

Тема 5: Одномерные массивы

Задание 1

1) Что будет выведено?

```
int x[] = { 1, 2, 3, 4, 5 };  
std::cout << (x + 1) << '\n';
```

- 2
- **адрес в памяти (0x...)**
- 1
- ошибка времени выполнения

2) Что будет выведено?

```
int p[] = { 5, 4, 3, 2, 1 };  
std::cout << *(p + 1) << '\n';
```

- 5
 - 6
 - адрес в памяти (0x...)
 - **нет правильного ответа**
- 4

3) Что будет выведено?

```
int x[] = { 5, 4, 3, 2, 1 };  
std::cout << *x + 1 << '\n';
```

- 4
 - 5
 - **нет правильного ответа**
- 6

4) Что будет выведено?

```
int x[] = { 5, 4, 3, 2, 1 };  
std::cout << *(x + 2) + 1 << '\n';
```

- 8
 - 6
 - **нет правильного ответа**
- 4

5) Что будет выведено в результате выполнения программы?

```
#include <iostream>

int fTest (int a);

int a = 10;

int main ()
{
    const int K = 5;
    int n[K] = {5,4,3};
    int x = 30;
    int y = 40;
    int j = 0;

    for (int j = 1; j <= K; ++j){
        std::cout << j << ": " << n[j]<< std::endl;
    }

    std::cout << "j=" << j << std::endl;
    std::cout << "a=" << a << std::endl;

    std::cout << "before fTest:" << " x=" << x <<" y=" << y << std::endl;
    int otv = fTest (y-x);
    std::cout << "after fTest:" << " x=" << x <<" y=" << y << " otv=" <<
        otv << "a=" << a << std::endl;

    int c = *(n+1);
    int d = *n+2;
    std::cout << "c=" << c << " d=" << d<< std::endl;

    return 0;
}

int fTest (int a)
{
    int x = 20;
    int res = a - x;
    a += 2;
    std::cout << "in fTest: " << "a="<< a << " x="<< x << " res="<<
        res << std::endl;
    return res - 3;
}
```

1:4
2:3
3:0
4:0
5: случайное число
j:0
a = 10
before test^ x = 30 y = 40
in test a = 12 x=20 res =- 10
after test : x =30 y =40 otv=-13 a =10
c = 4 d=7

Задание 2. Поточный ввод – вывод. Разберите пример

```
#include <iostream>
#include <fstream> // для чтения из файла и записи в файл
#include <string>

const std::string ERROR_FILE_NOT_OPEN = " Не открыт";
const std::string ERROR_FILE_NOT_NUMBER = " Должно быть целое число";
const std::string ERROR_FILE = "Ошибка: файл ";

int main()
{
```

```

setlocale(LC_ALL, "russian");

std::ifstream in;    // поток для ввода
std::ofstream out;   // поток для вывода

std::string inFileName = "";
std::string outFileName = "";

std::cout << "Введите имя входного файла: ";
std::cin >> inFileName; // не проверяется корректность ввода имени

std::cout << "Введите имя выходного файла: ";
std::cin >> outFileName; // не проверяется корректность ввода имени

in.open(inFileName);
if (!in) {
    std::cerr<< inFileName + ERROR_FILE_NOT_OPEN;
    return -1;
}

int x = 0;
in >> x;    // ввод
if (!in) {
    std::cerr<< inFileName + ERROR_FILE_NOT_NUMBER;
    return -1;
}

int y = 0;
in >> y;
if (!in) {
    std::cerr<< inFileName + ERROR_FILE_NOT_NUMBER;
    return -1;
}

out.open(outFileName); // открытие потока для вывода
if (!out) {
    std::cerr << outFileName + ERROR_FILE_NOT_OPEN;
    return -1;
}

out << "x + y = " << " " << x + y;

in.close(); // завершение работы с потоком
out.close();// завершение работы с потоком
return 0;
}

```

Задание 3. Напишите функции и выполните их тестирование

- 1) Напишите функцию `inArray`, имеющую два параметра – адрес одномерного массива (целые числа) и количество элементов, выполняющую ввод всех элементов массива из потока `cin`. Функция должна инициировать исключение при обнаружении ошибки ввода.
- 2) Напишите функцию `outArray`, имеющую два параметра – адрес одномерного массива (целые числа) и количество элементов, выполняющую вывод всех элементов массива в поток `cout`.

- 3) Напишите функцию `isOrderedArray`, имеющую два параметра – адрес одномерного массива (целые числа) и количество элементов, возвращающую значение `true`, если элементы массива упорядочены по возрастанию, и `false` —в противном случае.
- 4) Напишите функцию `permutationArray`, имеющую два параметра – адрес одномерного массива (целые числа) и количество элементов, выполняющую перестановку соседних элементов массива (0-й элемент поменять с 1-м, 2-й с 3-м и т. д. Если элементов нечетное число, то последний элемент остается на своем месте).
- 5) Напишите функцию `main` для тестирования функций, в которой
- опишите массивы;
 - введите число элементов для массива в динамической памяти и выделите память для массива, предусмотрите обработку исключений, если количество элементов задано некорректно и если не память выделена;
 - вызовите функции `inArray`, `outArray` и `isOrderedArray`, `permutationArray` для встроенных массивов и массивов, расположенных в динамической памяти