

### 1. Передача масиву функції. Передається адреса масиву.

```
#include <iostream>
using namespace std;

int* mass(int *a)
{
    for(int i=0; i<10;i++)
    {
        a[i] = rand() % 100;
    }
    return a;
}

int main()
{
    int b[10];
    mass(b);

    for (int i = 0; i < 10; i++)
    {
        cout << b[i]<<endl;
    }

    cout << endl;
    system("pause");
    return 0;
}
```



```
D:\c++ projects\Project10\x64\Debug\Project10.exe
41
67
34
0
69
24
78
58
62
64
Press any key to continue . . .
```

### 2. Попередній код має зайвий код. Головно повернення з функції вказівника. Це зайве, оскільки функція ітак працює по адресах переданих з головної програми

```
#include <iostream>
using namespace std;

void mass(int *a)
{
    for(int i=0; i<10;i++)
    {
        a[i] = rand() % 100;
    }
}

int main()
{
    int b[10];
    mass(b);

    for (int i = 0; i < 10; i++)
    {
        cout << b[i]<<endl;
    }

    cout << endl;
    system("pause");
    return 0;
}
```



```
D:\c++ projects\Project10\x64\Debug\Project10.exe
41
67
34
0
69
24
78
58
62
64
Press any key to continue . . .
```

### 3. Недолік коду вище, можна віднести те, що слід проаналізувати весь код, щоб зрозуміти, що в якості аргумента функції передається вказівник саме на масив. Під вказівником може бути і одиничне значення, хоча синтаксично, код цілком правильний. Тому стандарт дозволяє інший механізм передачі масиву функції, і це відразу дає розуміння для програміста, що передається масив.

Замість `void mass(int *a)` прописати `void mass(int a[])`.

4. Повернення масивів з функцій доцільне, коли ці масиви створює сама функція. В якості аргумента можна передавати розмірність масиву.

В коді виводиться сміття. Оскільки доступ до створеного масиву втрачається після виходу з функції.

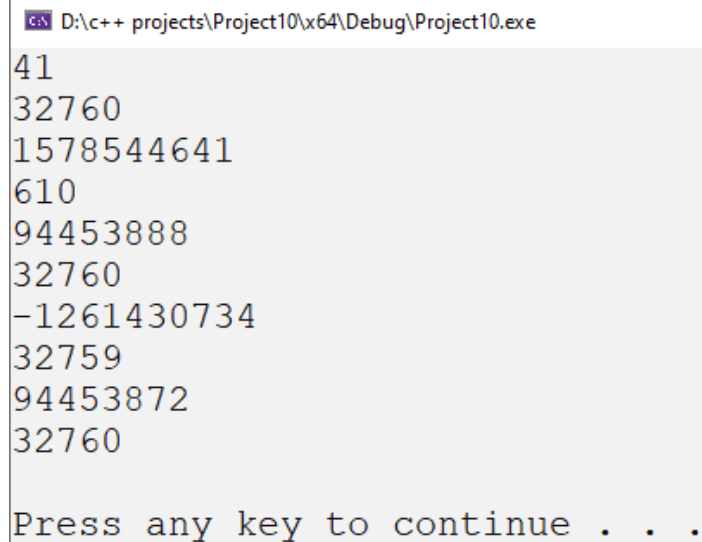
```
#include <iostream>
using namespace std;

int* mass(int a)
{
    int A[20];
    for(int i=0; i<a;i++)
    {
        A[i] = rand() % 100;
    }
    return A;
}

int main()
{
    int *b=mass(10);

    for (int i = 0; i < 10; i++)
    {
        cout << b[i]<<endl;
    }

    cout << endl;
    system("pause");
    return 0;
}
```



D:\c++ projects\Project10\x64\Debug\Project10.exe

41  
32760  
1578544641  
610  
94453888  
32760  
-1261430734  
32759  
94453872  
32760  
Press any key to continue . . .

5. Виправлення ситуації – динамічний масив. І не забути очистити пам'ять.

```
#include <iostream>
using namespace std;

int* mass(int a)
{
    int *A = new int[a];
    for(int i=0; i<a;i++)
    {
        A[i] = rand() % 100;
    }
    return A;
}

int main()
{
    int *b=mass(10);

    for (int i = 0; i < 10; i++)
    {
        cout << b[i]<<endl;
    }

    cout << endl;
    system("pause");
    delete[] b;
    return 0;
}
```



D:\c++ projects\Project10\x64\Debug\Project10.exe

41  
67  
34  
0  
69  
24  
78  
58  
62  
64  
Press any key to continue . . .