boshlang'ich ketma-ketliklarga bo'linadi. Har bir qadamda ($i=2$ dan boshlanib, har bir qadamda bir birlikka oshirib boriladi) boshlang'ich ketma-ketlikdan i -chi element ajratib olinib tayyor ketma-ketlikning kerakli joyiga qo'yiladi.

To'g'ridan-to'g'ri qo'shish orqali saralash algoritmi quyidagicha bo'ladi:

```
for (int i=1; i<n; i++)
{
    x=a[i];
    x ni a[0]...a[i] // oraliqning mos joyiga qo 'shish
}
```

Kerakli joyni qidirish jarayonini quyidagi tartibda olib borish qulay bo'ladi. 2-elementdan boshlab har bir elementni qarab chiqamiz, ya'ni har bir element o'zidan oldin turgan element bilan solishtiriladi. Agar qaralayotgan element kichik bo'lsa, oldinda turgan element bilan o'rin almashadi va yana o'zidan oldinda turgan element bilan solishtiriladi, jarayon shu kabi davom etadi. Bu jarayon quyidagi shartlarning birortasi bajarilganda to'xtatiladi:

1. x elementi oldida uning kalitidan kichik kalitli $a(j)$ elementi chiqqanda.
2. x elementi oldida element qolmaganda.

```
for (int i=1; i<n; i++)
{
    int j=i;
    while(a[j]<a[j-1])
    {
        int t=a[j-1];
        a[j-1]=a[j];
        a[j]=t;
        j=j-1;
    }
}
```

Tanlash orqali saralash algoritmi

Mazkur usul quyidagi tamoyillarga asoslangan:

- Eng kichik kalitga ega element tanlanadi.
- Ushbu element a_0 birinchi element bilan o'rin almashinadi.
- Keyin mazkur jarayon qolgan $n-1$, $n-2$ elementlar bilan takrorlanib, to bitta eng "katta" element qolguncha davom ettiriladi.

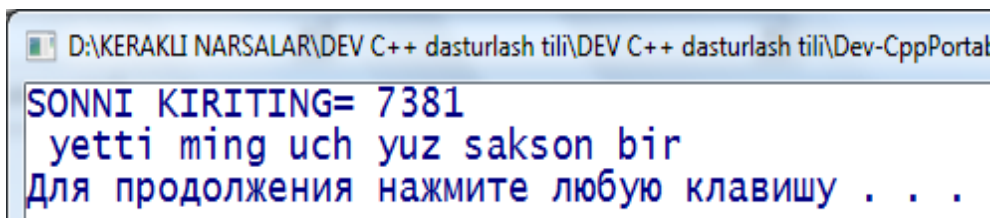
```
for(int i=0; i<n-1; i++)
for(int) j=i+1; j<n; j++)
if(a[i]>a[j])
{
    int k = a[j];
    a[j]=a[i];
    a[i]= k;}
}
```

1-misol. Kiritilgan sonni so'zlar bilan ifodalaydigan dasturni yaratish.

Dastur kodi:

```
string a[]={"bir","ikki","uch","to'rt","besh","olti","yetti","sakkiz","to'qqiz"};
string b[]={"o'n","yigirma","o'ttiz","qirq","ellik","oltmish","yetmish",
"sakson","to'qson"};
string nat="";
int n,g,k,p,x;
cout<<"SONNI KIRITING= ";      cin>>n;
x=n/1000;
if(x>=1) {nat=nat+a[x-1]+" ming ";}
if(n%1000==0) {nat=nat;}      cout<<" ";
n=n%1000;
p=n/100;
if(p>=1) {nat=nat+a[p-1]+" yuz ";}
if(n%100==0) {nat=nat;}
n=n%100;
k=n/10;
if(k>=1) {nat=nat+b[k-1]+" ";}
if(n%100==0) {nat=nat;}
g=n%10;
if(g>=1) {nat=nat+a[g-1]+" ";}
if(n%100==0) {nat=nat;}
cout<<nat<<endl;
```

Natija:

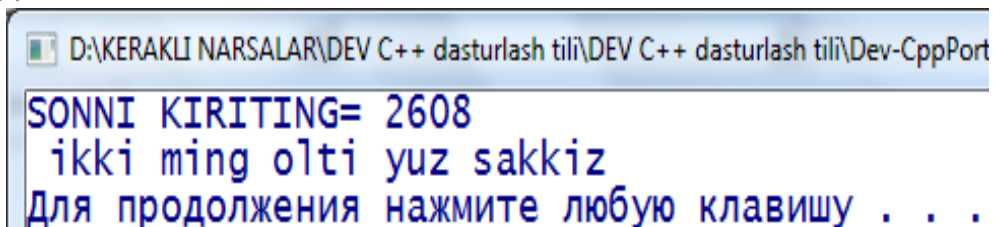


D:\KERAKLI NARSALAR\DEV C++ dasturlash tili\DEV C++ dasturlash tili\Dev-CppPortal
SONNI KIRITING= 7381
yetti ming uch yuz sakson bir
Для продолжения нажмите любую клавишу . . .

1-rasm.

Har bir talaba o'zining tug'ilgan kuni va oyini sonlar orqali kiritib, uni yozuvdagi ko'rinishini hosil qilishi zarur. (Masalan talaba 2608 da tug'ilgan bo'lsa, ekranga : ikki ming olti yuz sakkiz holatida chiqshi kerak).

Namuna:



D:\KERAKLI NARSALAR\DEV C++ dasturlash tili\DEV C++ dasturlash tili\Dev-CppPort
SONNI KIRITING= 2608
ikki ming olti yuz sakkiz
Для продолжения нажмите любую клавишу . . .

2-rasm.

7-LABORATORIYA ISHI.

MAVZU: Ma'lumotlarning integrallashgan strukturasi, yozuvlar.

Ishdan maqsad: Ushbu laboratoriya ishining maqsadi talabalar yozuvlar va jadvallar asosida tez saralash algoritmlarini qo'llashni, "Bo'laklarga bo'l, hukmronlik qil" tamoyili haqida tushunchaga ega bo'lish va ularga oid dasturlar tuzishni o'rganishlari kerak.

Qo'yilgan masala: Talabalar topshiriq variantiga mos saralash usullari va algoritmlari yordamida masalani yechish dasturini yaratish ko'nikmasiga ega bo'lish va o'rin almashtirish amalga oshirilganini aniqlang.

Ish tartibi:

- Tajriba ishi nazariy ma'lumotlarini o'rganish;
- Berilgan topshiriqning algoritmini ishlab chiqish;
- C++ dasturlash muhitida dasturni yaratish;
- Natijalarni tekshirish;
- Hisobotni tayyorlash va topshirish.

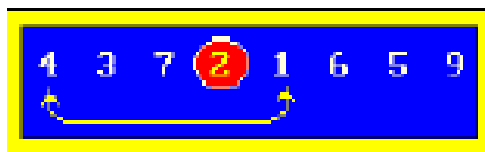
Quicksort – tez saralash algoritmi.

Bu algoritim "Bo'laklarga bo'l, hukmronlik qil" tamoyilining yaqqol misolidir. Bu algoritim rekursiv bo'lib, o'rtacha $N \cdot \log_2 N$ ta solishtirish natijasida saralaydi. Algoritim berilgan massivni saralash uchun uni 2 taga bo'lib oladi. Bo'lib olish uchun ixtiyoriy elementni tanlab undan 2 ta qismga ajratiladi. Lekin o'rtadagi elementni tanlab, massivning teng yarmidan 2 ga ajratgan ma'qul. Tanlangan kalit elementga nisbatan chapdagi va o'ngdagi har bir element solishtiriladi. Kalit elementdan kichiklar chapga, kattalar o'ng tomonga o'tkaziladi (1-rasm). Endi massivning har ikkala tomonida xuddi yuqoridagi amallar takrorlanadi. Ya'ni bu oraliqlarning o'rtasidagi elementlar kalit sifatida olinadi va h.k.

Misol uchun rasmdagi massivni saralash algoritmini ko'rib chiqamiz.

1. Oraliq sifatida 0 dan n-1 gacha bo'lgan massivning barcha elementlarini olamiz.
2. Oraliq o'rtasidagi kalit elementni tanlaymiz, ya'ni

$$\text{key} = (\langle \text{oraliq_boshi} \rangle + \langle \text{oraliq_oxiri} \rangle) / 2, \\ i = \langle \text{oraliq_boshi} \rangle, j = \langle \text{oraliq_oxiri} \rangle.$$



1-rasm. Quicksort algoritmidagi o'rinalmashtirish.

3. Chapdagi **i**-elementni **key** bilan solishtiramiz. Agar **key** kichik bo'lsa, keyingi qadamga o'tamiz. Aks holda $i++$ va shu qadamni takrorlaymiz.

4. O'ngdagi **j**-element bilan **key** solishtiriladi. Agar **key** katta bo'lsa, keyingi qadamga o'tamiz, aks holda **j--** va shu qadamni takrorlaymiz.

5. va **j**-elementlarning o'rni almashtiriladi. Agar **i<=j** bo'lsa, 3-qadamga o'tiladi. Birinchi o'tishdan keyin tanlangan element o'zining joyiga kelib joylashadi.

6. Endi shu ko'rilayotgan oraliqda **key** kalitning chap tomonida elementlar mavjud bo'lsa, ular ustida yuqoridagi amallarni bajarish lozim, ya'ni ko'riladigan oraliq **0** dan **key-1** gacha deb belgilanadi va 2-qadamga o'tiladi. Aks holda keyingi qadamga o'tiladi.

7. Endi shu ko'rilayotgan oraliqda **key** kalitning o'ng tomonida elementlar mavjud bo'lsa, ular ustida yuqoridagi amallarni bajarish lozim, ya'ni ko'riladigan oraliq **key+1** dan **n-1** gacha deb belgilanadi va 2-qadamga o'tiladi. Aks holda algoritm tugaydi.

Shu algoritmgaga misol ko'rib chiqamiz.

Misol: Talabalar ism-sharifi va tartib raqamidan iborat jadvalni quicksort algoritmi bilan saralang va nechta o'rin almashtirish amalga oshirilganini aniqlang.

Dastur kodi:

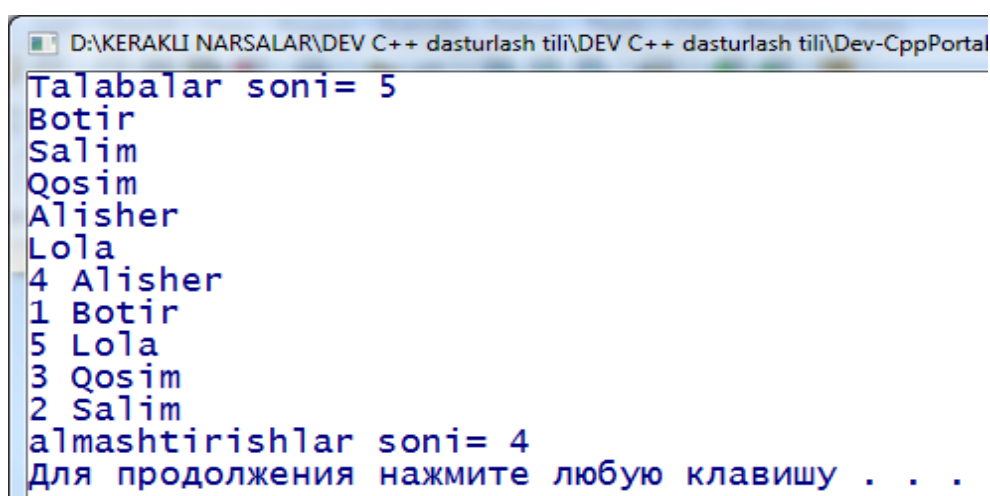
```
#include <cstdlib>
#include <iostream>
#include <string>
using namespace std;
struct talaba
{
    int t;
    string ism;
};
int q=0;
void almashsoni(talaba*a,int boshi,int oxiri)
{
    int i=boshi, j=oxiri;
    talaba x=a[(boshi+oxiri)/2];
    do{
        while(a[i].ism<x.ism)i++;
        while(a[j].ism>x.ism)j--;
        if(i<=j)
        { if(i<j){swap(a[i],a[j]);q++;}
          i++; j--;
        }
    }
    while(i<=j);
    if(i<oxiri)
        almashsoni(a,i,oxiri);
    if(boshi<j)
        almashsoni(a,boshi,j);
}
int main(int argc, char *argv[])
{
    int n;
```

```

cout<<"Talabalar soni= "; cin>>n;
talaba b[n];
for(int i=0; i<n; i++)
{
b[i].t=i+1;
cin>>b[i].ism;
}
almashsoni(b,0,n-1);
for(int i=0; i<n; i++)
cout<<b[i].t<<" "<<b[i].ism<<endl;
cout<<"almashtirishlar soni= "<<q<<endl;

```

Natija:



```

Talabalar soni= 5
Botir
Salim
Qosim
Alisher
Lola
4 Alisher
1 Botir
5 Lola
3 Qosim
2 Salim
almashtirishlar soni= 4
Для продолжения нажмите любую клавишу . . .

```

2-rasm. Dasturning natijasi.

Ushbu laboratoriya ishimizda har bir talaba o'zining oila a'zolarini ismlari Quicksort – tez saralash algoritmi yozdamida saralab, o'rin almashtirishlar isbotlab tusuntirishlari kerak.

8-LABORATORIYA ISHI.

MAVZU: Qatorlar ustida amallar.

Ishdan maqsad: Talabalar daraxtsimon tuzilmalar, binar daraxtlarni e'lon qilish, uning ustida amallar bajarish algoritmlarini tadqiq qilishlari va o'rganishlari kerak, bu algoritmlarning dasturiy realizatsiyasini amalga oshirish ko'nikmasiga ega bo'lishlari kerak.

Qo'yilgan masala: Har bir talaba topshiriq varianti olib, undagi masalaning qo'yilishiga mos binar daraxtlarni tadqiq qilishga oid dasturni ishlab chiqishlari kerak.

Ish tartibi:

- Tajriba ishi nazariy ma'lumotlarini o'rganish;
- Berilgan topshiriqning algoritmini ishlab chiqish;
- C++ dasturlash muhitida dasturni yaratish;
- Natijalarni tekshirish;
- Hisobotni tayyorlash va topshirish.

Daraxt ko'rinishidagi ma'lumotlar tuzilmasi haqida umumiy tushunchalar.

Elementlar va ularning munosabatlaridan iborat elementlar to'plamining ierarxik tuzilmasiga daraxtsimon ma'lumotlar tuzilmasi deyiladi.

Daraxt – bu shunday chiziqsiz bog'langan ma'lumotlar tuzilmasi, u quyidagi belgilari bilan tavsiflanadi:

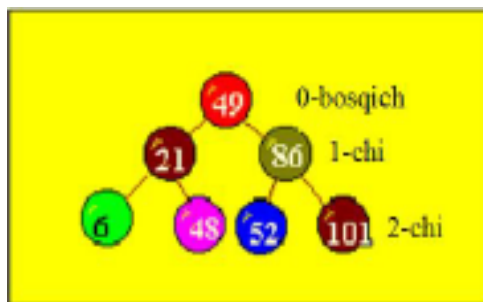
- daraxtda shunday bitta element borki, unga boshqa elementlardan murojaat yo'q. Bu element daraxt ildizi deyiladi;
- daraxtda ixtiyoriy element chekli sondagi ko'rsatkichlar yordamida boshqa tugunlarga murojaat qilishi mumkin;
- daraxtning har bir elementi faqatgina o'zidan oldingi kelgan bitta element bilan bog'langan.

Binar daraxtlarni qurish.

Binar daraxtda har bir tugun-elementdan ko'pi bilan 2 ta shox chiqadi. Daraxtlarni xotirada tasvirlashda uning ildizini ko'rsatuvchi ko'rsatkich berilishi kerak. Daraxtlarni kompyuter xotirasida tasvirlanishiga ko'ra har bir element (binar daraxt tuguni) to'rtta maydonga ega yozuv shaklida bo'ladi, ya'ni kalit maydon, informatsion maydon, ushbu elementni o'ngida va chapida joylashgan elementlarning xotiradagi adreslari saqlanadigan maydonlar.

Shuni esda tutish lozimki, daraxt hosil qilinayotganda, otaga nisbatan chap tomondagi o'g'il qiymati kichik kalitga, o'ng tomondagi o'g'il esa katta qiymatli kalitga ega bo'ladi. Har safar daraxtga yangi element kelib qo'shilayotganda u avvalambor daraxt ildizi bilan solishtiriladi. Agar element ildiz kalit qiymatidan kichik bo'lsa, uning chap shoxiga, aks holda o'ng shoxiga o'tiladi. Agar o'tib ketilgan shoxda tugun mavjud bo'lsa, ushbu tugun bilan ham solishtirish amalga oshiriladi, aks holda, ya'ni u shoxda tugun mavjud bo'lmasa, bu element shu tugunga joylashtiriladi.

Masalan, daraxt tugunlari quyidagi qiymatlarga ega 6, 21, 48, 49, 52, 86, 101. U holda binar daraxt ko'rinishi quyidagi 1-rasmdagidek bo'ladi:



1-rasm. Binar daraxt ko'rinishi.

Natijada, o'ng va chap qism daraxtlari bir xil bosqichli tartiblangan binar daraxt hosil qildik. Agar daraxtning o'ng va chap qism daraxtlari bosqichlarining farqi birdan kichik bo'lsa, bunday daraxt ideal muvozanatlangan daraxt deyiladi. Yuqorida hosil qilgan binar daraxtimiz ideal muvozanatlangan daraxtga misol bo'ladi. Daraxtni muvozanatlash algoritmini sal keyinroq ko'rib chiqamiz. Undan oldin binar daraxtni yaratish algoritmini o'rganamiz.

Misol: C++ dasturlash tilida uzbekcha-ruscha-inglizcha lug'ot dasturini tuzish.

Dastur kodi:

```

#include <cstdlib>
#include <iostream>
#include <string>
using namespace std;
int main(int argc, char *argv[])
{
    string uzb[]={"ruchka","kitob","qalam","sumka"};
    string rus[]={"ruchka","kniga","karandash","rugzag"};
    string eng[]={"pen","book","pencil","bag"};
    string soz,til;
    string massiv[4];
    cout<<"TILNI KIRITING: "; cin>>til;
    if (til=="uzb")
    { for (int i=0; i<4; i++)
      { massiv[i]=uzb[i]; } }
    if (til=="rus")
    { for (int i=0; i<4; i++)
      { massiv[i]=rus[i]; }}
    if (til=="eng")
    { for (int i=0; i<4; i++)
      { massiv[i]=eng[i]; }}
    int s;
    cout<<"so'zni kiriting: "; cin>>soz;
    for (int i=0; i<4; i++)
    { if (soz!=massiv[i])
      { cout<<"xxxxxxxxxxxx"<<endl; }
    }
}

```

```

else
{ for (int i=0; i<4; i++)
{ if (soz==massiv[i])
{ s=i; } }
if (til=="uzb")
{cout<<"RUS: "<<rus[s]<<endl<<"ENG: "<<eng[s]<<endl;}
if (til=="rus")
{cout<<"UZB: "<<uzb[s]<<endl<<"ENG: "<<eng[s]<<endl;}
if (til=="eng")
{cout<<"RUS: "<<rus[s]<<endl<<"UZB: "<<uzb[s]<<endl;}
}}
system("PAUSE");
return EXIT_SUCCESS;}

```

Dastur natijasi:

```

D:\Kerakli Narsalar\DEV C++ dasturlash tili\DEV C++ dasturlash tili\Dev-CppPort
TILNI KIRITING: uzb
so'zni kiriting: kitob
XXXXXXXXXXXX
RUS: kniga
ENG: book
XXXXXXXXXXXX
XXXXXXXXXXXX
Для продолжения нажмите любую клавишу . . .

```

2-rasm. Dastur natijasi.

Ushbu laboratoriya ishini bajarish davomida talabalar quyidagi masalalarni o'zining jurnalidagi tartib raqami bo'yicha dasturlarni tuzishlari kerak:

1. C++ dasturlash tili operatorlarini izohli lug'atini dasturini tuzing (C++ dasturlash tilida).
2. Kompyuter tashqi qurilmalarini o'zbekcha-ruscha-inglizcha lug'atini dasturini tuzing (C++ dasturlash tilida).
3. Inglizcha-o'zbekcha lug'at dasturini tuzing (C++ dasturlash tilida).
4. C++ dasturlash tili amallarini izohli lug'atini dasturini tuzing (C++ dasturlash tilida).
5. TATU FF binosi 3-qavati xonalarini izohli lug'atini dasturini tuzing (C++ dasturlash tilida).
6. Kompyuter ichki qurilmalarini o'zbekcha-ruscha-inglizcha lug'atini dasturini tuzing (C++ dasturlash tilida).
7. O'quv qurollarini o'zbekcha-ruscha-inglizcha lug'atini dasturini tuzing (C++ dasturlash tilida).
8. Kompyuter tashqi qurilmalarini izohli lug'atini dasturini tuzing (C++ dasturlash tilida).
9. Ruscha-o'zbekcha lug'at dasturini tuzing (C++ dasturlash tilida).
10. Mevalarni o'zbekcha-ruscha-inglizcha lug'atini dasturini tuzing (C++ dasturlash tilida).

11. Kompyuter ichki qurilmalarini izohli lug'atini dasturini tuzing (C++ dasturlash tilida).
12. TATU FF binosi 2-qavati xonalarini izohli lug'atini dasturini tuzing (C++ dasturlash tilida).
13. Ruscha-inglizcha lug'at dasturini tuzing (C++ dasturlash tilida).
14. Uy jonivorlarini o'zbekcha-ruscha-inglizcha lug'atini dasturini tuzing (C++ dasturlash tilida).
15. Informatika fanidan izohli lug'at dasturini tuzing (C++ dasturlash tilida).
16. Zamonaviy ta'lim texnologiyalarini izohli lug'atini dasturini tuzing (C++ dasturlash tilida).
17. O'zbekcha-inglizcha lug'at dasturini tuzing (C++ dasturlash tilida).
18. O'rmon jonivorlarini o'zbekcha-ruscha-inglizcha lug'atini dasturini tuzing (C++ dasturlash tilida).
19. O'zbekcha-ruscha lug'at dasturini tuzing (C++ dasturlash tilida).
20. "Ma'lumotlar tuzilmasi" fanidan izohli lugatini dasturini tuzing (C++ dasturlash tilida).
21. O'quv xonasi jihozlarini o'zbekcha-ruscha-inglizcha lug'atini dasturini tuzing (C++ dasturlash tilida).
22. TATU FF binosi 1-qavati xonalarini izohli lug'atini dasturini tuzing (C++ dasturlash tilida).
23. Inglizcha-ruscha lug'at dasturini tuzing (C++ dasturlash tilida).
24. Matematika fanidan izohli lug'at dasturini tuzing (C++ dasturlash tilida).
25. Fizika fanidan izohli lug'at dasturini tuzing (C++ dasturlash tilida).

9-LABORATORIYA ISHI.

MAVZU: To'g'ridan-to'g'ri qo'shish orqali saralash.

Ishdan maqsad: Ushbu laboratoriya ishining maqsadi talabalar qanday saralash usullari va algoritmlari mavjudligini va ularning samaradorliklarini baholashni o'rganishlari kerak. Shu asosda saralash usullarini qiyosiy tahlil qilishlari, C++ dasturlash tilida fayllar bilan ishlashni va ularga oid dasturlar tuzishni o'zlashtirishlari kerak.

Qo'yilgan masala: Talabalar topshiriq variantiga mos saralash usuli yordamida masalani yechish dasturini yaratish ko'nikmasiga ega bo'lishlari kerak.

Ish tartibi:

- Tajriba ishi nazariy ma'lumotlarini o'rganish;
- Berilgan topshiriqning algoritmini ishlab chiqish;
- C++ dasturlash muhitida dasturni yaratish;
- Natijalarni tekshirish;
- Hisobotni tayyorlash va topshirish.

To'g'ridan-to'g'ri qo'shish usuli bilan saralash algoritmi.

Bunday usul karta o'yinida keng qo'llaniladi. Elementlar (kartalar) hayolan "tayyor" $a(1), \dots, a(i-1)$ va boshlang'ich ketma-ketliklarga bo'linadi. Har bir qadamda ($i=2$ dan boshlanib, har bir qadamda bir birlikka oshirib boriladi) boshlang'ich ketma-ketlikdan i -chi element ajratib olinib tayyor ketma-ketlikning kerakli joyiga qo'yiladi.

To'g'ridan-to'g'ri qo'shish orqali saralash algoritmi quyidagicha bo'ladi:

```
for (int i=1;i<n;i++)
{
    x=a[i];
    x ni a[0]...a[i] // oraliqning mos joyiga qo 'shish
}
```

Kerakli joyni qidirish jarayonini quyidagi tartibda olib borish qulay bo'ladi. 2-elementdan boshlab har bir elementni qarab chiqamiz, ya'ni har bir element o'zidan oldin turgan element bilan solishtiriladi. Agar qaralayotgan element kichik bo'lsa, oldinda turgan element bilan o'rin almashadi va yana o'zidan oldinda turgan element bilan solishtiriladi, jarayon shu kabi davom etadi. Bu jarayon quyidagi shartlarning birortasi bajarilganda to'xtatiladi:

3. x elementi oldida uning kalitidan kichik kalitli $a(j)$ elementi chiqqanda.
4. x elementi oldida element qolmaganda.

```
for (int i=1;i<n;i++)
{
    int j=i;
    while(a[j]<a[j-1])
    {
        int t=a[j-1];
        a[j-1]=a[j];
    }
}
```

```
a[j]=t; j=j-1;}}
```

Tanlash orqali saralash algoritmi

Mazkur usul quyidagi tamoyillarga asoslangan:

- Eng kichik kalitga ega element tanlanadi.
- Ushbu element a_0 birinchi element bilan o'rin almashinadi.
- Keyin mazkur jarayon qolgan **n-1**, **n-2** elementlar bilan takrorlanib, to bitta eng “katta” element qolguncha davom ettiriladi.

```
for(int i=0;i<n-1;i++)
for(int j=i+1;j<n;j++)
if(a[i]>a[j])
{
int k = a[j];
a[j]=a[i];
a[i]= k;}
```

Saralashning quyidagicha usullari bor:

- qat'iy (to'g'ridan-to'g'ri) usullar;
- yaxshilangan usullar.

Qat'iy usullarning afzalliklarini ko'rib chiqaylik:

- Bilamizki, dasturlarning o'zlari ham xotirada joy egallaydi. To'g'ridan- to'g'ri saralash usullarining dasturlari qisqa bo'lib, ular tushunishga oson.
- To'g'ridan-to'g'ri saralash usullari orqali saralash tamoyillarining asosiy xususiyatlarini tushuntirish qulay.
- Murakkablashtirilgan usullarda uncha ko'p amallarni bajarish talab qilinmasada, ushbu amallarning o'zlari ham ancha murakkabdir. Garchi yetarlicha katta n larda ulardan foydalanish tavsiya etilmasada, kichik n larda mazkur usullar tezroq ishlaydi.

Shu joyni o'zida qat'iy usullarni ishlash tamoyillariga ko'ra 3 ta toifaga bo'lish mumkin:

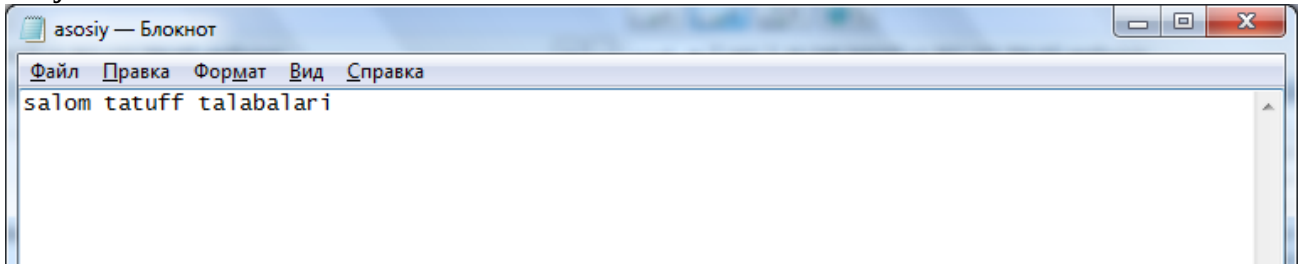
1. To'g'ridan-to'g'ri qo'shish usuli (by insertion);
2. To'g'ridan-to'g'ri tanlash usuli (by selection);
3. To'g'ridan-to'g'ri almashtirish usuli (by exchange).

1-misol: Asosiy nomli .txt fayl ochib, uning ichiga “Salom tatuff talabalari” matnini yozish kerak.

Dastur kodi:

```
#include <cstdlib> #include <iostream> #include <fstream>
using namespace std;
int main(int argc, char *argv[])
{
ofstream f;
f.open("asosiy.txt",ios::out);
f<<"salom tatuff talabalari";
system("PAUSE");
return EXIT_SUCCESS;}
```

Natija:



1-rasm. Asosiy.txt fayl ko'rinishi.

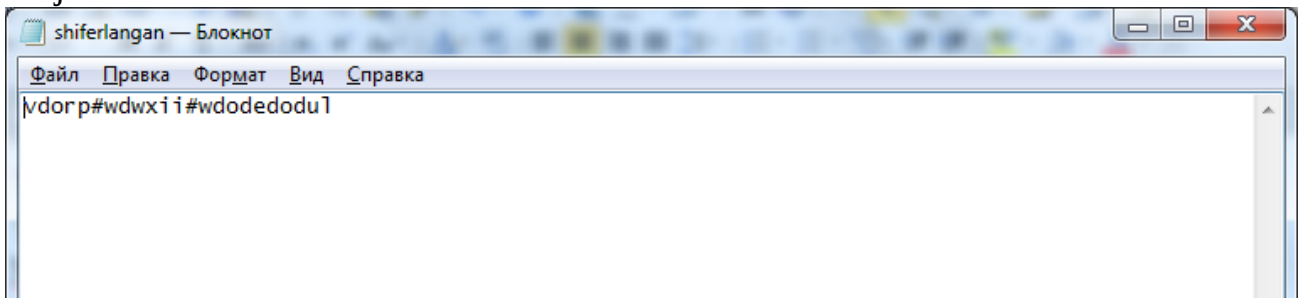
2-misol: Asosiy nomli .txt fayldagi "Salom tatuff talabalari" matnini shiferlangan.txt faylga shiferlab joylaymiz.

Dastur kodi:

```
#include <cstdlib> #include <iostream> #include <fstream> #include <string>
using namespace std;
int main(int argc, char *argv[])
{
    string a;
    ifstream f;
    ofstream y;
    f.open("asosiy.txt",ios::in);
    y.open("shiferlangan.txt",ios::out);
    getline(f,a);
    for(int i=0; i<a.length(); i++)
    {
        int p=int(a[i]);
        y<<char(p+3);
    }
    system("PAUSE");
    return EXIT_SUCCESS;}

```

Natija:



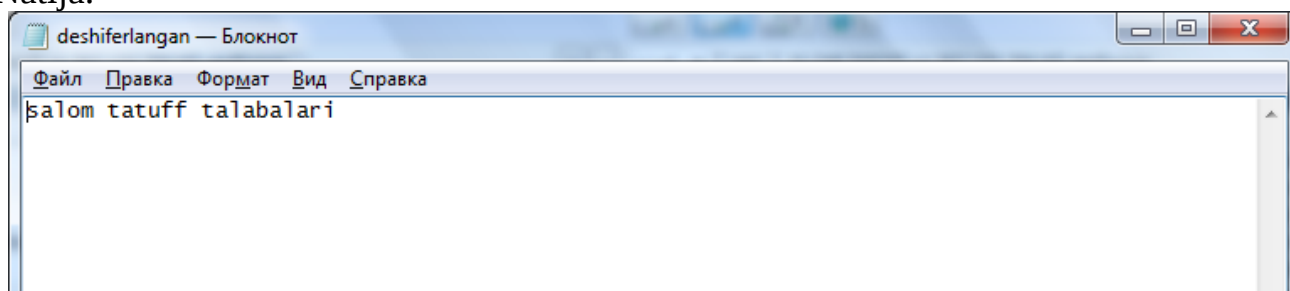
2-rasm. Asosiy.txt fayldagi matning shiferlangan.txt fayldagi shiferlangan ko'rinishi.

3-misol: Shiferlangan.txt faylga shiferlangan matnni deshiferlangan.txt fayga deshiferlab, ya'ni aslni joylashtiramiz.

Dastur kodi:

```
#include <cstdlib> #include <iostream> #include <fstream> #include <string>
using namespace std;
int main(int argc, char *argv[])
{
string a;
ifstream y;
ofstream z;
y.open("shiferlangan.txt",ios::in);
z.open("deshiferlangan.txt",ios::out);
getline(y,a);
for(int i=0; i<a.length(); i++)
{
int p=int(a[i]);
z<<char(p-3); }
system("PAUSE");
return EXIT_SUCCESS;}
```

Natija:



3-rasm. Deshiferlangan.txt fayldagi deshiferlangan ko'rinishi.

Topshiriq:

Ushbu laboratoriya ishimizda talabalar o'zlarining FIOlarini biror nom bilan fayga joylaydilar, shundan so'ng fayldagi FIOsini chaqirib shiferlab, boshqa faylga guruh jurnalidagi nomerlari bo'yicha ma'lumotlarni deshiferlab joylashalari kerak.

FOYDALANILGAN ADABIYOTLAR:

1. Роберт Седжвик. Фундаментальные алгоритмы на С++. Анализ, Структуры данных, Сортировка, Поиск//К.: Изд. «ДиаСофт», 2001.- 688 с.
2. Динман М.И. С++. Освой на примерах//СПб.:БХВ-Петербург, 2006, 384.
3. Шилдт, Герберт. Полный справочник по С#/М. : Изд. дом "Вильямс", 2004, 752 с.
4. Вирт Н. Алгоритмы и структуры программы//М., Мир,
5. Лойко В.И. Структуры и алгоритмы обработки данных. Учебное пособие для вузов.- Краснодар: КубГАУ. 2000. - 261 с., ил.
6. Джон Бентли. Жемчужины программирования. СПб.: Питер, 2002.-272 с.

QO'SHIMCHA ADABIYOTLAR:

1. Акбаралиев Б.Б. Конспект лекций по курсу “Маълумотлар тузилмаси ва алгоритмлар” для студентов по специальности 5521900 “Информатика и информационные технологии”, Ташкент, 2008 г.
2. Акбаралиев Б.Б. Методические указания к практическим занятиям по курсу “ Маълумотлар тузилмаси ва алгоритмлар” для студентов по специальности 5521900 “Информатика и информационные технологии” , Ташкент, 2008 г.
3. Акбаралиев Б.Б. Методические указания к лабораторным работам по курсу “Маълумотлар тузилмаси ва алгоритмлар” для студентов по специальности 5521900 “Информатика и информационные технологии” , Ташкент, 2008 г.
4. Акбаралиев Б.Б. Методические указания к самостоятельной работе по курсу “Структуры данных и алгоритмы ” для студентов по специальности 5521900 “Информатика и информационные технологии” , Ташкент, 2008 г.

Axborot-resurs manbalar

1. <http://www.atdt.uz>
2. <http://www.intuit.ru>
3. <http://www.atdt.tuit.uz>
4. <http://www.ref.uz>
5. www.tuit.uz
6. www.ziyonet.uz
7. itportal.uz
8. <https://www.google.com>