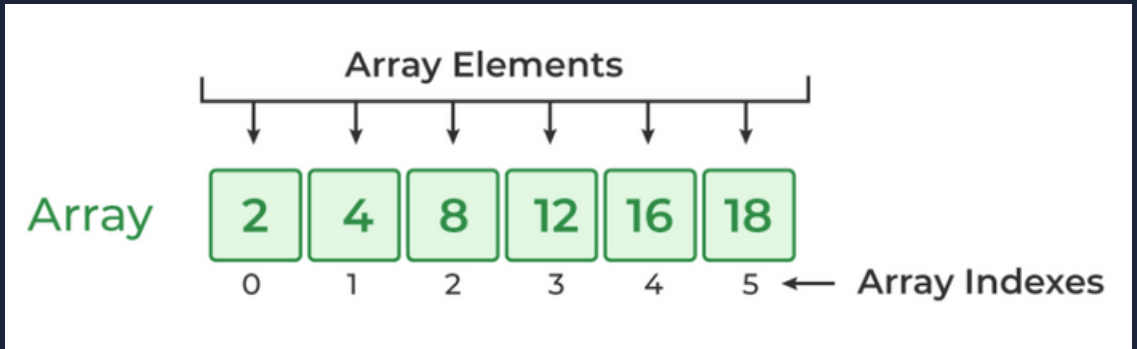

TENX

ARRAYS
(MASSIVLAR)

Massivlar bir necha qiymatlarning har birini alohida o'zgaruvchilarga o'zlashtirish o'rniga 1 ta o'zgaruvchiga saqlash imkonini beradi.

C++ da massivlar deb o'xshash ma'lumot elementlarining belgilangan o'lchamdagi to'plamiga aytiladi



Massivlarni yaratishda huddi o'zgaruvchi yaratish kabi yo'l tutiladi va unga qo'shimcha ravishda massiv nomi yoniga [] (to'rtburchak qavs) ichiga massivning nechta element qabul qilishini ya;ni o'lchamini kirtish orqali amalga oshiriladi.

```
dataType arrayName [arraySize]
```

Massiv elementlari esa tenglik belgisi (=) dan keyin {} ichida har birini vergul (,) bilan ajratish orqali ketma-ket kiritiladi

```
int myNum[3] = {10, 20, 30};

string cars[4] = {"Volvo", "BMW", "Ford", "Mazda"};
```

{} ichidagi elementlar soni massiv yaratilayotganda kiritilgan elementlar sonidan katta bo'lmasiligi kerak. Aks holda xatolik beradi.

Agar massiv nechta elementdan tashkil topishi aniq bo'lmasa [] ichini shunchaki bo'sh qoldirish kifoya

```
int myNum[] = {10, 20, 30, 40, 50};
```

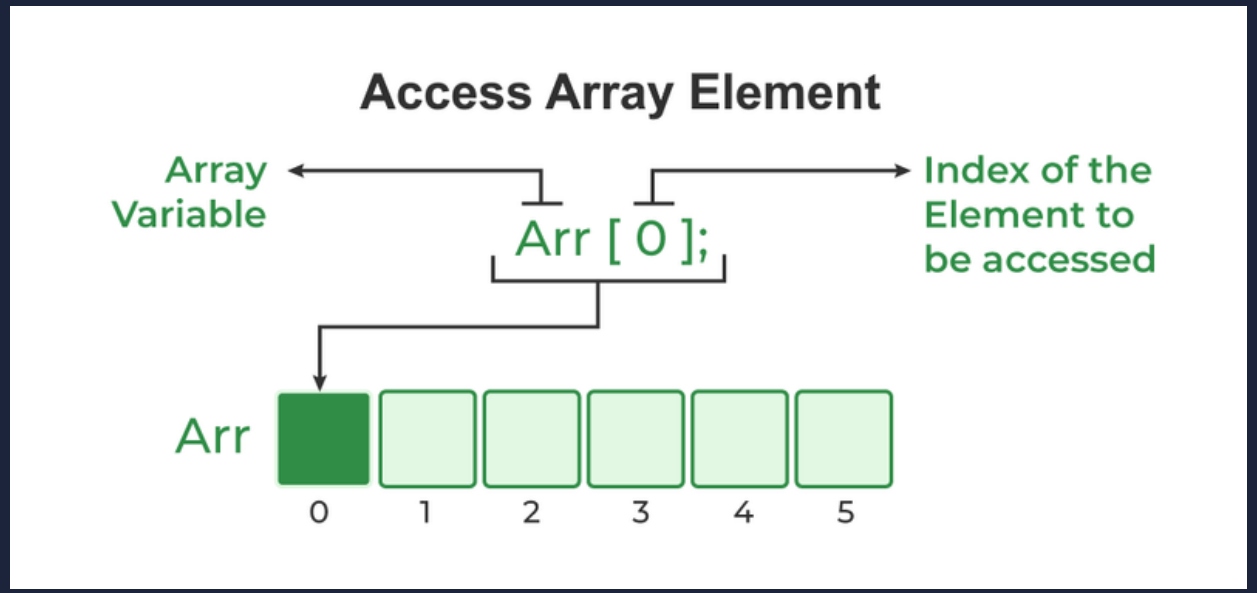
Agar massiv o'lchami berilib, uning elementlari yetarlicha berilmasa qolgan o'rinlari bo'sh qiymatlar bilan avtomatik to'ldiriladi.

```
int nums[5] = {1};
```

MASSIV ELEMENTLARIGA MUROJAAT QILISH

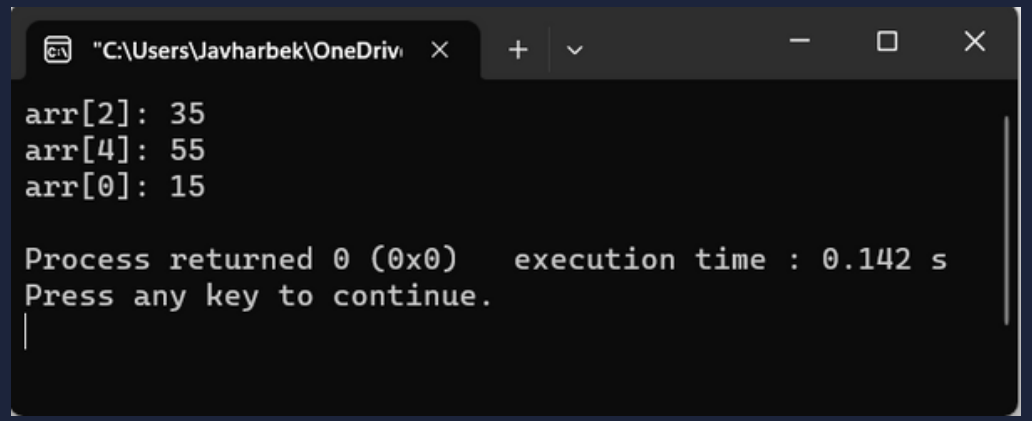
Massiv elementiga murojaat qilish uchun [] ichiga murojaat qilinayotgan element indeksi kiritilib massiv nomidan keyin yoziladi

```
arrayName [index]
```



MASSIV ELEMENTLARIGA MUROJAAT QILISH

```cpp
#include <iostream>

using namespace std;


int main()
{
    // massiv e'lon qilish
    int arr[5] = { 15, 25, 35, 45, 55 };

    // 2-indexdagi elementga murojaat qilish
    cout << "arr[2]: " << arr[2] << endl;

    // 4-indexdagi elementga murojaat qilish
    cout << "arr[4]: " << arr[4] << endl;

    // 0-indexdagi elementga murojaat qilish
    cout << "arr[0]: " << arr[0] << endl;

    return 0;
}
```



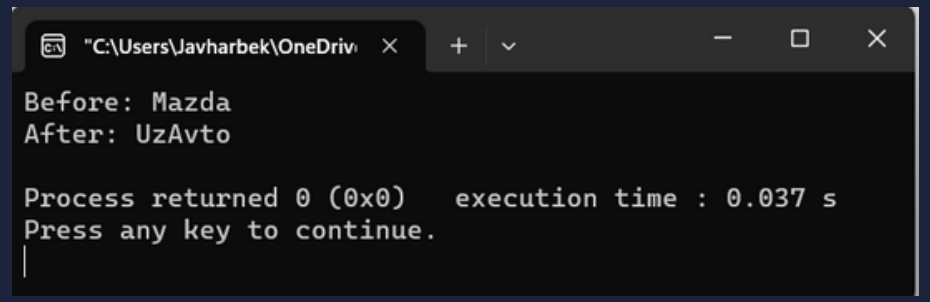
TENX

MASSIV ELEMENTI QIYMATINI O'ZGARTIRISH

Massiv elementiga murojaat qilish bilan birgalikda uning qiymatini o'zgartirish imkoniyatiga egamiz. Buning uchun murojaat qilingandan so'ng tenglik belgisi(=) dan keyin yangi qiymat beriladi

```
arrayName [index] = newValue;
```

```
// massiv e'lon qilish
string cars[4] = {"Volvo", "BMW", "Ford", "Mazda"};

cout << "Before: " << cars[3] << endl;

// 3-indeksdagi element qiymatini o'zgartish
cars[3] = "UzAvto";

cout << "After: " << cars[3] << endl;
```

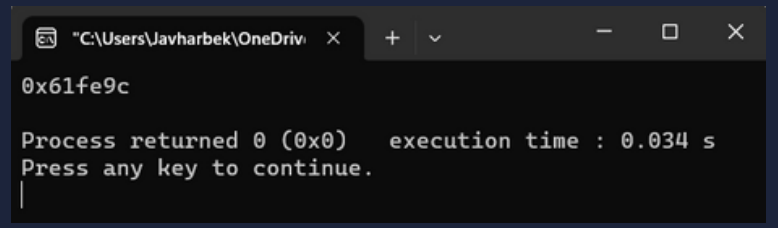


TENX

ARRAY TRAVERSAL
(MASSIV ICHIDA YURIB CHIQISH)

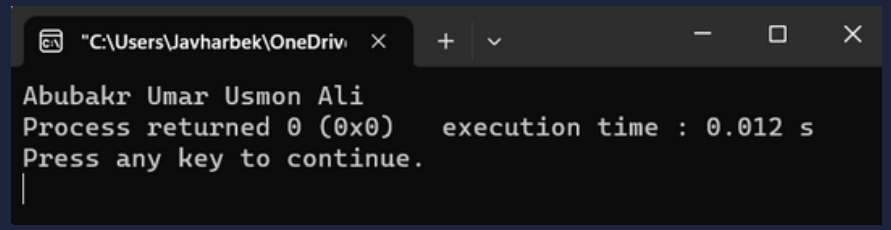
Agar biz yaratgan massivni shundoq ekranga chiqarsak, bizga elementlarni emas, xotiradagi joy manzilni ko'rsatadi

```
// massiv e'lon qilish
string names[4] = {"Abubakr", "Umar", "Usmon", "Ali"};

cout << names << endl;
```



Yaratilgan massivni elementlarini ko'rish uchun, uning ichida har bir elementiga murojaat qilib, yurib chiqishimiz kerak. Buni for sikl orqali bajaramiz

```
// massiv e'lon qilish
string names[4] = {"Abubakr", "Umar", "Usmon", "Ali"};

for(int i = 0; i < 4; i++) {
    cout << names[i] << " ";
}
```



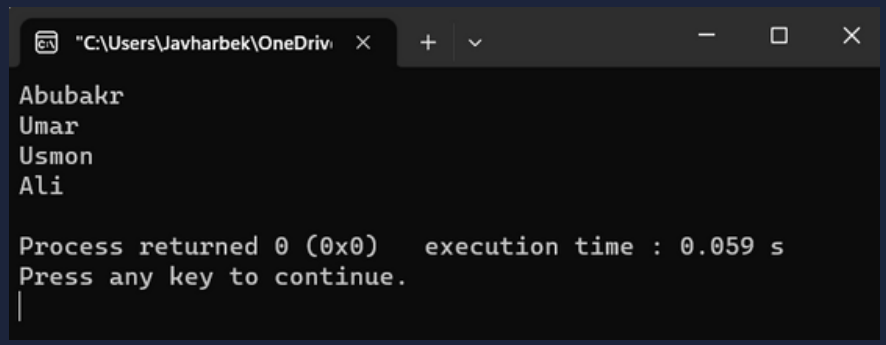
TENX

FOR-EACH

Shuningdek, massivlar uchun alohida mo'ljallangan for-each sikli ham mavjud

```
for (type variableName : arrayName) {
    // sikl tanasi
}
```

```
// massiv e'lon qilish
string names[4] = {"Abubakr", "Umar", "Usmon", "Ali"};

// for-each
for (string i : names) {
  cout << i << "\n";
}
```

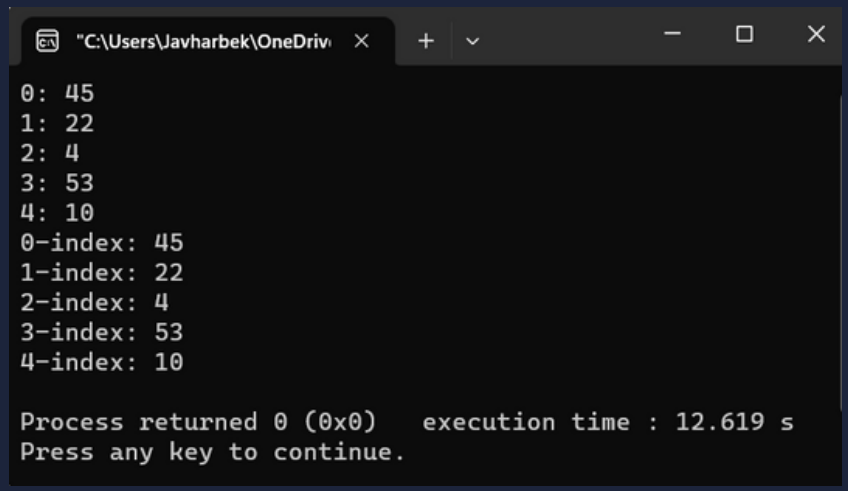


TENX

MASSIV ELEMENTLARIGA QIYMAT BERISH

Biz bo'sh massiv ochib, unga keyinchalik qiymat ya'ni elementlarini kritishimiz ham mumkin.

```cpp
int nums[5];

// Elementlarni qabul qilish
for(int i = 0; i < 5; i++) {
    cout << i << ": ";
    cin >> nums[i];
}

// Elementlarni ekranga chiqarish
for(int i = 0; i < 5; i++) {
    cout << i << "-index: " << nums[i] << endl;
}
```

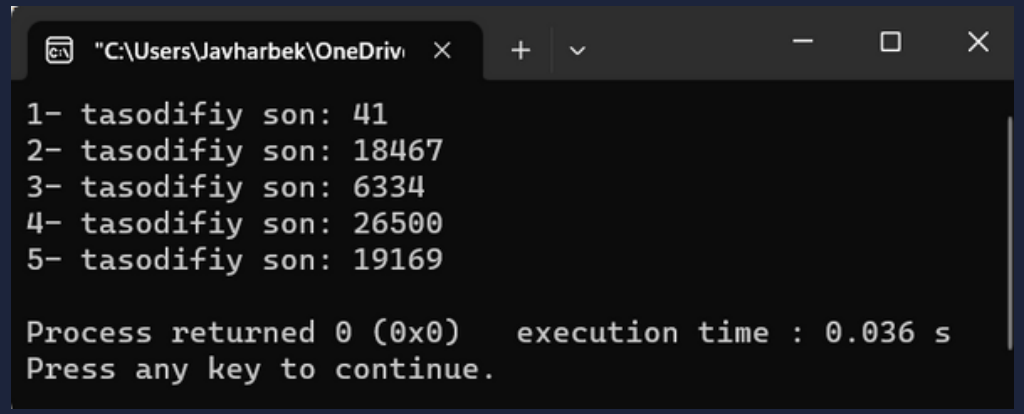


TENX

rand()

Ushbu funkiya random (tasodfiy) sonlarni hosil qilish uchun ishlatiladi. Bu funksiya har chaqirilganda ma'lum ketma-ketlikda qiymatlar qaytaradi. Qaytaruvchi qiymat oralig'i [0, 32767]
Ushbu funksiyadan foydalanish uchun <cstdlib> kutubxonasi cahqirilishi kerak

```
for (int i = 1; i <= 10; i++) {
    int tasodiySon = rand();
    cout << i << "- tasodifiy son: ";
    cout << tasodiySon << endl;
}
```



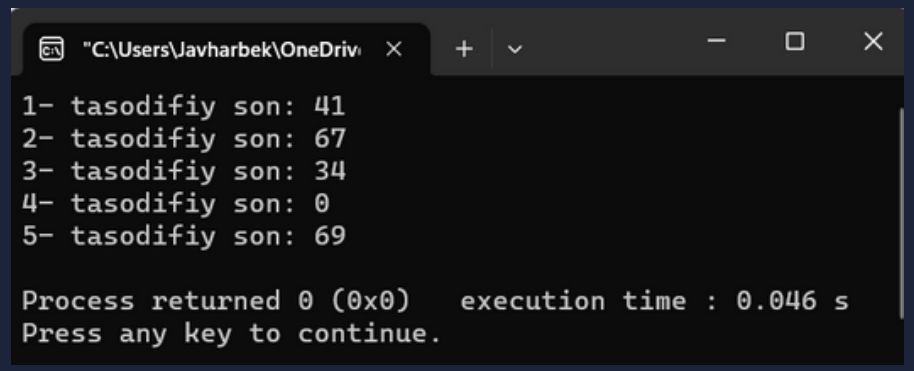
TENX

Agar bizga ma'lum sonlar oralig'ida tasodifiy sonlar kerak bo'lsa, quyidagicha amalga oshiriladi:

rand() % N

Ushbu ifoda [0, N-1] ora'ligidagi tasodifiy sonlarni qaytaradi

```cpp
// 0 dan 100 gacha bo'lgan tasodifiy sonlar

for (int i = 1; i <= 5; i++) {
    int tasodiySon = rand() % 100;
    cout << i << "- tasodifiy son: ";
    cout << tasodiySon << endl;
}
```

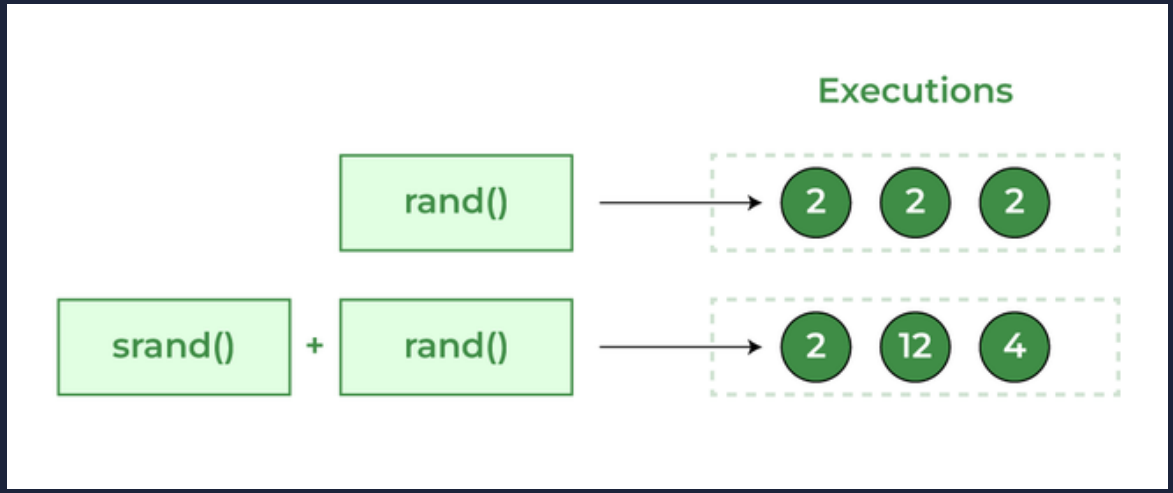


TENX

srand()

Biz rand() funksiyasini qayta qayta chaqirganimizda bir xil ketma-ketlikdagi sonlarni qabul qilyapmiz.

Agar biz har chaqirganimizda turli ketma-ketlikda kelishini xohlasak rand() funksiyasiga qo'shimcha srand() funksiyasiga murojaat qilamiz. Bu funksiya ketma-ketlikni hosil qilish uchun boshlang'ich nuqtani belgilaydi.



srand() funksiyasi o'ziga 1 ta parametr qabul qiladi. Bu parametr tasodifiy sonlar ketma-ketligi uchun asos vazifasini bajaradi.

```
srand(unsigned seed)
```

Amaliyotda srand() funksiyasi time() funksiyasi bilan birgalikda ishlatiladi. Chunki time() funksiyasi har safar o'zgarib turadigan qiymat qaytaradi. Bu funksiyani ishlatish uchun <time.h> kutubxonasi chaqilishi shart.

```
#include <iostream>
#include <cstdlib>
#include <time.h>

using namespace std;

int main()
{
    srand(time(0));

    for (int i = 1; i <= 5; i++) {
        int tasodiySon = rand() % 100;
        cout << i << "-tasodifiy son: ";
        cout << tasodiySon << endl;
    }

    return 0;
}
```

#1

```
1-tasodifiy son: 52
2-tasodifiy son: 24
3-tasodifiy son: 46
4-tasodifiy son: 11
5-tasodifiy son: 10
```

#2

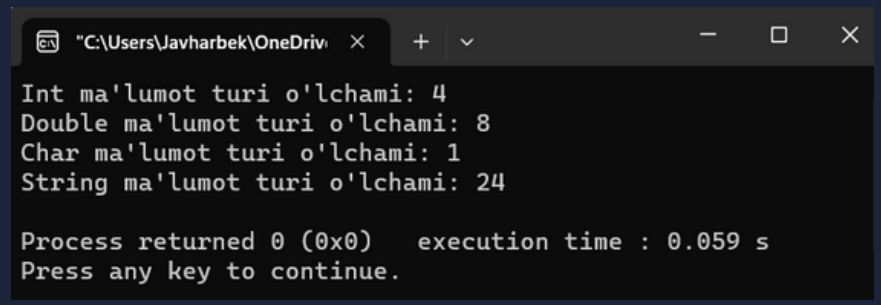
```
1-tasodifiy son: 95
2-tasodifiy son: 1
3-tasodifiy son: 38
4-tasodifiy son: 21
5-tasodifiy son: 0
```

TENX

SIZE OF ARRAY
MASSIV O'LCHAMI

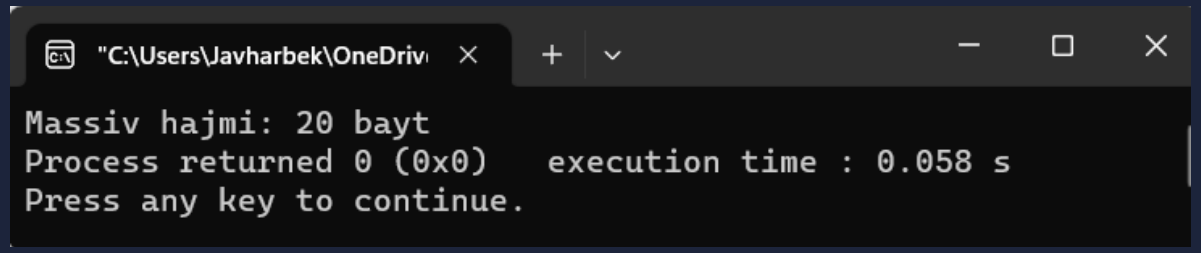
Massiv o'chamini aniqlashda sizeof() operatoridan foydalanamiz. Bu operator o'zgaruvchining ma'lumot turi o'lchamini baytdagi qiymatini qaytaradi

```
cout << "Int ma'lumot turi o'lchami: " << sizeof(int) << endl;

cout << "Double ma'lumot turi o'lchami: " << sizeof(double) << endl;

cout << "Char ma'lumot turi o'lchami: " << sizeof(char) << endl;

cout << "String ma'lumot turi o'lchami: " << sizeof(string) << endl;
```



TENX

```cpp
int nums[5] = {1, 2, 3, 4, 5};

cout << "Massiv hajmi: " << sizeof(nums) << " bayt";
```



Nima uchun 20 bayt ?

Sababi shundaki, integer hajim 4 bayt, bizning massiv o'lchami esa 5 ta integerdan tashkil topgan:

4 x 5 (4 bayt x 5 element) = 20 bayt

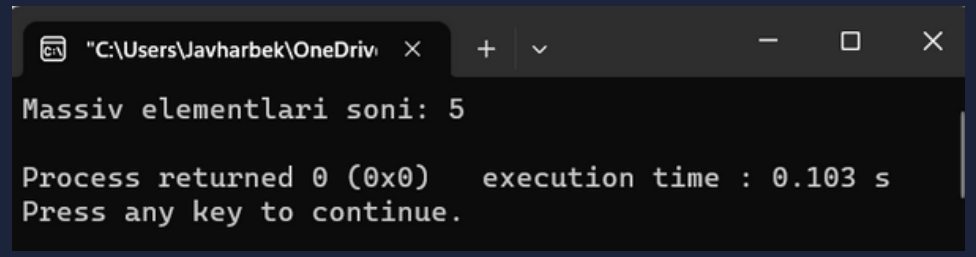
TENX

MASSIV ELEMENTLARI SONI

Massivning nechta elementi borligini bilish uchun massiv hajmini undagi ma’lumotlar turi hajmiga bo‘lish kerak.

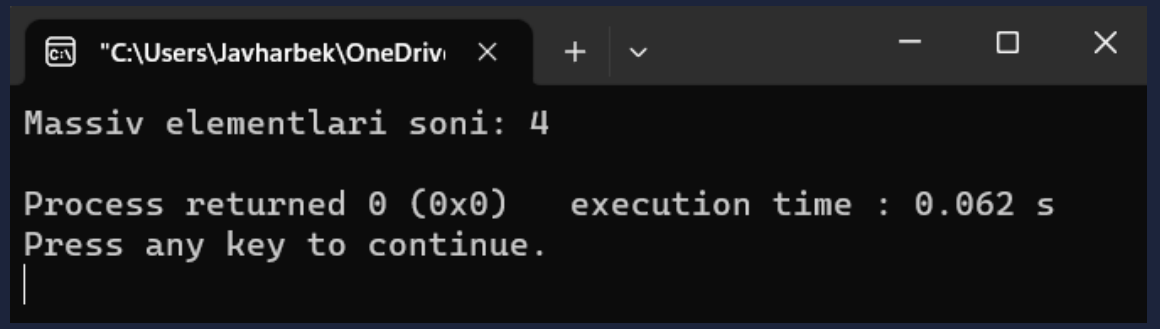
```
sizeof(array) / sizeof(dataType)
```

```
int nums[5] = {1, 2, 3, 4, 5};

int arrayLength = sizeof(nums) / sizeof(int);

cout << "Massiv elementlari soni: " << arrayLength << endl;
```

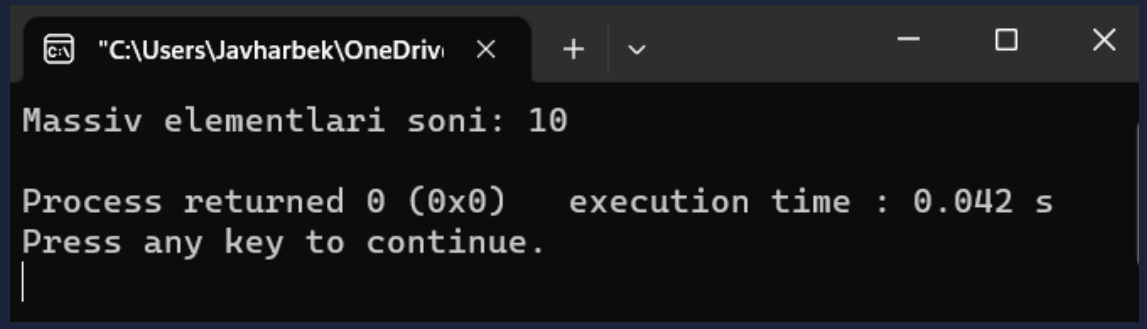


TENX

```cpp
string names[4] = {"Abubakr", "Umar", "Usmon", "Ali"};

int arrayLength = sizeof(names) / sizeof(string);

cout << "Massiv elementlari soni: " << arrayLength << endl;
```




```cpp
char chars[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j'};

int arrayLength = sizeof(chars) / sizeof(char);

cout << "Massiv elementlari soni: " << arrayLength << endl;
```



TOPSHIRIQLAR

1. [5, 2, 1, 3, 9, 6] ushbu massiv yeg'indisini toping.
2. [435, 1232, 8734, -123, 0, 12] ushbu massiv ichidagi eng katta son va u turgan indexni toping.
3. Foydananuvchi tomonidan birma bir sonlar kirtib boriladi. Kiritlgan sonlarni dastlab massivga saqlab, so'ngra ekranga chiqaring.
4. ['olma', 'anor', 'behi', 'anjir'] ushbu massiv elementlari joyini testkari tartibda almashtirish orqali yangi massiv yarating. Natija: ['anjir', 'behi, 'anor', 'olma']

E'TIBORINGIZ UCHUN RAHMAT