Konteynerlar

Reja:

1. **Konteynerlar haqida.**
2. **Ketma-ket konteynerlar;**
3. **array sinfi;**
4. **vector sinfi;**
5. **deque sinfi;**
6. **Forward_list sinfi;**
7. **list sinfi;**
8. **Xulosa.**

PhD: Mallayev O.U.

Konteynerlar kutubxonasi

- **Konteynerlar kutubxonasi** - bu dasturchilarga **navbat, ro'yxat va stek** kabi keng tarqalgan ma'lumotlar tuzilmalarini osonlikcha amalga oshirishga imkon beradigan sinf andozalari va algoritmlarning universal to'plamidir. Uch xil konteyner mavjud:
- **ketma-ket konteynerlar**,
- **assosiativ konteynerlar** va
- **tartibga solinmagan assotsiativ konteynerlar**.
- Ularning har biri turli xil operatsiyalar to'plamini qo'llab-quvvatlashga mo'ljallangan.

- Konteyner uning elementlari uchun ajratilgan xotirani boshqaradi va ularga to'g'ridan-to'g'ri yoki iteratorlar orqali kirish uchun funksiyalarni ta'minlaydi (ko'rsatkichlarga o'xshash xususiyatlarga ega obyektlardir).

Ketma-ket konteynerlar

- Sinf a’zolariga ketma-ket kirish huquqiga ega bo'lgan ma'lumotlar tuzilmasini amalga oshirishda ketma-ket konteynerlardan foydalaniladi.
- **Ketma-ket konteynerlarni turlari:**

- **array -** statik doimiy massiv;
- **vector –** dynamik doimiy massiv;
- **deque -** ikki tomonlama navbat;
- **forward_list -** bog'langan ro'yxat;
- **list –** ikki tomonlama bog'langan ro'yxat.

array sinfi shabloni

- **template<class T,**
  std::size_t N>
  struct array;

array - N o'lchamdagi massivni o'rab turadigan konteyner.

Ketma-ket konteynerda ishlatiladigan turlar

Tur	Aniqlanishi
value_type	T
size_type	size_t
difference_type	std::ptrdiff_t
reference	value_type&
const_reference	const value_type&
pointer	T*
const_pointer	const T*
iterator	RandomAccessIterator
const_iterator	Константный итератор с произвольным доступом
reverse_iterator	std::reverse_iterator<iterator>
const_reverse_iterator	std::reverse_iterator<const_iterator>

array ning funksiya – a’zolari

Nomi	Izoh
at	Ko'rsatilgan elementga indeks tekshiruvi bilan kirishni ta'minlaydi
operator[]	Belgilangan elementga kirishni ta'minlaydi
front	Birinchi elementga kirishni ta'minlaydi
back	Oxirgi elementga kirishni ta'minlaydi
data (C++11)	Massivning birinchi haqiqiy elementiga ko'rsatgichni qaytaradi

Iteratorlar

Iteratorlar to’plam elementlariga kirishni ta'minlaydi. Iteratorlardan foydalanib, elementlarni takrorlash juda qulay. Iterator turi iterator tomonidan tavsiflanadi. Ammo har bir to’plam uchun iteratorning o'ziga xos turlari mavjud.

Nomi	Izoh
begin, cbegin	Iterator birinchi elementni qaytaradi.
end, cend	Iterator oxirgi elementni qaytaradi.
rbegin, crbegin	Iteratorni birinchi elementga teskarisini qaytaradi.
rend, crend	Oxirgi elementning teskarisini qaytaradi.

array sinfining o’lchamlari va jarayonlari

Nomi	Izoh
empty	Konteynerning qoldirilgan elementlarini tekshirish.
size	Konteyner elementlarining sonini qaytaradi.
max_size	Kenteynerdagi elementlarning maksimal miqdorini qaytaradi.
fill	Konteynerni belgilangan qiymatga to'ldirish.
swap	Tarkibni almashtirish.

array sinfi uchun qayta yuklamagan operatorlar

Nomi	Izoh
operator==	Bir qatordagi qiymatlarni leksikografik jihatdan taqqoslaydi
operator!=	
operator<	
operator<=	
operator>	
operator>=	

```
3 2 1
a b
```

```cpp
#include "stdafx.h"
#include <string>
#include <iterator>
#include <iostream>
#include <algorithm>
#include <array>

int main()
{
    // konstruktor agregat boshlang'ichni ishlatadi
    std::array<int, 3> a1={ {1,2,3} };  // ikki qavs kerak
    std::array<int, 3> a2 = {1, 2, 3};
    // tayinlash operatsiyalari bundan mustasno
    std::array<std::string, 2> a3 = { {std::string("a"), "b"} };

    // umumlashtirilgan algoritmlar qo'llab-quvvatlanadi
    std::sort(a1.begin(), a1.end());
    std::reverse_copy(a2.begin(), a2.end(),
        std::ostream_iterator<int>(std::cout, " "));

    // qo'llab-quvvatlanadigan qatorga qo'yildi
    std::cout<<std::endl;
    for(auto& s: a3)
        std::cout << s << ' ';
    getchar();
}
```

Masala:

- ***Talaba*** nomli class yaratish. Konteynerlarning array sinfi yordamida to’plamlar yartish. **Array** ning maxsus metodidan foydalanib, Talabalar to’plamini talabalarning yoshlari bo’yicha saralovchi dastur tuzish.
- **Masalani yechish g‘oyasi:**
- talaba nomli yangi nolar fazosi yartiladi. Uning tarkibiga Talaba classi va Ism, Fam, Yosh, Step nomli to’plamlar joylashtiriladi. Talaba classining ruxsat berilmagan maydonlaridan foydalanish uchun friend funksiya yaratiladi.

```
1 - talaba haqida haqida
Ismi: Islom
Familiyasi: Salimov
Yoshi: 22
Stependiyasi: 400000
2 - talaba haqida haqida
Ismi: Oqil
Familiyasi: Qodirov
Yoshi: 19
Stependiyasi: 600000
3 - talaba haqida haqida
Ismi: Komil
Familiyasi: Tashev
Yoshi: 20
Stependiyasi: 120000

Yoshlari bo'yicha saralangan talabalar ro'yxati
1 - talaba haqida haqida
Ismi: Oqil
Familiyasi: Qodirov
Yoshi: 19
Stependiyasi: 600000
2 - talaba haqida haqida
Ismi: Komil
Familiyasi: Tashev
Yoshi: 20
Stependiyasi: 120000
3 - talaba haqida haqida
Ismi: Islom
Familiyasi: Salimov
Yoshi: 22
Stependiyasi: 400000
```

vector sinfi

- **Vector sinfining shabloni:**

```
template< class T,
class Allocator = std::allocator<T>
 > class vector;
namespace pmr {
template <class T>
using vector = std::vector<T,
std::polymorphic_allocator<T>>;
}
```

- 1) **std :: vector** - o'zgaruvchan kattalikdagi ketma-ket konteynerni qamrab oluvchi massiv.
- 2) **std :: pmr :: vector** shablon taxalluslari polimorf ajratuvchi yordamida.
- **Allocator** - elementlar uchun xotira ajratishda ishlatiladigan sinf.

vector sinfining funksiya – a’zolari

Nomi	Izoh
at	Ko'rsatilgan elementga indeks tekshiruvi bilan kirishni ta'minlaydi
operator[]	Belgilangan elementga kirishni ta'minlaydi
front	Birinchi elementga kirishni ta'minlaydi
back	Oxirgi elementga kirishni ta'minlaydi
data (C++11)	Massivning birinchi haqiqiy elementiga ko'rsatgichni qaytaradi
operator=	Konteynerdagi qiymatlarni o'rnatadi
assign	Konteynerdagi qiymatlarni o'rnatadi

vector sinfining modifikatorlari

Nomi	Izoh
clear	Konteynerni tozalaydi.
insert	Konteynerga element qo’shadi.
emplace (C++11)	Elementlarni "joyida" quradi va berilgan pozitsiyadan boshlab ularni joylashtiradi.
erase	Konteynerdan element ochirish.
push_back	Oxiriga element qo'shadi.
emplace_back (C++11)	Konteyner oxiridan elementlarni qo’shadi.
pop_back	Oxirgi elementni o’chirish.
resize	Saqlangan eleentlar sonini o'zgartiradi.
swap	Tarkibni almashtirish.

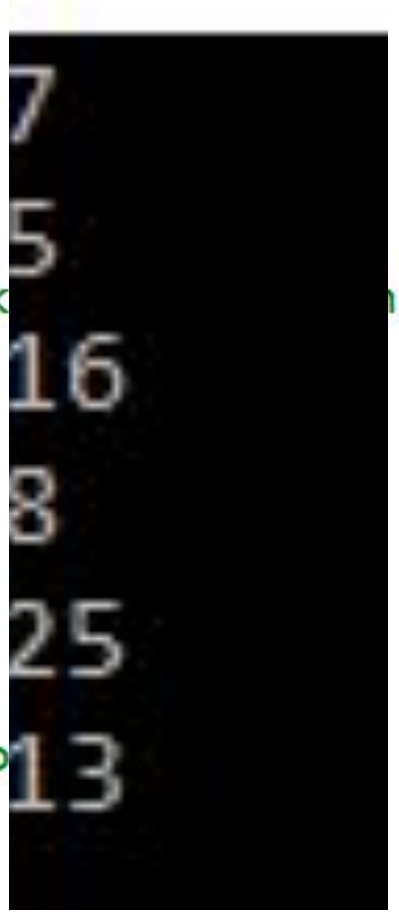
vector sinfi uchun qayta yuklanmagan operatorlar

Nomi	Izoh
operator==	Bir qatordagi qiymatlarni leksikografik jihatdan taqqoslaydi
operator!=	
operator<	
operator<=	
operator>	
operator>=	

```cpp
#include "stdafx.h"
#include <iostream>
#include <vector>

int main ( ) {
    // Butun sonlarni o'z ichiga olgan vek
    int A[] = {7, 5, 16, 8};
    std::vector<int> v(4);
     v[0] = A[0];
     v[1] = A[1];
     v[2] = A[2];
     v[3] = A[3];
    // Vektorga yana ikkita butun sonni qo
    v.push_back(25);
    v.push_back(13);

    // Qiymatlar chiqishi uchun vektor orqali o'tish
    for ( int n : v ) {
        std::cout << n << '\n';
    }
    getchar();
}
```



- **Masala:** Berilgan int turidagi to’plamda – ishorali elementlarning eng kattasini o’chiruvchi dastur tuzing.

- **Masalani yechish g‘oyasi:**
- **talaba** nomli yangi nomlar fazosi yartiladi. Uning tarkibiga **Talaba** classi va **Ism, Fam, Yosh, Step** nomli to’plamlar joylashtiriladi. Talaba classining ruxsat berilmagan maydonlaridan foydalanish uchun **friend** funksiya yaratiladi.

```
void Max_Vector(std::vector<int> &A, int n){
    for (int i = 0; i < n; i++)
     {
         if(A[i]<0 && max<A[i]){max = A[i]; index=i;}
```

```
Berilgan int turidagi toTplamda - ishorali
elementlarning maksimalini oTchiruvchi dastur
5
1 son: 4
2 son: -55
3 son: -3
4 son: -4
5 son: 23
To'plam elementlari
4 -55 -3 -4 23
Max= -3
To'plam elementlari
4 -55 -4 23
```

```
"<<endl;
    Vectorlar::Max_Vector(A,n);
    Vectorlar::Delete_Max_Vector(A);
    Vectorlar::get_Vector(A,n);
    getchar();
}
```

deque sinfi

- **Deque sinfining shabloni:**
- **template<class T,**
- **class Allocator = std::allocator<T>**
- **> class deque;**
- **std :: deque** (ikki tomonlama navbat) - indekslangan ketma-ket konteyner, bu sizga elementlarni boshidan va oxiridan tezda kiritish va olib tashlash imkonini beradi. Bundan tashqari, ikki tomonlama navbatning ikkala uchiga o'rnatish va o'chirish ko'rsatgichlar va boshqa elementlarga bog'lanishlarni qoldiradi.
- **std :: vektor**dan farqli o'laroq, **deque** elementlari doimiy ravishda saqlanmaydi: odatda bu belgilangan o'lchamdagi ajratilgan qatorlar to'plamidan foydalanib amalga oshiriladi. **deque** avtomatik ravishda qayta ishlanadi, kerak bo'lganda kengayadi. **deque** kengaytmasi **std :: vector** kengaytmasiga qaraganda qulayroq, chunki u mavjud elementlarni yangi xotiraga nusxalashni talab qilmaydi.

deque sinfining funksiya – a’zolari

Nomi	Izoh
at	Ko'rsatilgan elementga indeks tekshiruvi bilan kirishni ta'minlaydi
operator[]	Belgilangan elementga kirishni ta'minlaydi
front	Birinchi elementga kirishni ta'minlaydi
back	Oxirgi elementga kirishni ta'minlaydi
get_allocator	Bog'langan ajratuvchini qaytaradi
operator=	Konteynerdagi qiymatlarni o'rnatadi
assign	Konteynerdagi qiymatlarni o'rnatadi

deque sinfining modifikatorlari

Nomi	Izoh
clear	Konteynerni tozalaydi.
insert	Konteynerga element qo’shadi.
emplace (C++11)	Elementlarni "joyida" quradi va berilgan pozitsiyadan boshlab ularni joylashtiradi.
erase	Konteynerdan element ochirish.
push_back	Oxiriga element qo'shadi.
emplace_back (C++11)	Konteyner oxiridan elementlarni qo’shadi.
pop_back	Oxirgi elementni o’chirish.
resize	Saqlangan elementlar sonini o'zgartiradi.
swap	Tarkibni almashtirish.
push_front	Ro'yxatning boshiga elementlarni joylashtiradi.
emplace_front (C++11)	Ro'yxatning boshidan boshlab, elementlarni yaratadi.
pop_front	Birinchi elementni o’chirish.

deque sinfi uchun qayta yuklanmagan operatorlar

Nomi	Izoh
operator==	Bir qatordagi qiymatlarni leksikografik jihatdan taqqoslaydi
operator!=	
operator<	
operator<=	
operator>	
operator>=	

- **Masala:**
- Berilgan **string** turidagi to’plamda eng uzun so’zning belgilar sonini aniqlovchi dastur tuzing.

- **Masalani yechish g‘oyasi:**
- **deque** nomli yangi nolar fazosi yartiladi. Uning tarkibiga **Ism, Fam, Yosh, Step** nomli to’plamlar joylashtiriladi. Talaba classining ruxsat berilmagan maydonlaridan foydalanish uchun **friend** funksiya yaratiladi.

```
int main()
{
Berilgan string turidagi toTplamda eng
uzun soTzni belgilar sonini aniqlovchi dastur
To'plam elementlari soni n= 3
1 - satrni kiriting:
Salom dunyo
2 - satrni kiriting:
Hello world
3 - satrni kiriting:
Assalomu aleykum
To'plam elementlari
1 - satr:
        Salom dunyo
2 - satr:
        Hello world
3 - satr:
        Assalomu aleykum

Salom dunyo Hello world Assalomu aleykum

 So'zlarning uzunliklari:
5 5 5 5 8 7 Max= 8
}
```

forward_list sinfi

- **forward_list sinfining shabloni:**
- **template <class T,**
  class Allocator = std::allocator<T>
- **> class forward_list;**
- **forward_list** - konteynerdan elementlarni kiritish va olib tashlash mexanizmini ta'minlaydigan **sinf**. Tez tasodifiy kirish qo'llab-quvvatlanmaydi. U bir yo'naltirilgan ro'yxat sifatida amalga oshiriladi va C tilidagi shunga o'xshash dastur bilan solishtirganda qo'shimcha xarajatlarga ega emas: **std :: list** dan farqli o'laroq, ushbu turdagi konteyner ikki tomonlama iteratsiyani qo'llab-quvvatlamaydi.

forward_list sinfining o’lchamlari va jarayonlari

Nomi	Izoh
merge	Ikkita tartiblangan ro'yxatlarni birlashtirish.
splice_after	Elementlarni boshqa forward_listdan ko'chiradi.
remove remove_if	Ma'lum belgilarga javob beradigan elementlarni olib tashlaydi.
reverse	Elementlarning tartibini o'zgartiradi.
unique	Ketma-ket takrorlanadigan elementlar o'chiriladi.
sort	Elementlarni tartiblash.

- **Masala:**
- Berilgan **int** turidagi to’plam qiymatlarining raqamlari yig’indisini Z to’plamga joylashtiruvchi va ularni ekranga chiqaruvchi dastur tuzing.

- **Masalani yechish g‘oyasi:**
- **forward_list** konteyneri to’plami yaratiladi. **forward_list** ning **push_front( )** iteratoridan foydalanib, A to’plamga qiymatlar o’zlashtiriladi. **Z** to’plamga esa **A** to’plam qiymatlarini raqamlari yig’indisi yoziladi.

```
int main()
{
    cou
    int

    std
    std
    Set

    aut
    int
    cou                                          1;
    whi

    s+
    }
    cou                                          1;
    for                                          ; i++)
    {

    }
    get
}
```

```
Forward_list
5
1 son: 222222
2 son: 5555
3 son: 3333
4 son: 999
5 son: 1111111
A to'plam qiymatlari
1111111
999
3333
5555
222222
Z to'plam qiymatlari
12
20
12
27
7
```

list sinfi

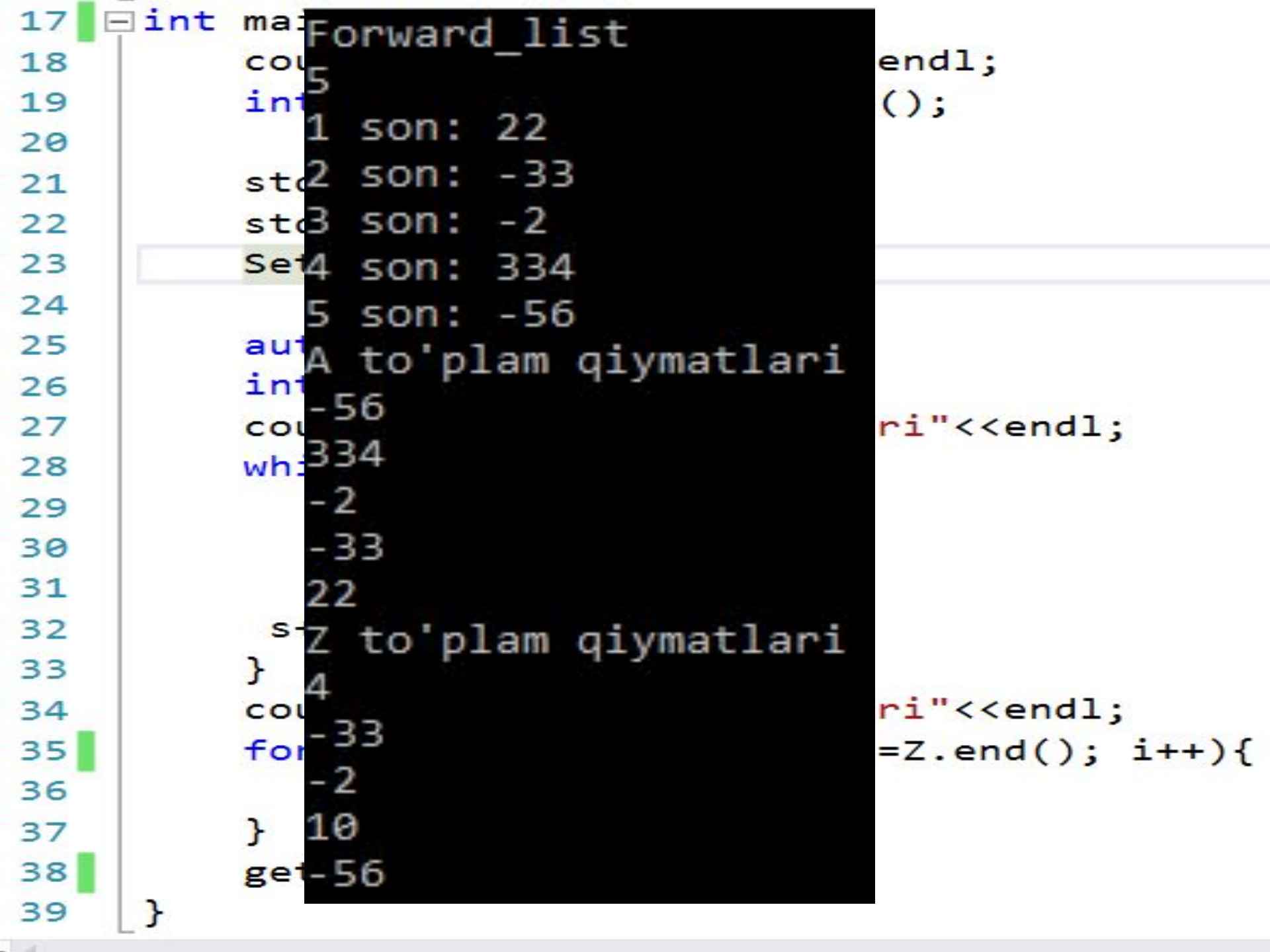
- **list sinfi shabloni:**

```
template < class T,
           class Allocator = std::allocator<T>
         > class list;
```

- **List** - bu konteynerning har qanday pozitsiyasidan elementlarni tezda kiritish va olib tashlashni qo'llab-quvvatlaydigan sinf. Tez tasodifiy kirish qo'llab-quvvatlanmaydi. Ikkala bog'langan ro'yxat sifatida amalga oshiriladi. **std::forward_list**-dan farqli o'laroq, ushbu konteyner ikki tomonlama iteratsiyani ta'minlaydi, shu bilan birga foydalanilgan xotiraga nisbatan unumli emas.

- **Masala:**
- Berilgan **int** turidagi to’plam qiymatlarini juftlarini Z to’plamga joylashtiruvchi va ularni ekranga chiqaruvchi dastur tuzing.

- **Masalani yechish g‘oyasi:**
- **list** konteyneri to’plami yaratiladi. **list** ning **push_back()** iteratoridan foydalanib, **A** to’plamga qiymatlar o’zlashtiriladi. **Z** to’plamga esa **A** to’plam qiymatlarini juftlari yoziladi.


Forward_list
5
1 son: 22
2 son: -33
3 son: -2
4 son: 334
5 son: -56
A to'plam qiymatlari
-56
334
-2
-33
22
Z to'plam qiymatlari
4
-33
-2
10
-56

Foydalanilgan adabiyotlar

- https://ru.cppreference.com/w/cpp/container
- https://ru.cppreference.com/w/cpp/container/array
- https://ru.cppreference.com/w/cpp/container/vector
- https://ru.cppreference.com/w/cpp/container/deque
- https://ru.cppreference.com/w/cpp/container/forward_list
- https://ru.cppreference.com/w/cpp/container/list

Xulosa

C++ da 3 xil konteynerlar mavjud:

- **ketma-ket** konteynerlar,
- **assosiativ** konteynerlar va
- **tartibga solinmagan assotsiativ** konteynerlar.

- **Ketma-ket konteynerlar:**

array - statik doimiy massiv;

vector – dinamik doimiy massiv;

deque - ikki tomonlama navbat;

forward_list - bog'langan ro'yxat;

list – ikki tomonlama bog'langan ro'yxat.