

C++

Урок 5

A hand-drawn blue oval frame with a double-line border, centered on the page. The text "Разбор ДЗ" is written inside this frame in a bold, black, sans-serif font.

Разбор ДЗ

Разбор дз

1 задача:

На вход подается число, вывести да, если оно палиндром. (Решать через одномерный массив и сравнивать левую и правую часть через индексы.)

Число занести в массив через цикл while, по одной цифре.

Разбор дз

Алгоритм:

- Выделить память под одномерный массив и введенное число и нужные нам переменные.
- Узнать количество разрядов в числе и занести каждый разряд к своей ячейке.
- В цикле делать проверку первого и последнего разряда числа и тд.

Разбор дз

Листинг:

```
int arr[100] = {};  
long long int x, n, cnt, tmp;  
cin >> x; //получаем пользовательское значение  
cnt = 0;   //количество разрядов вводимого числа  
tmp = x;  
           //временная переменная для расчетов количества разрядов  
int i = 0; //для работы с массивом внутри while
```

Разбор дз

ЛИСТИНГ:

```
//считаем количество цифр и записываем цифры в массив
while (tmp != 0)
{
    arr[i] = tmp % 10; //записываем в массив в обратном порядке
    tmp /= 10;
    i++;
    cnt++;
}
```

Разбор дз

ЛИСТИНГ:

```
bool flag = 0; //для проверки числа
for (int i = 0; i < cnt; i++)
{
    if (arr[i] != arr[cnt - 1 - i])
    {
        flag = 1;
    }
}

if (flag)
    cout << "Not a Palindrom";
else
    cout << "Palindrom";
```

Разбор дз

ЛИСТИНГ:

```
bool flag = 0; //для проверки числа
for (int i = 0; i < cnt; i++)
{
    if (arr[i] != arr[cnt - 1 - i])
    {
        flag = 1;
    }
}

if (flag)
    cout << "Not a Palindrom";
else
    cout << "Palindrom";
```

Можно ли чуть оптимизировать?

Разбор дз

ЛИСТИНГ:

```
bool flag = 0; //для проверки числа
for (int i = 0; i < cnt / 2; i++)
{
    if (arr[i] != arr[cnt - 1 - i])
    {
        flag = 1;
    }
}

if (flag)
    cout << "Not a Palindrom";
else
    cout << "Palindrom";
```

Задача

Пользователь вводит размер массива n и вводит значения. Затем пользователь вводит размер второго массива и вводит в него значения.

Задача: Вставьте значения из массива 2 на позицию индекса k первого.

Алгоритм:

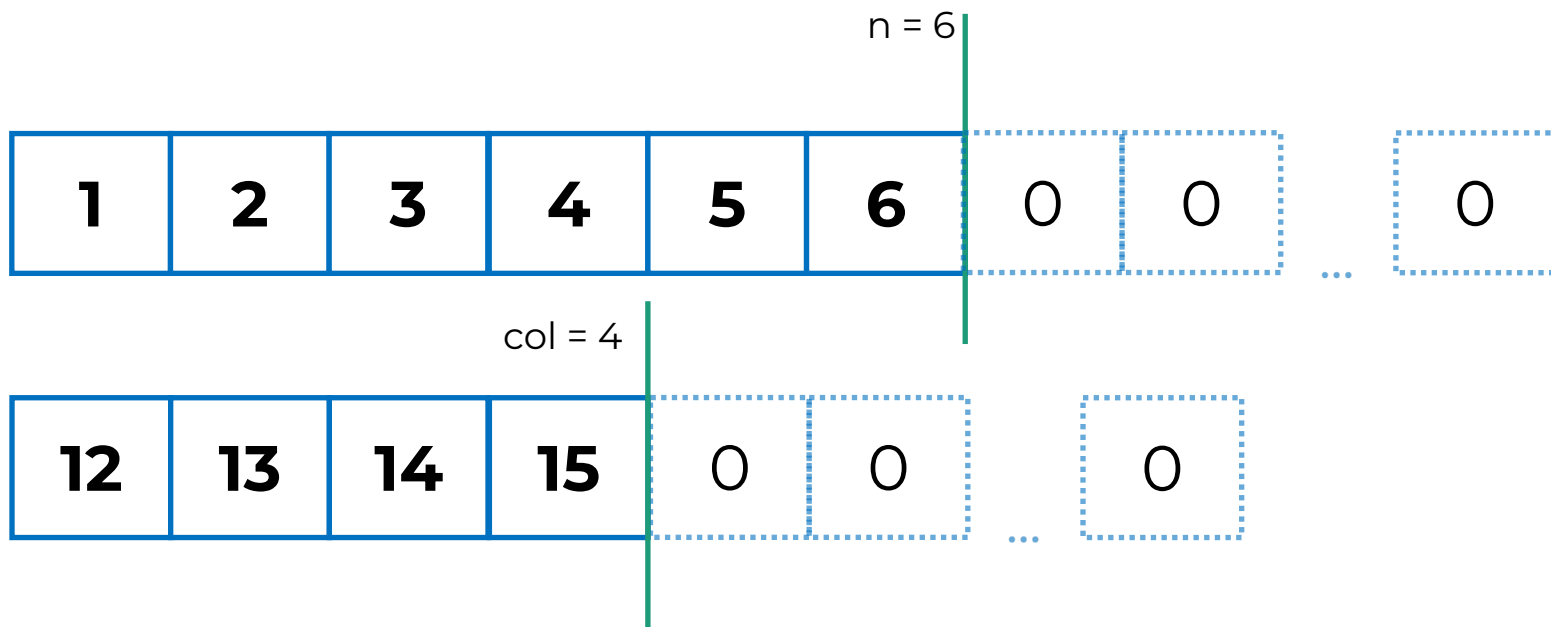
- Выделить память под 2 одномерных массива, их размерности и число (индекс вставки).
- Ввести 2 массива, индекс вставки.
- Реализовать вставку.
- Вывести получившийся массив.

Листинг:

```
int arr[100] = {1, 2, 3, 4, 5, 6}; //изначальный массив
int arr2[100] = {12, 13, 14, 15}; //массив вставки
int n = 6; //размер первого
int pos = 3; //индекс вставки
int col = 4; //количество символов вставки (размер второго массива)
```

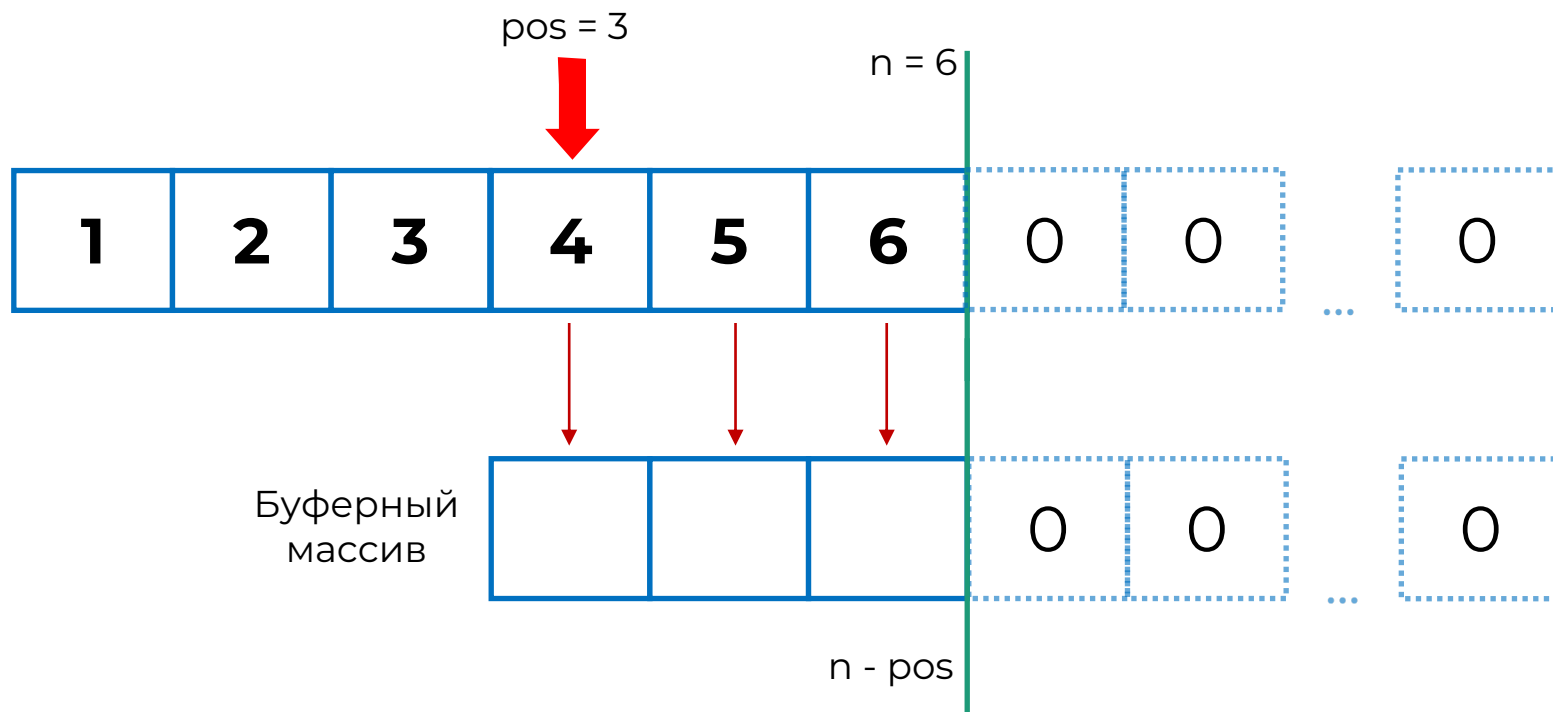
Объявление + инициализация

1 вариант:



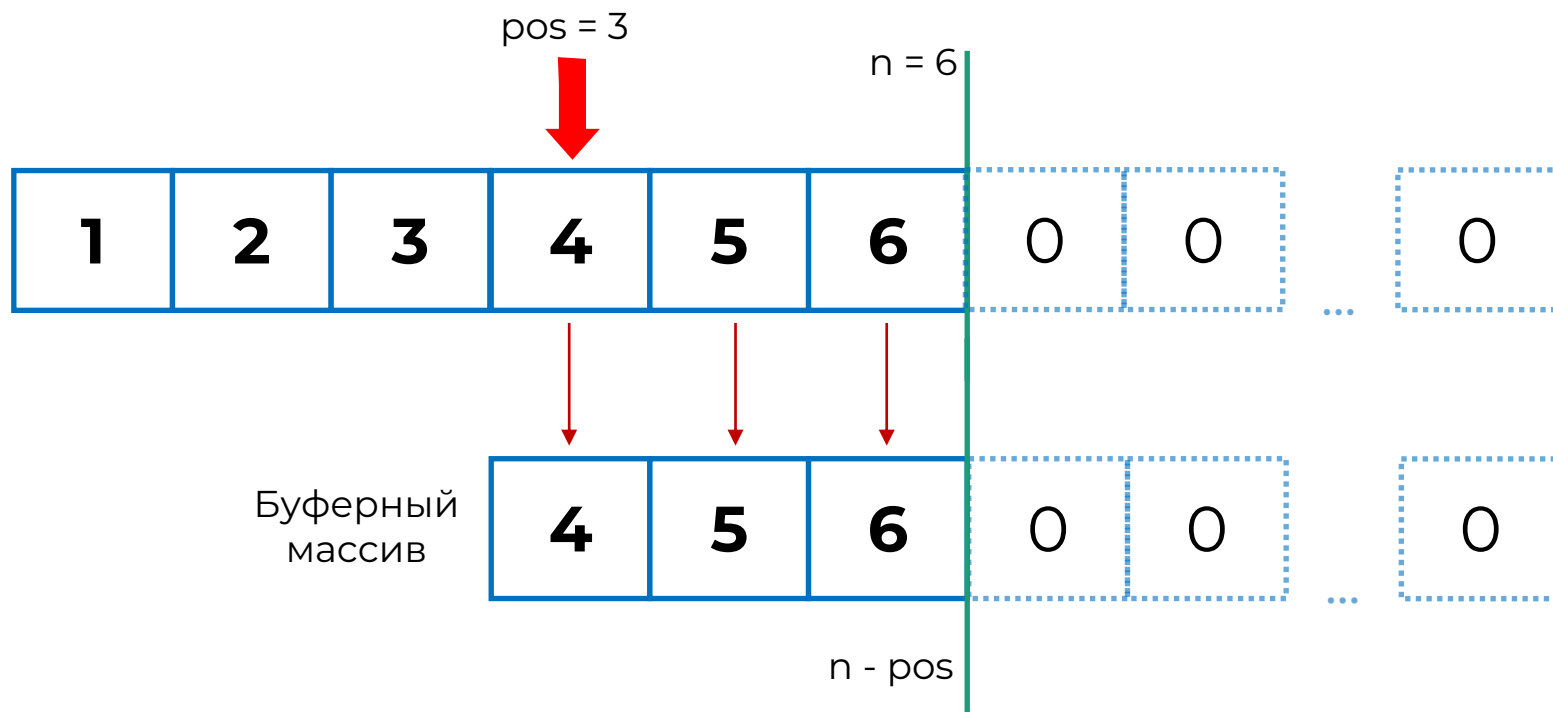
Исходные данные

1 вариант:



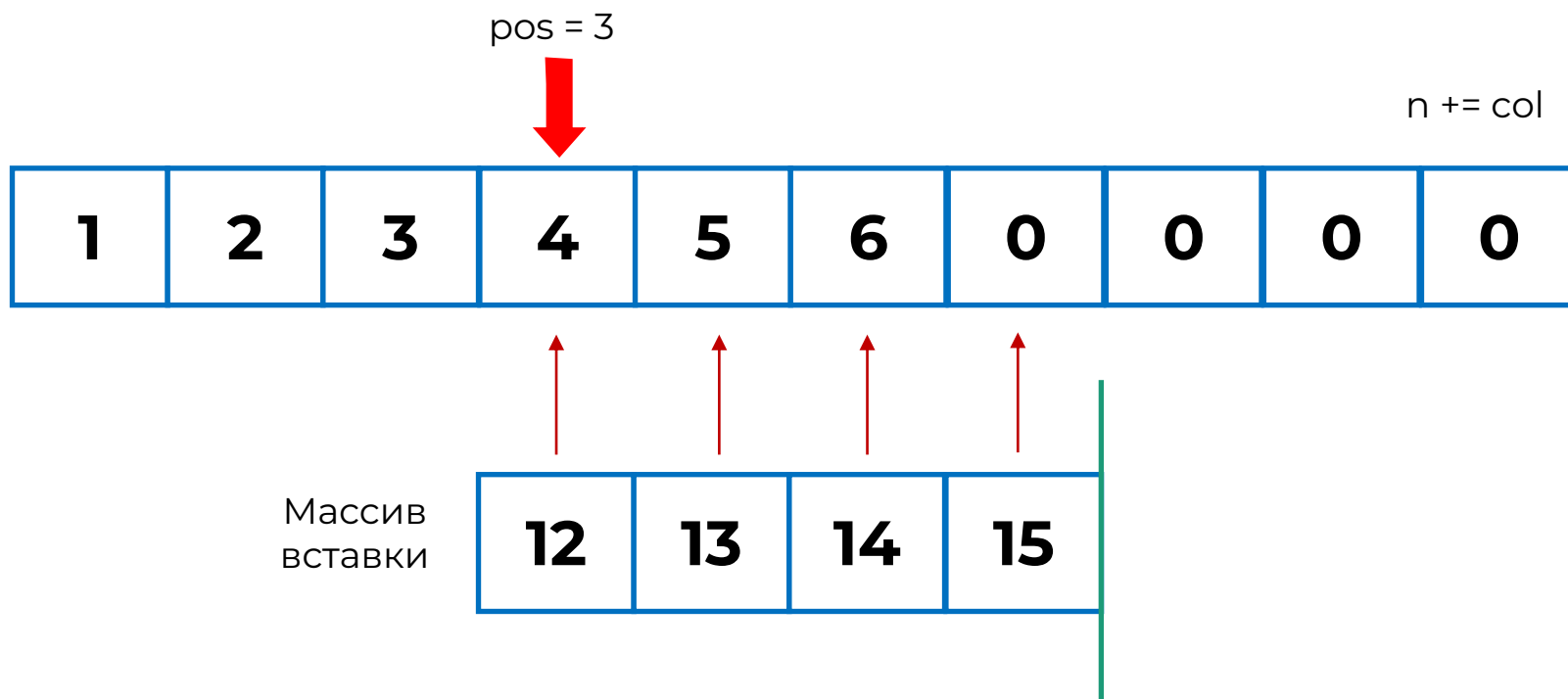
Копирование значение с позиции вставки

1 вариант:



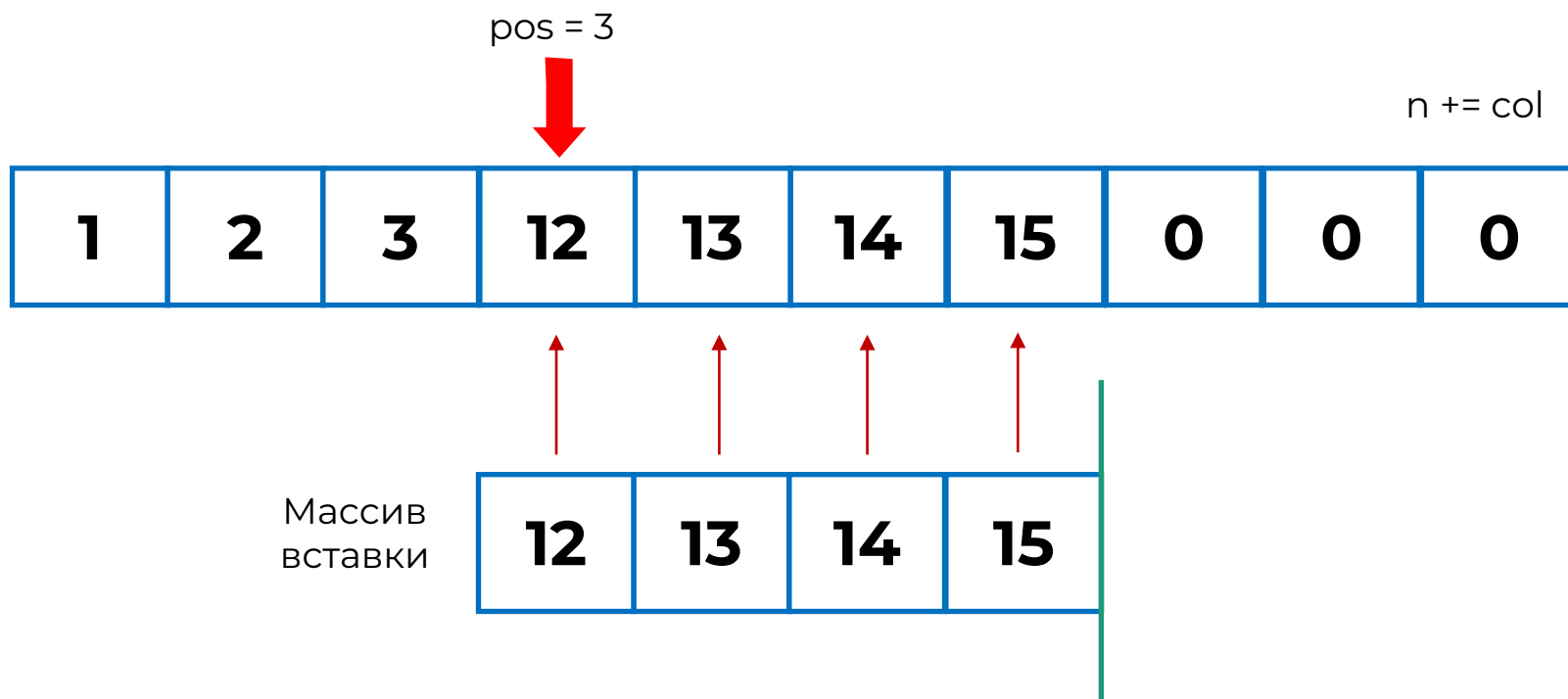
Копирование значение с позиции вставки

1 вариант:



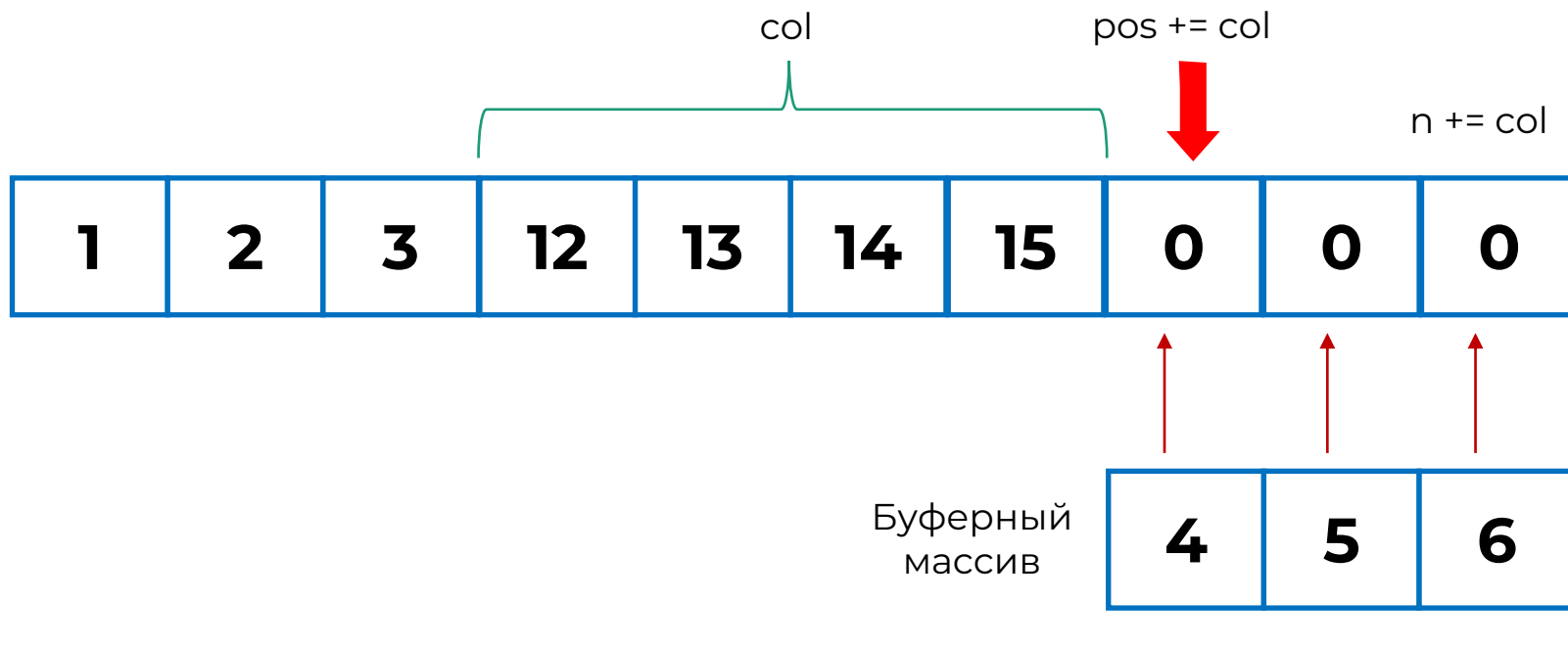
Вставка значений в исходный массив

1 вариант:



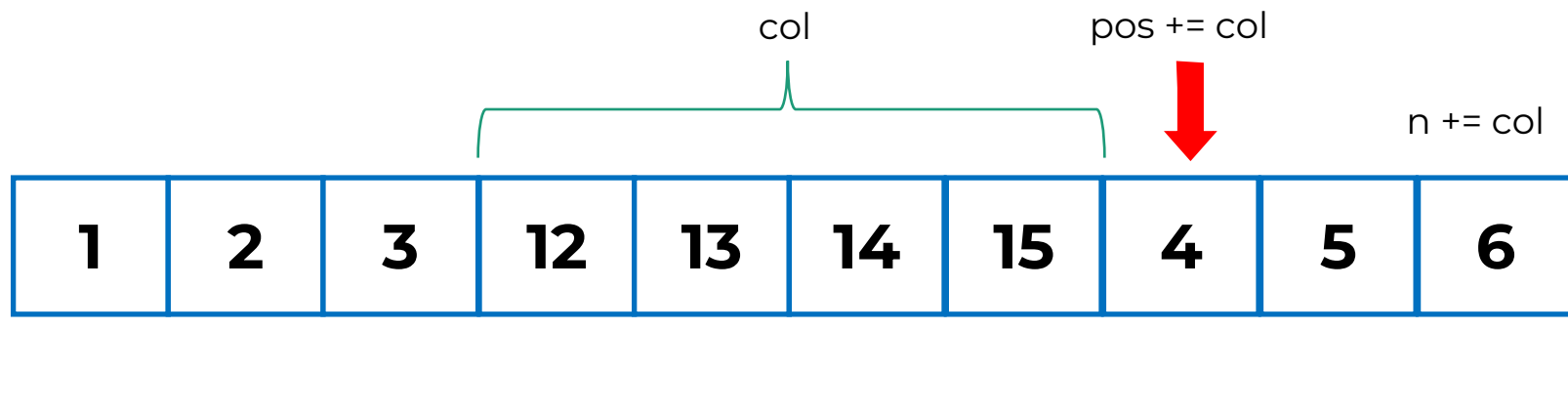
Вставка значений в исходный массив

1 вариант:



Вставка значений в исходный массив
из буфера

1 вариант:



Вывод получившегося массива

Листинг 1 вариант:

```
int arr3[100] = {};  
for (int i = pos; i < n; i++)  
{  
    arr3[i - pos] = arr[i];  
    //запись в новый массив всех значений после данного индекса  
}  
  
n += col;  
    //увеличиваем размер 1 массива на кол-во эл – тов вставки  
  
for (int i = pos; i < n; i++)  
{  
    arr[i] = arr2[i - pos]; //вставка значений из массива 2  
}  
pos += col; //увеличиваем позицию для вставки из массива 3  
  
for (int i = pos; i < n; i++)  
{  
    arr[i] = arr3[i - pos]; //запись в исходный массив  
}
```

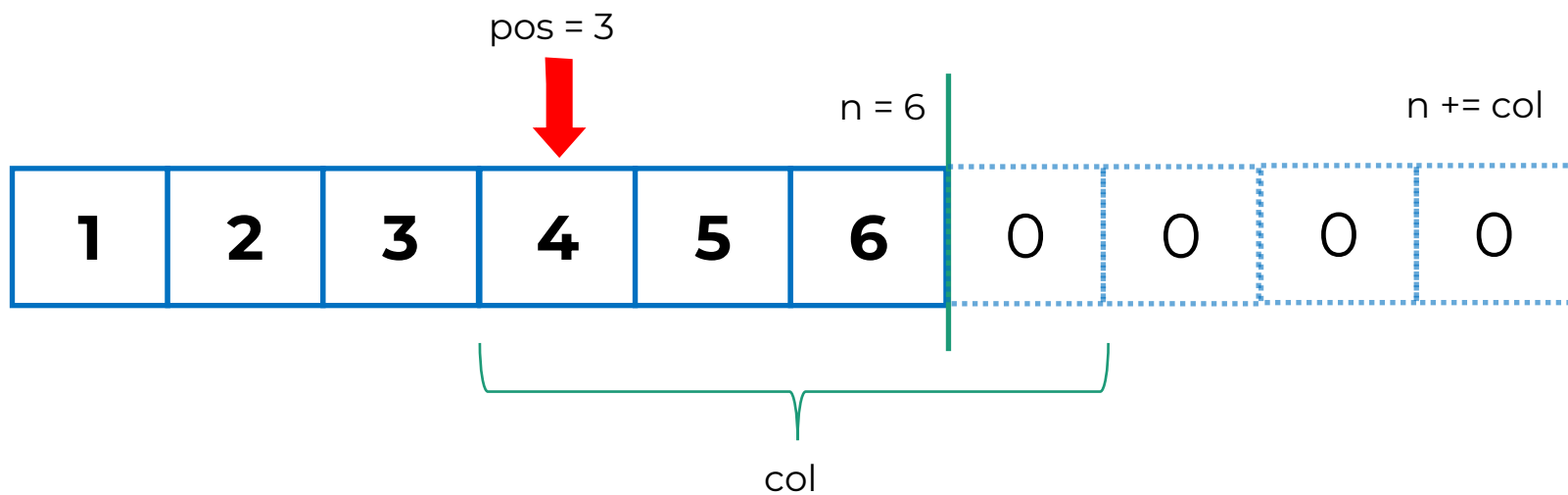
Через дополнительный массив

**Можно ли обойтись без
буферного массива?**



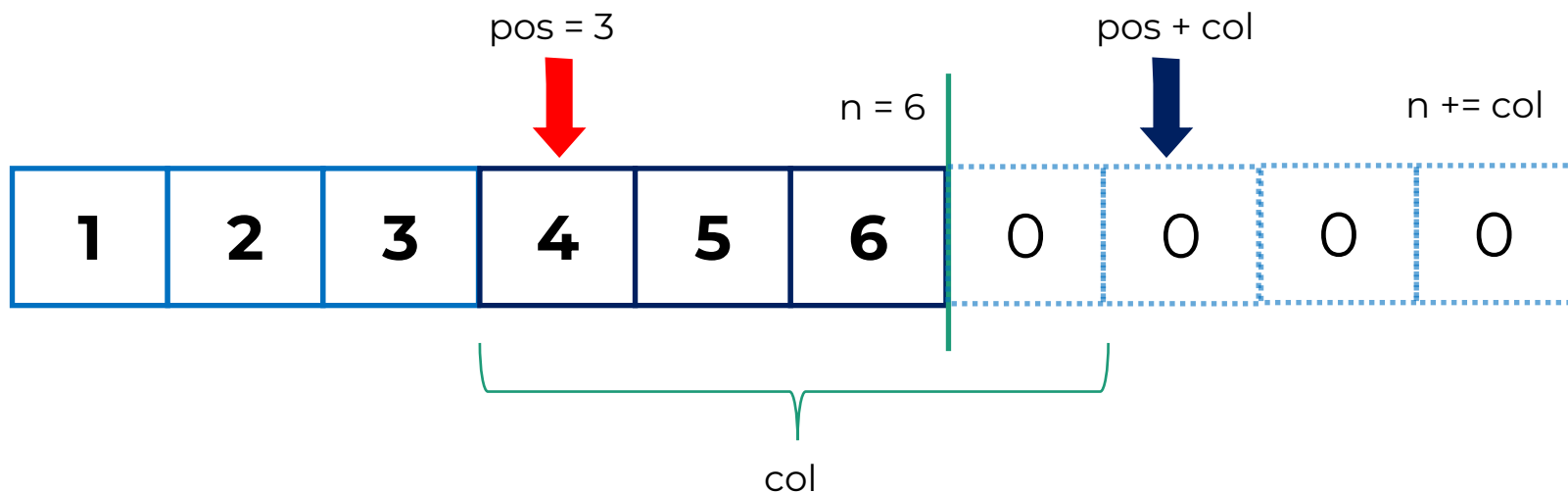
Если да, то как?

2 вариант:



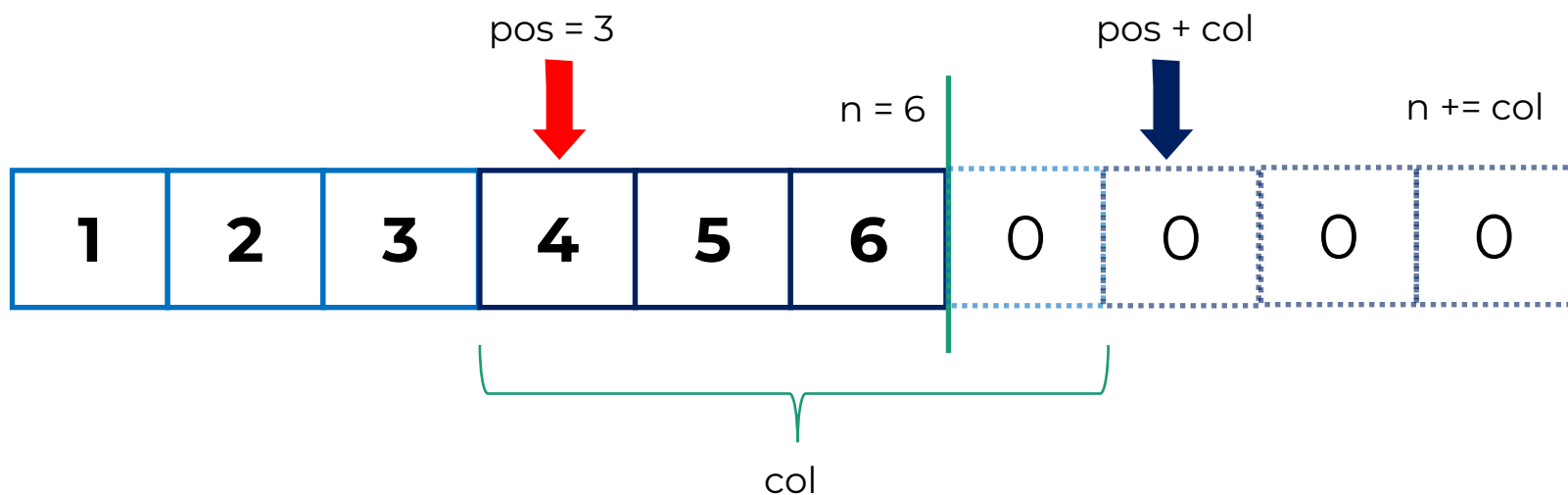
Копия значений с pos на pos + col

2 вариант:



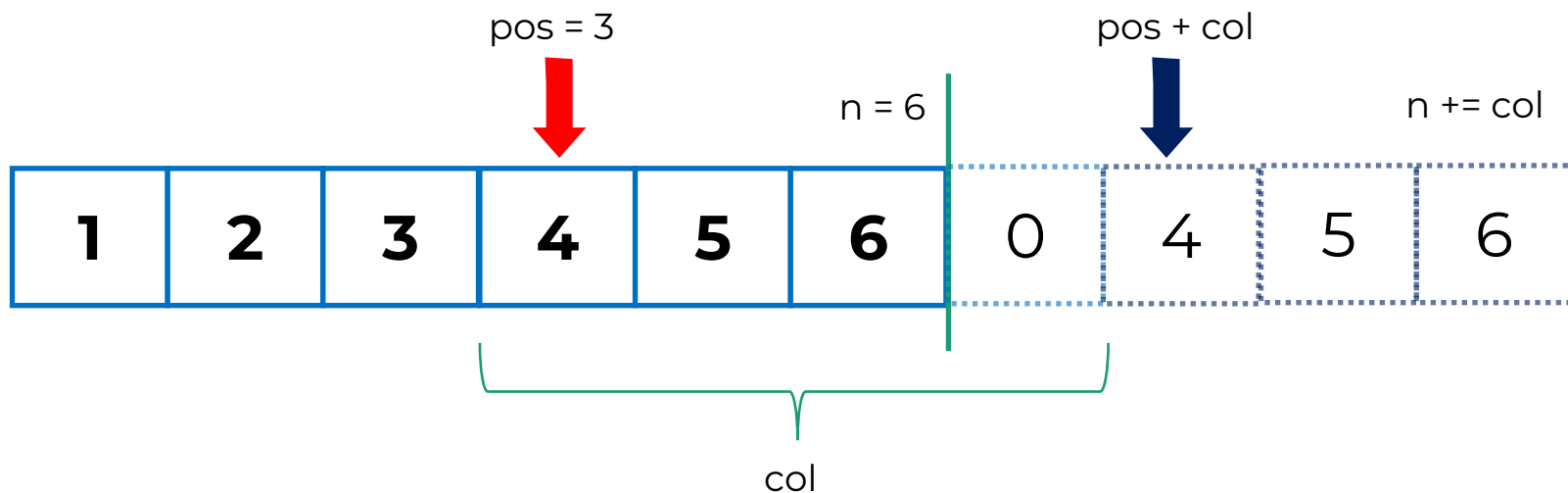
Копия значений с pos на pos + col

2 вариант:



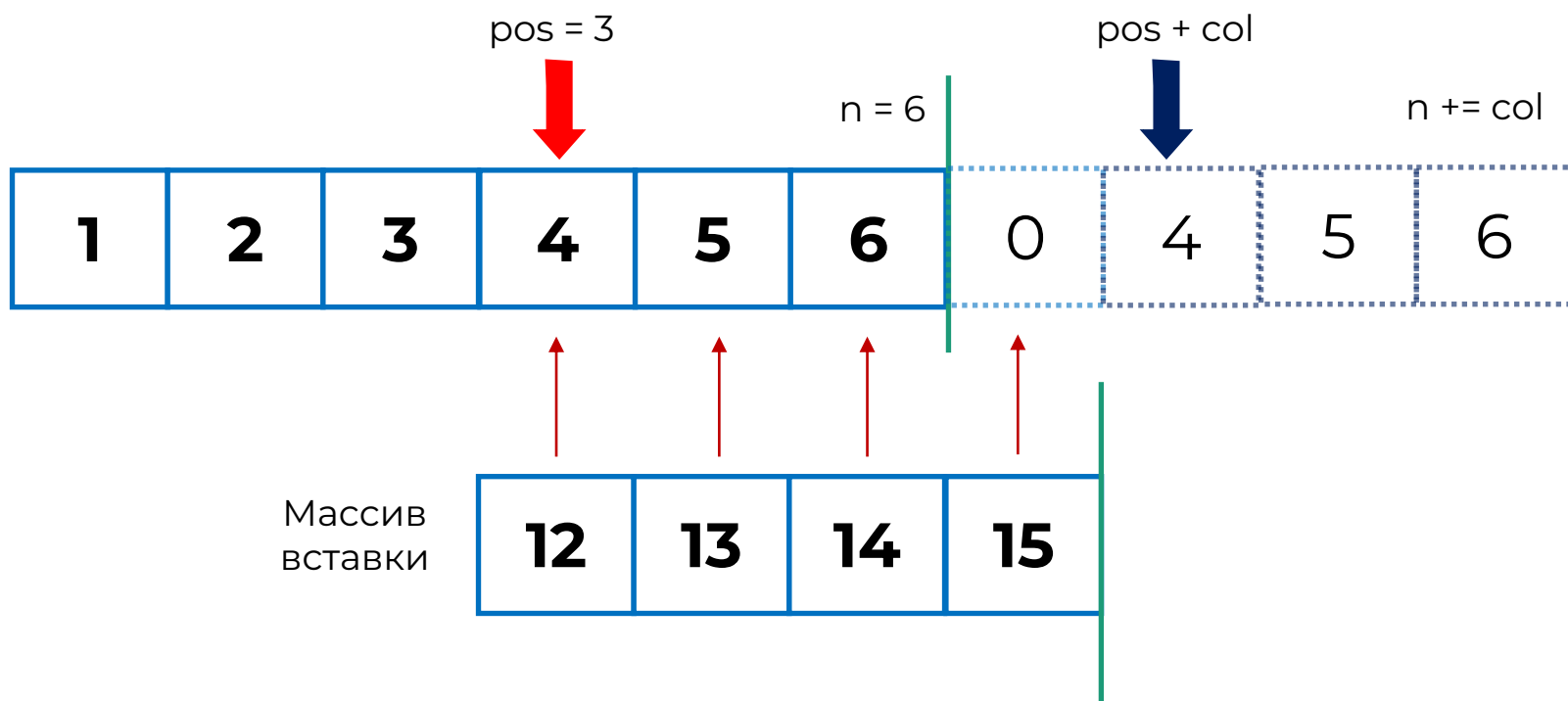
Копия значений с pos на pos + col

2 вариант:



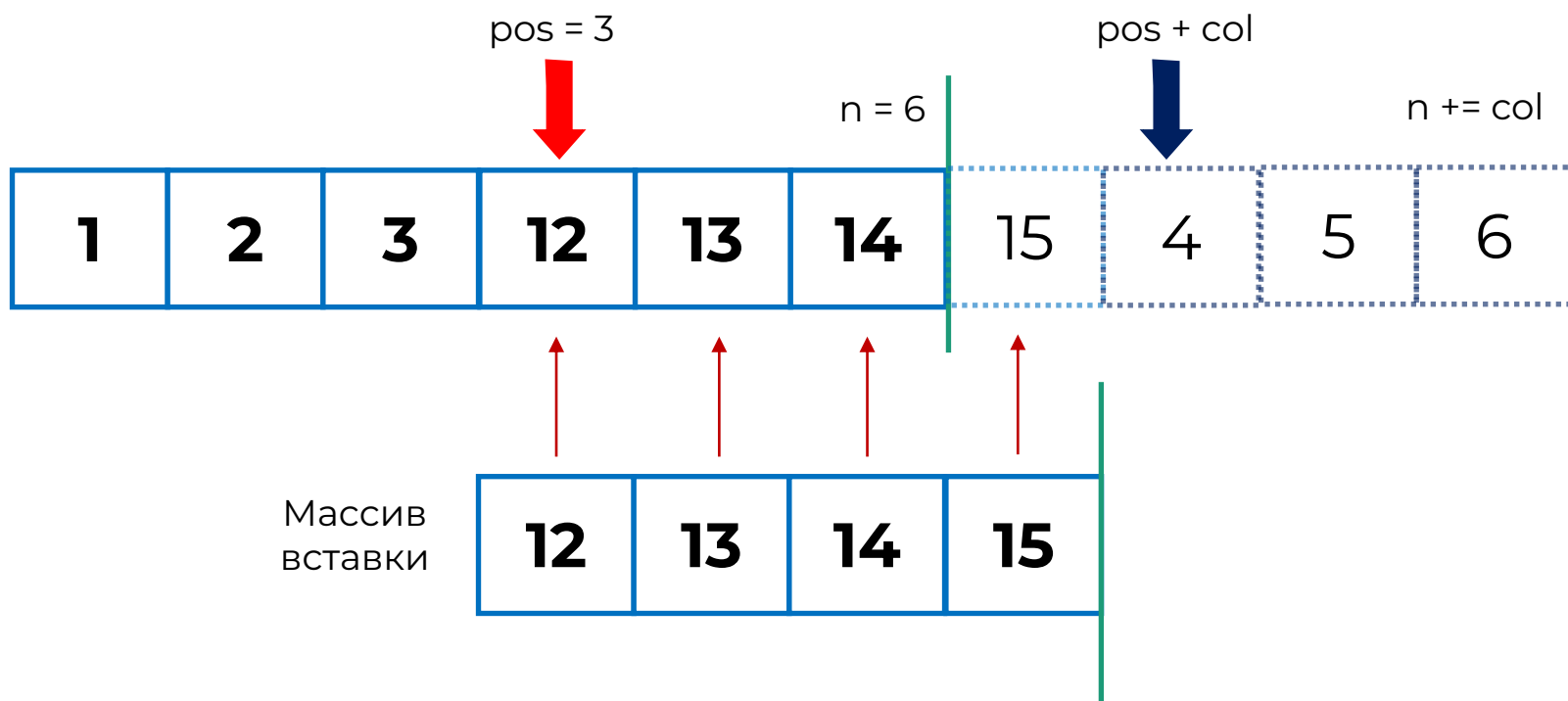
Копия значений с pos на pos + col

2 вариант:



Вставка значений в исходный массив

2 вариант:



Вставка значений в исходный массив

2 вариант:

n += col									
1	2	3	12	13	14	15	4	5	6

Вывод получившегося массива

Листинг 2 вариант:

```
n+= col;
for (int i = n; i >= pos + col; i--){
    arr[i] = arr[i - col]; //делаем копию
}

for (int i = 0; i < col; i++)
{
    arr[i + pos] = arr2[i]; //реализуем вставку
}
```

Без дополнительного массива

A hand-drawn blue oval frame with a double-line border, centered on the page. The word "Повторение" is written inside this frame in a bold, black, sans-serif font.

Повторение

Что такое массив?

На что похож?

Какие виды массивов узнаете?

**Как лучше инициализировать
одномерный массив?**

A hand-drawn blue oval frame with a slightly irregular, sketchy border, centered on the page. It contains the word 'Сортировка' in bold black text.

Сортировка



Что такое сортировка?

Сортировка

Сортировка – это алгоритм для упорядочивания данных по некоторым признакам.



Где применяется сортировка?

Сортировка

Для того, чтобы отсортировать массив по **возрастанию**:

```
for (int i = 0; i < n - 1; i++)  
{  
    for (int j = 0; j < n - 1; j++)  
    {  
        if (arr[j] > arr[j + 1])  
            swap(arr[j], arr[j + 1]);  
    }  
}
```

Проходим в циклах до **n - 1**

Сортировка

Для того, чтобы отсортировать массив **не по возрастанию**:

```
for (int i = 0; i < n - 1; i++)  
{  
    for (int j = 0; j < n - 1; j++)  
    {  
        if (arr[j] < arr[j + 1])  
            swap(arr[j], arr[j + 1]);  
    }  
}
```

Проходим в циклах до **n - 1**

Почему до $n - 1$?



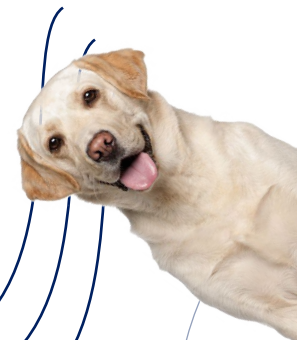
Можно ли оптимизировать?

Сортировка

Для того, чтобы отсортировать массив по **возрастанию**:

```
for (int i = 0; i < n - 1; i++)  
{  
    for (int j = 0; j < n - i - 1; j++)  
    {  
        if (arr[j] > arr[j + 1])  
            swap(arr[j], arr[j + 1]);  
    }  
}
```

Проходим во внутреннем
цикле до **$n - i - 1$**



Задача

Умпа-Лумпы решили взять с собой в лилипутию песелей. При прохождении инопланетного таможенного контроля нужно выполнять взвешивание домашних животных. (массив весов)

Задача: Отсортируйте по возрастанию веса всех песелей Умпа-Лумпов и выведите их на экран таможенника.

Задача

Пользователь вводит в консоль число - n (размер одномерного массива), сами значения и k (число).

Задача: Найти индекс числа k в в отсортированном массиве.

Генератор чисел

Генератор чисел

Листинг:

```
int num = rand(); //функция генерации ранд значения
```


Генератор чисел

Листинг:

```
int num = rand(); //функция генерации ранд значения
```

Консоль:

```
s_1/Гит плюсы/8 less/Практика/"8_01  
16807
```

- 1 запуск

```
s_1/Гит плюсы/8 less/Практика/"8_01  
16807
```

- 2 запуск

```
s_1/Гит плюсы/8 less/Практика/"8_01  
16807
```

- 3 запуск

Генератор чисел

Листинг:

```
int num = rand(); //функция генерации ранд значения
```

Консоль:

```
s_1/Гит плюсы/8 less/Практика/"8_01  
16807
```

- 1 запуск

```
s_1/Гит плюсы/8 less/Практика/"8_01  
16807
```

- 2 запуск

```
s_1/Гит плюсы/8 less/Практика/"8_01  
16807
```

- 3 запуск

Поменялось ли значение?

Генератор чисел

Листинг:

```
int num = rand(); //функция генерации ранд значения
```

Консоль:

```
s_1/Гит плюсы/8 less/Практика/"8_01  
16807
```

- 1 запуск

```
s_1/Гит плюсы/8 less/Практика/"8_01  
16807
```

- 2 запуск

```
s_1/Гит плюсы/8 less/Практика/"8_01  
16807
```

- 3 запуск

НЕТ

Генератор чисел

Для того, чтобы значения **изменялись**:

```
#include <ctime>
```

1) Подключаем библиотеку **<ctime>**

```
srand(time(0));  
int num = rand(); //функция генерации ранд значения
```

2) Прописываем srand (time(0)).

Генератор чисел

Для того, чтобы значения **изменялись**:

```
#include <ctime>
```

1) Подключаем библиотеку **<ctime>**

```
srand(time(0));  
int num = rand(); //функция генерации ранд значения
```

2) Прописываем srand (time(0)).

Консоль:

```
1319500485  
artemgudzenko@192 Практика % cd "/Users/artemgudzenko/Desktop/Less_1/Гит плюсы/8 less/Практика/"8_01  
1319534099  
artemgudzenko@192 Практика % cd "/Users/artemgudzenko/Desktop/Less_1/Гит плюсы/8 less/Практика/"8_01  
1319584520
```

**Как устранить проблему
больших чисел?**

Генератор чисел

Ограничиваем вывод **остатком от деления**.

```
srand(time(0));  
int num = rand() % 100; //только двузначные
```

Консоль:

```
15  
artemgudzenk  
&& "/Users/  
36  
artemgudzenk  
&& "/Users/  
57
```

**Как добавить отрицательные
числа?**

Генератор чисел

Положительные и отрицательные числа:

```
int num = -45 + rand() % 100; //двузначные (полож и отриц числа)
cout << num << endl;
```

Консоль:

```
-35
artemgudzenko@19
&& "/Users/arte
-14
artemgudzenko@19
&& "/Users/arte
0
```

Генератор чисел

Альтернативный способ:

```
srand(time(0)); //для обновления результатов  
int num = -34 + (rand() % static_cast<int>(100 - -34 + 1));
```

Задача

Пользователь вводит в консоль число - n (размер одномерного массива).

Задача: Заполните массив случайными значениями в диапазоне $[-120; 340]$.

Двумерные массивы

**В виде чего можно представить
двумерные массивы**

Двумерный массив

Двумерный массив – это массив, в котором каждый элемент является массивом.



Двумерный массив

Двумерный массив – это массив, в котором каждый элемент является массивом.



Двумерный массив

Двумерный массив – это массив, в котором каждый элемент является массивом.



Двумерный массив

Объявление и инициализация:

```
int arr[100][100]; //выдел под него память;  
  
int arr1 [3][7] =  
{  
  {1,2,3,4,5,6,7,}, // 1 2 3 4 5 6 7  
  {2,3,4}, //2 3 4 0 0 0 0  
  {4,5,6,7} //4 5 6 7 0 0 0  
};  
  
int arr2[][8] = { //сам определяет ко-во строк  
  {1,2,3,4,5,6,7,8},  
  {1,2,3,4,5,6,7,8}  
};
```

Двумерный массив

Ввод:

```
int n, m; // переменные для строк и столбцов
cout << "Вв кол-во строк и кол-во столб"<<endl;
cin >> n >> m;
for (int i = 0; i < n; i++){
    for (int j = 0; j < m; j++){
        cin >> arr[i][j];
    }
}
```

Двумерный массив

Ввод:

```
int n, m; // переменные для строк и столбцов
cout << "Вв кол-во строк и кол-во столб"<<endl;
cin >> n >> m;
for (int i = 0; i < n; i++){
    for (int j = 0; j < m; j++){
        cin >> arr[i][j];
    }
}
```

Как сделать **вывод**?

Двумерный массив

Вывод:

```
for (int i = 0; i < n; i++){  
    for (int j = 0; j < m; j ++){  
        cout <<arr[i][j]<<" ";  
    }  
    cout << endl;  
}
```

Задача

Пользователь вводит в консоль числа – n и m (размер двумерного массива).

Задача: Заполните массив случайными значениями и выведите на экран.

Лайфхаки у массивов

Хаки массивов

Задача: Пользователь вводит размер массива $n * n$.
Заполните в двумерном массиве эл-ты главной
диагонали значением 1 и выведите результат.

Хаки массивов

Листинг:

```
for (int i = 0; i < n; i++)  
{  
    for (int j = 0; j < n; j++)  
    {  
        if (i == j) //если на главной диагонали  
        {  
            arr[i][j] = 1; //заполняем 1  
        }  
    }  
}
```


Хаки массивов

Листинг:

```
for (int i = 0; i < n; i++)  
{  
    for (int j = 0; j < n; j++)  
    {  
        if (i == j) //если на главной диагонали  
        {  
            arr[i][j] = 1; //заполняем 1  
        }  
    }  
}
```

Можно обойтись **без if**?

Хаки массивов

Листинг:

```
for (int i = 0; i < n; i++)  
{  
    for (int j = 0; j < n; j++)  
    {  
        arr[i][i] = 1; //условие главной диагонали  
    }  
}
```

Да

Хаки массивов

Листинг:

```
for (int i = 0; i < n; i++)  
{  
    for (int j = 0; j < n; j++)  
    {  
        arr[i][i] = 1; //условие главной диагонали  
    }  
}
```

Нужно ли теперь проходить $n * n$ раз?

Хаки массивов

Листинг:

```
for (int i = 0; i < n; i++)  
{  
    for (int j = 0; j < n; j++)  
    {  
        arr[i][i] = 1; //условие главной диагонали  
    }  
}
```

Нужно ли теперь проходить **$n * n$** раз? **НЕТ**

Хаки массивов

Листинг:

```
for (int i = 0; i < n; i++)  
{  
    arr[i][i] = 1; //условие главной диагонали  
}
```

Хаки массивов

Листинг:

```
for (int i = 0; i < n; i++)  
{  
    arr[i][i] = 1; //условие главной диагонали  
}
```

Уменьшили кол-во итераций (сложность) в n раз.

Хаки массивов

Листинг:

```
for (int i = 0; i < n; i++)  
{  
    arr[i][i] = 1; //условие главной диагонали  
}
```

Для **главной** диагонали

Хаки массивов

Листинг:

```
for (int i = 0; i < n; i++)  
{  
    arr[i][n - i - 1] = 1;  
}
```

Для **побочной** диагонали

Задача

Задача: Пользователь вводит размер массива $n * n$. Массив заполняется случайными числами. Требуется найти произведение **нечетных** элементов **главной** диагонали.

Задача

Задача: Пользователь вводит размер массива $n * n$. Массив заполняется случайными числами. Требуется поменять местами эл-ты главной и побочной диагоналей.