

Массив (vector)

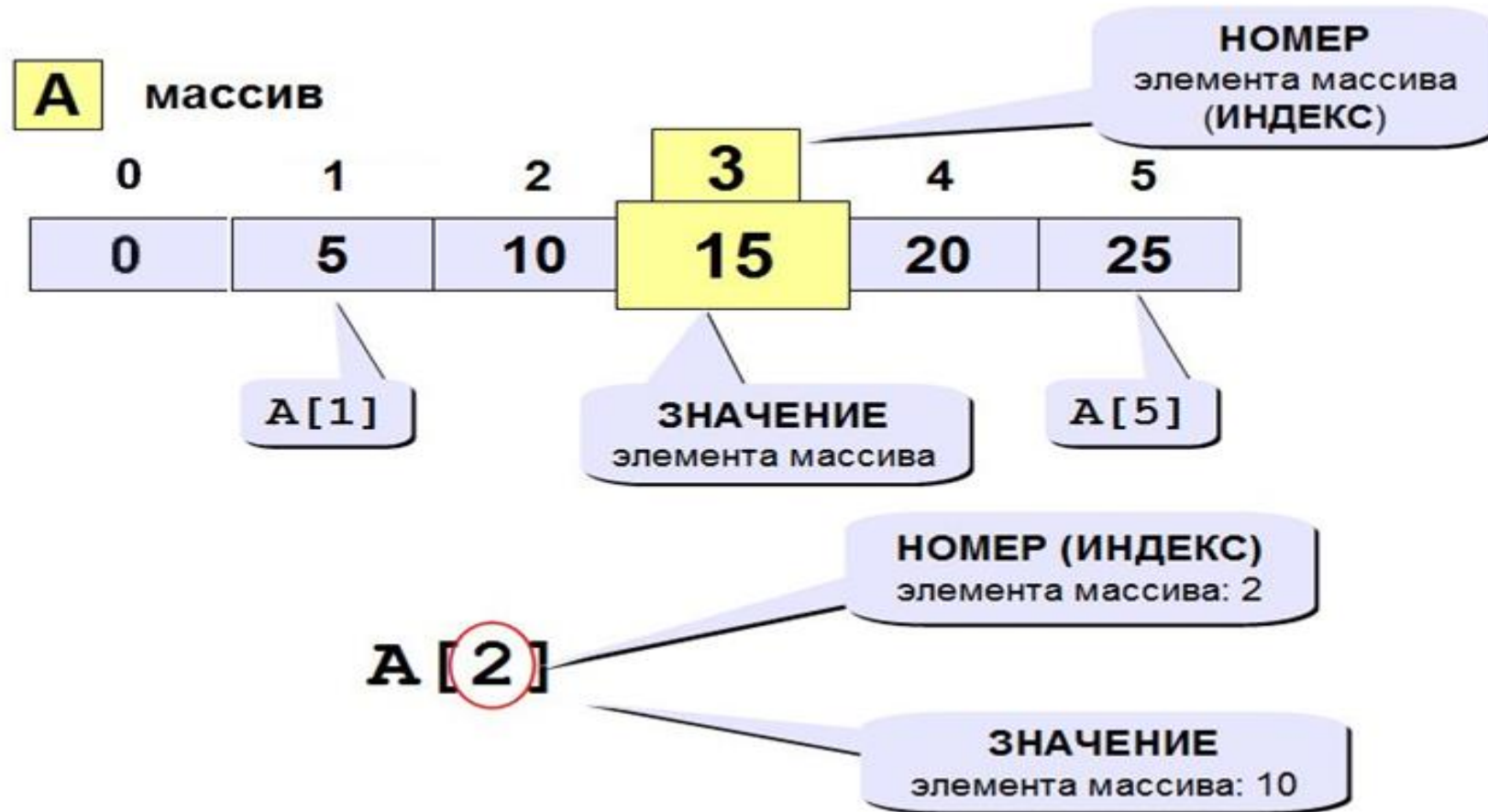
Основные функции работы с типом vector

Массив

структурированный тип данных, который состоит из фиксированного числа элементов одного и того же типа, объединенных одним именем, где каждый элемент имеет свой номер.

К каждому **элементу массива** можно обращаться по его имени и номеру (индексу).

Одномерный массив



Описание

Библиотека

#include <vector>

```
vector <int> v;
```

```
n = 10;
```

```
vector <int> v (n) ;
```

```
cin >> n;
```

```
vector <int> v (n, -1) ;
```

- Вектор целых чисел
- Вектор из n целых чисел
- Вектор из n целых чисел, равных -1.

Ввод элементов

```
n = 10;  
vector < int > v(n);  
for(int i = 0; i < n; i++)  
    cin >> v[i];
```

```
n = 10;  
vector < int > v;  
for(int i = 0; i < n; i++) {  
    cin >> x;  
    v.push_back(x);  
}
```

Ввод элементов

```
#include <cstdlib>
#include <ctime>

...

srand(time(NULL));
for (int i = 0; i < n; i++)
    a[i] = rand() % 10 - 5;
for (int i = 0; i < n; i++)
    cout << a[i] << " ";
cout << endl;
```

Вывод элементов

```
for (int i = 0; i < n; i++)  
    cout << a[i] << " ";  
cout << endl;
```


Некоторые функции класса `vector`

Функция	Действие
<code>size()</code>	возвращает размер вектора
<code>empty()</code>	возвращает <code>true</code> или <code>false</code> в зависимости от того пуст вектор или нет
<code>push_back(x)</code>	добавление элемента <code>x</code> в конец вектора (после последнего элемента, для которого выделена память)
<code>pop_back()</code>	удаление из вектора последнего элемента, не возвращая его значения:
<code>clear();</code>	удаляет все элементы вектора

Некоторые функции класса `vector`

Функция	Действие
<code>insert(v.begin()+k, x);</code>	вставляет значение <code>x</code> в позицию <code>k</code> вектора <code>v</code> , сдвигая все элементы вектора после <code>k</code> на 1 вправо
<code>insert(v.begin()+k, n, x);</code>	вставляет <code>n</code> значений равных <code>x</code> в позицию <code>k</code> вектора <code>v</code> , сдвигая все элементы вектора после <code>k</code> на <code>n</code> вправо
<code>erase(v.begin()+k);</code>	удаляет элемент с номером <code>k</code> , сдвигая все остальные на 1 влево.
<code>erase(v.begin()+k1, v.begin()+k2);</code>	удаляет элементы от номера <code>k1</code> до номера <code>k2</code> , сдвигая все остальные влево.
<code>clear();</code>	удаляет все элементы вектора

Изменение размера

- Для изменения размер вектора, используется метод `resize(n)`:
- После вызова метода `resize(n)` вектор будет содержать ровно `n` элементов.
 - Если параметр `n` меньше, чем размер вектора до вызова `resize(n)`, то вектор уменьшится и «лишние» элементы будут удалены.
 - Если же `n` больше, чем размер вектора, то вектор увеличит свой размер и заполнит появившиеся элементы нулями.
 - Если хранимым типом данных является более сложный объект, нежели стандартный тип данных новые элементы будут инициализированы конструктором по умолчанию.
- Важно помнить, что если использовать `push_back(x)` после `resize(n)`, то элементы будут добавлены после области, выделенной `resize(n)`, а не в неё.

Изменение размера

- Использование `reserve(n)`.
- Вызов метода `reserve(size_t n)` выделяет дополнительную память в будущее пользование `vector`.
- Параметр `n` имеет следующий смысл: вектор должен выделить столько памяти, чтобы вплоть до размера в `n` элементов дополнительных операций выделения памяти не потребовалось

Инициализация вектора

```
vector<int> v1;
```

```
vector<int> v2 = v1;
```

```
vector<int> v3(v1);
```

```
int a[13] = { 1, 1, 2, 3, 3, 3, 3, 4, 4, 4, 5, 5, 6 };
```

```
vector<int> data(a, a+13);
```

Начиная со стандарта C++11

```
vector<int> s={1, 2, 3};
```

Одномерный массив

Объявление:

```
int a[16];
```

где,

int - целочисленный тип данных;

a - имя одномерного массива;

16 — размер одномерного массива, 16 ячеек.

```
int mas[10], a[16];
```

Одномерный массив

- Массивы могут быть инициализированы при объявлении:

```
int a[7] = { 5, -12, -12, 9, 10, 0, -9};
```

```
int a[] = { 5, -12, -12, 9, 10, 0, -9};
```

- Размер массива можно не указывать только при его инициализации, компилятор сам определит размер массива

Ввод элементов одномерного массива

```
int n, a[100];  
cout << "vvedi n=";  
cin >> n;  
cout << "vvedi elementy" << endl;  
for (int i = 0; i < n; i++)  
    cin >> a[i];
```

Вывод элементов одномерного массива

```
for (int i = 0; i < n; i++)  
    cout << a[i] << " ";  
cout << endl;
```

Будем решать задачи на сайте <https://acmp.ru/>

курсы/язык программирования C++/массивы/линейный поиск

https://acmp.ru/asp/do/index.asp?main=topic&id_course=1&id_section=5&id_topic=113

ЗАДАЧА №1214

https://acmp.ru/asp/do/index.asp?main=task&id_course=1&id_section=5&id_topic=113&id_problem=691

Линейный поиск

(Время: 1 сек. Память: 16 Мб Сложность: 12%)

Требуется вычислить, сколько раз встречается некоторое число X в массиве $A[1..N]$.

Входные данные

Первая строка входного файла INPUT.TXT содержит натуральное число N – количество элементов в массиве. Во второй строке записаны N целых чисел A_i , разделенные пробелом. Третья строка содержит число X . Все числа во входных данных не превосходят 1000 по абсолютной величине.

Выходные данные

В выходной файл OUTPUT.TXT выведите количество вхождений числа X в массив A .

Нужно идти по элементам массива, **(for (int i=0;i<n;i++))**
и когда будем встречать X
(if (a[i] == X)
увеличиваем счетчик вхождений X на 1.

ЗАДАЧА №1216

https://acmp.ru/asp/do/index.asp?main=task&id_course=1&id_section=5&id_topic=113&id_problem=693

Максимальный элемент

(Время: 1 сек. Память: 16 Мб Сложность: 16%)

В целочисленном массиве $A[1..N]$ на отрезке $[L, R]$ необходимо найти максимальный элемент и его номер.

Входные данные

Первая строка входного файла INPUT.TXT содержит натуральное число N – количество элементов в массиве. Во второй строке записаны N целых чисел A_i , разделенные пробелом. Третья строка содержит два натуральных числа L и R , которые задают отрезок поиска $[L, R]$ ($-1000 \leq A_i \leq 1000$, $1 \leq L \leq R \leq N \leq 1000$).

Выходные данные

В выходной файл OUTPUT.TXT выведите значение максимального элемента массива на отрезке $[L, R]$ и его индекс. В случае неоднозначности определения индекса следует вывести наименьший из них.

Пример

Найдем индекс k наибольшего элемента между l и r .

Пройдемся по элементам массива от левой до правой границы, `(for (int i=l+1;i<=r;i++) )`

первонач

`(if (a[i] == X)`

увеличиваем счетчик вхождений X на 1.

ЗАДАЧА №1216

https://acmp.ru/asp/do/index.asp?main=task&id_course=1&id_section=5&id_topic=113&id_problem=693

Максимальный элемент

(Время: 1 сек. Память: 16 Мб Сложность: 16%)

В целочисленном массиве $A[1..N]$ на отрезке $[L, R]$ необходимо найти максимальный элемент и его номер.

Входные данные

Первая строка входного файла INPUT.TXT содержит натуральное число N – количество элементов в массиве. Во второй строке записаны N целых чисел A_i , разделенные пробелом. Третья строка содержит два натуральных числа L и R , которые задают отрезок поиска $[L, R]$ ($-1000 \leq A_i \leq 1000$, $1 \leq L \leq R \leq N \leq 1000$).

Выходные данные

В выходной файл OUTPUT.TXT выведите значение максимального элемента массива на отрезке $[L, R]$ и его индекс. В случае неоднозначности определения индекса следует вывести наименьший из них.

Пример

Найдем индекс k наибольшего элемента между l и r .

Пройдемся по элементам массива от левой до правой границы, `(for (int i=l+1;i<=r;i++) )`

первоначально $k=l$ если выполняется условие *if ($a[k] < a[i]$) тогда $k=i$*

ЗАДАЧА №1217

https://acmp.ru/asp/do/index.asp?main=task&id_course=1&id_section=5&id_topic=113&id_problem=694

Контроперация

(Время: 1 сек. Память: 16 Мб Сложность: 17%)

Хакер Василий получил доступ к классному журналу и хочет заменить все свои минимальные оценки на максимальные. Напишите программу, которая заменяет оценки Василия, но наоборот: все максимальные – на минимальные.

Входные данные

Первая строка входного файла INPUT.TXT содержит натуральное число N – количество оценок в журнале. Во второй строке записаны N целых чисел A_i – оценки Василия. Все числа во входных данных не превышают 1000 по абсолютной величине.

Выходные данные

В выходной файл OUTPUT.TXT выведите исправленные оценки, сохранив порядок.

1. *Найдем в массиве самую максимальную и минимальную отметки*
2. *Заменим максимальную на минимальную и выведем массив или при выводе массива находим максимальную и меняем её на минимальную*

ЗАДАЧА №496

https://acmp.ru/asp/do/index.asp?main=task&id_course=1&id_section=5&id_topic=113&id_problem=695

Сбор черники

(Время: 1 сек. Память: 16 Мб Сложность: 17%)

В фермерском хозяйстве в Карелии выращивают чернику. Она растет на круглой грядке, причем кусты высажены только по окружности. Таким образом, у каждого куста есть ровно два соседних. Всего на грядке растет N кустов.

Эти кусты обладают разной урожайностью, поэтому ко времени сбора на них выросло различное число ягод – на i -ом кусте выросло a_i ягод.

В этом фермерском хозяйстве внедрена система автоматического сбора черники. Эта система состоит из управляющего модуля и нескольких собирающих модулей. Собирающий модуль за один заход, находясь непосредственно перед некоторым кустом, собирает ягоды с этого куста и с двух соседних с ним.

Напишите программу для нахождения максимального числа ягод, которое может собрать за один заход собирающий модуль, находясь перед некоторым кустом заданной во входном файле грядки.

Входные данные

Первая строка входного файла INPUT.TXT содержит целое число N ($3 \leq N \leq 1000$) – количество кустов черники. Вторая строка содержит N целых положительных чисел $a_1, a_2, \dots, a_N$ – число ягод черники, растущее на соответствующем кусте. Все a_i не превосходят 1000.

Выходные данные

В выходной файл OUTPUT.TXT выведите ответ на задачу.

1. Проходим по массиву и рассматриваем сумму трех элементов. Находим наибольшую
2. Не забываем, что черника посажена по кругу и нам нужно еще рассмотреть два последних куста плюс первый куст и два первых куста плюс последний.

Попробуйте решить остальные задачи из блока **линейный поиск** (№1215 и №293).

Полный разбор задач блока **линейный поиск** и блока **Преобразования и анализ данных** добавлю после 07.03.25