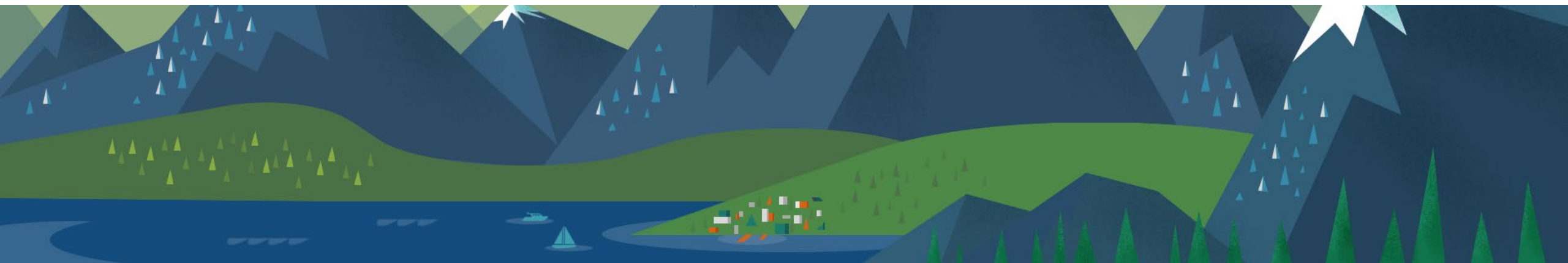


# C++da dinamik massivlar(ko'rsatkichlar)



# Amaliy mashg'ulotning maqsadi

- C++da dinamik massivlar (ko'rsatkichlar) bilan ishlatish ko'nikmasiga ega bo'lish.

# Dinamik massivlar ?



# Yechim usuli

- Etishimiz mumkin agar 5000 element (int) beradigan bo'lsak dastur talabai qondirilishi mumkin

*\* Lekin estan chiqarish kerak emas massivlar o'ziga har bir o'zgarish uchun RAM hotira talab etadi ( $5\ 000 * 4 \text{ (int)} = 20\ 000 \text{ byte}$ ) teng bo'ladi*

*\* Dasturda 10 element ishlatilsa 40 byte hajimni egalagan bo'lamiz 1960 byte bo'sh qoladi.*

```

1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     setlocale(LC_ALL, "eng");
7
8     const int SizeOfArray = 5000;
9     int arrWithDigits[SizeOfArray] = {};
10    cout << "Massivning hotirada olgan joy hajmi " << sizeof(arrWithDigits) << " bayt" << endl;
11
12    int amount = 0;
13    cout << "massivga nechta son kiritasiz? ";
14    cin >> amount;
15    cout << "talab etilatigan hajm" << amount * sizeof(int) << " bayt" << endl;
16
17    for (int i = 0; i < amount; i++)
18    {
19        cout << i + 1 << "-son: ";
20        cin >> arrWithDigits[i];
21    }
22    cout << endl;
23
24    for (int i = 0; i < amount; i++)
25    {
26        cout << arrWithDigits[i] << " ";
27    }
28    cout << endl;
29    return 0;
30 }

```

/tmp/6rM9ASpJ37.o

Massivning hotirada olgan joy hajmi 20000 bayt

massivga nechta son kiritasiz? 5

talab etilatigan hajm20 bayt

1-son: 1

2-son: 300

3-son: 450

4-son: 900

5-son: 8000

1 300 450 900 8000

```

1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     setlocale(LC_ALL, "eng");
7
8     const int SizeOfArray = 5000;
9     int arrWithDigits[SizeOfArray] = {};
10    cout << "Massivning hotirada olgan joy hajmi " << sizeof(arrWithDigits) << " bayt" << endl;
11
12    int amount = 0;
13    cout << "massivga nechta son kiritasiz? ";
14    cin >> amount;
15    cout << "talab etilatigan hajm" << amount * sizeof(int) << " bayt" << endl;
16
17    for (int i = 0; i < amount; i++)
18    {
19        cout << i + 1 << "-son: ";
20        cin >> arrWithDigits[i];
21    }
22    cout << endl;
23
24    for (int i = 0; i < amount; i++)
25    {
26        cout << arrWithDigits[i] << " ";
27    }
28    cout << endl;
29    return 0;
30 }

```

Standart kutubxona funksiyasi **sizeof()** da biz e'lon qilingan **arrWithDigits** qatorini 10-qatorga o'tkazamiz.

15-satrdan **amount \* sizeof(int)** ifodasi **10 \* 4** ga aylanadi, chunki **sizeof(int)** funksiyasi 4 ni qaytaradi (int tipidagi baytlardagi o'lchamini ifodalaydi).

# Hotirani yangilash va o'chirish?

- C++ da new va delete operatorlar mavjud
- **new-(berilgan hotirani ajratib olish)**
- **delete –(berilgan hotirani o'chirib tashlash)**

```

1  #include <iostream>
2  #include <locale>
3  using namespace std;
4  int main()
5  {
6  setlocale(LC_ALL, "rus");
7  int sizeOfArray = 0; // Massiv hajmi (foydalanuvchi tomonidan kiritiladi)
8  cout << "Massivlarni hosil qilish uchun, uning hajmini kiriting: ";
9  cin >> sizeOfArray;
10 // etibor bering! int* arrWithDigits - ko'rsatkichlarni e'lon qilish
11 // new ajratishi kerak bo'lgan hotira aynan shu shakilda amalga
   oshiriladi
12 int* arrWithDigits = new int [sizeOfArray];
13 for (int i = 0; i < sizeOfArray; i++)
14 {
15     arrWithDigits[i] = i + 1;
16     cout << arrWithDigits[i] << " ";
17 }
18 cout << endl;
19 delete [] arrWithDigits; // bo'shatilgan hotira
20 return 0;}

```

quyidagicha dekodlash mumkin: new (xotirani ajratish) int (butun sonlarni saqlash uchun) [sizeOfArray] (sizeOfArray miqdorida).

12 qatorda foydalanuvchi o'zi raqamlarni kiritish imkoniyatiga ega **arrWithDigits**-ko'rsatkich

**new** tegishli bo'lmagan barcha yecheykalar o'chirib tashaladi



