

Решение задач с астр.ru – курсы – массивы

Линейный поиск

1214. 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 .

Решение.

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    int n, k=0, x;
    cin>>n;
    vector<int>a(n);
    for(int i=0; i<n; i++) cin>>a[i];    Ввод вектора
    cin>>x;
    for (int i=0; i<n; i++)                Ищем элемент
        if (a[i]==x) k++;                равный x
    cout<<k;

    return 0;
}
```

1216. 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.

Решение.

```
#include <iostream>
using namespace std;
int main()
{
    int a[1001], i, n, l, r, k;
    cin >> n;
    for (i = 1; i <= n; i++) cin >> a[i];
    cin >> l >> r;
    k = l;
    for (i = l + 1; i <= r; i++)
        if (a[i] > a[k]) k = i;

    cout << a[k] << ' ' << k;
    return 0;
}
```

Ищем индекс максимального элемента между границами l и r . Первоначально индекс наибольшего значения приравниваем левой границе

1217. 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.

Решение.

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

int main()
{
    int a[1000], i, n, x, y;
    cin >> n >> a[0];
    x = a[0];
    y = a[0];
    for (i = 1; i < n; i++)
    {
        cin >> a[i];
        if (a[i] > x) x = a[i];
        else if (a[i] < y) y = a[i];
    }
    for (i = 0; i < n; i++)
    {
        if (a[i] == x) a[i] = y;
        cout << a[i] << ' ';
    }
    return 0;
}
```

Вводим массив и сразу определим наибольшее и наименьшее значение

При выводе максимальные значения заменим на минимальные

496. 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.

Решение.

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    int n, m=0;
    cin >> n;
    vector<int> a(n);
    for (int i=0; i<n; i++) cin >> a[i];
    for (int i=0; i<n-2; i++)
        m=max(m, a[i]+a[i+1]+a[i+2]);
    m=max(m, a[n-1]+a[0]+a[1]);
    m=max(m, a[n-2]+a[n-1]+a[0]);
    cout << m << endl;
    return 0;
}
```

Обходим вектор и находим среди суммы трех значений максимальную.

После цикла нужно проверить суммы двух последних плюс первый элемент и двух первых плюс последний элемент

1215. Ближайшее число

(Время: 1 сек. Память: 16 Мб Сложность: 15%)

Требуется найти в массиве $A[1..N]$ самый близкий по величине элемент к заданному числу X .

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 значение элемента массива, ближайшее к X . Если таких значений несколько, следует вывести наименьшее среди них.

Решение.

```
#include <iostream>
#include <cmath>
#include <vector>

using namespace std;

int main()
{
    int n, x, m_x, Min_r;
    cin >> n;
    vector<int> a(n);
    for(int i=0; i<n; i++) cin >> a[i];
    cin >> x;

    m_x = 1e9;
    Min_r = 1e9;
    for (int i : a)
        if (abs(x - i) < Min_r)
        {
            m_x = i;
            Min_r = abs(x - i);
        }
    else
        if (abs(x - i) == Min_r)
            m_x = min(m_x, i);

    cout << m_x;
    return 0;
}
```

Если разница между x и элементом массива меньше минимальной разницы, тогда минимальную разницу сохраняем в Min_r , а элемент в m_x .

Если разница между x и элементом массива равна минимальной, тогда в m_x сохраняем наименьший элемент.

293. Налоги

(Время: 1 сек. Память: 16 Мб Сложность: 20%)

В некотором государстве действует N фирм, конкурирующих между собой. У каждой фирмы есть некоторая прибыль в год, равная $V[i]$ американских рублей. У царя есть любимые фирмы, а есть нелюбимые. Соответственно, налог для всех фирм разный и назначается царем в индивидуальном порядке. Налог на i -ую фирму равен $p[i]$ процентов.

Собиратели статистики решили посчитать, с какой фирмы в государственную казну идет наибольший доход (в казну идут все налоги). К сожалению, они не учили в детстве ни математику, ни информатику (так что учитесь, дети!), и их задача резко осложняется.

Помогите им в этой нелегкой задаче.

Входные данные

Во входном файле INPUT.TXT сначала записано число N - число фирм ($0 < N \leq 100$). Далее идет N целых неотрицательных чисел, не превышающих 154 - доходы фирм, а затем еще N целых чисел от 0 до 100 - налоги фирм в процентах.

Выходные данные

В выходной файл OUTPUT.TXT выведите одно число - номер фирмы, от которой государство получает наибольший налог. Если таких фирм несколько, выведите фирму с наименьшим номером.

Решение.

```
#include <iostream>

using namespace std;

int main()
{
    int n,i,p;
    cin>>n;
    int a[n+1];
    a[0]=0; int k=-1; int mx=-1;
    for (i=1;i<=n;i++) cin>>a[i];
    for (i=1;i<=n;i++)
    {
        cin>>p;
        if (a[i]*p>mx)
        {
            mx=a[i]*p;
            k=i;
        }
    }
    cout<<k;
    return 0;
}
```

Ищем индекс (номер)
фирмы, где налог больше.

Решение задач с астр.ру – курсы – массивы

Преобразование и анализ данных

149. Разворот

(Время: 1 сек. Память: 16 Мб Сложность: 9%)

Дано натуральное число N и последовательность из N элементов. Требуется вывести эту последовательность в обратном порядке.

Входные данные

В первой строке входного файла INPUT.TXT записано натуральное число N ($N \leq 10^3$). Во второй строке через пробел идут N целых чисел, по модулю не превосходящих 10^3 - элементы последовательности.

Выходные данные

В выходной файл OUTPUT.TXT выведите заданную последовательность в обратном порядке.

Решение.

```
#include <iostream>

using namespace std;

int main()
{
    int n;
    cin >> n;
    int a[n];
    for (int i = 0; i < n; i++) cin >> a[i];
    for (int i = n-1; i >= 0; i--) cout << a[i]<<' ';

    return 0;
}
```

284. Подмассив массива

(Время: 1 сек. Память: 16 Мб Сложность: 15%)

Пусть задан массив целых чисел $a_1, a_2, \dots, a_n$. Назовем его подмассивом $f(i,j)$ массив, составленный из чисел массива $a_i, a_{i+1}, \dots, a_{j-1}, a_j$. Напишите программу, которая будет выводить подмассивы массива a .

Входные данные

Первая строка входного файла INPUT.TXT содержит число n ($1 \leq n \leq 1000$) - количество элементов в массиве a . Во второй строке содержатся числа $a_1, a_2, \dots, a_n$ разделенные пробелом. Все a_i находятся в диапазоне от -2^{31} до $2^{31} - 1$. В третьей строке находится m ($1 \leq m \leq 100$) — количество подмассивов, которые необходимо вывести. Следующие m строк содержат пары чисел i_k, j_k ($1 \leq i_k \leq j_k \leq n$).

Выходные данные

В выходной файл OUTPUT.TXT для каждой пары (i_k, j_k) в отдельной строке выведите подмассив $f(i_k, j_k)$.

Решение.

```
#include<iostream>
using namespace std;
int main()
{
    int a[1000], i, n, k, L, R;
    cin >> n;
    for (i = 1; i <= n; i++) cin >> a[i];
    cin >> k;

    while (k)
    {
        k--;
        cin >> L >> R;
        for (i = L; i <= R; i++)
            cout << a[i] << ' ';
    }
    cout << endl;
    return 0;
}
```


1218. Шеренга

(Время: 1 сек. Память: 16 Мб Сложность: 18%)

Петя впервые пришел на урок физкультуры в новой школе. Перед началом урока ученики выстраиваются по росту, в порядке невозрастания. Напишите программу, которая определит на какое место в шеренге Пете нужно встать, чтобы не нарушить традицию, если заранее известен рост каждого ученика и эти данные уже расположены по невозрастанию (то есть каждое следующее число не больше предыдущего). Если в классе есть несколько учеников с таким же ростом, как у Пети, то программа должна расположить его после них.

Входные данные

Первая строка входного файла INPUT.TXT содержит натуральное число N ($N \leq 100$) – количество учеников (не считая Петю). Во второй строке записаны N натуральных чисел A_i ($A_i \leq 200$) – рост учеников в сантиметрах в порядке невозрастания. Третья строка содержит единственное натуральное число X ($X \leq 200$) – рост Пети.

Выходные данные

В выходной файл OUTPUT.TXT выведите единственное целое число – номер Пети в шеренге учеников.

Решение.

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n, x, t, b;
    cin >> n;
    vector<int> a;
    for(int i = 0 ; i < n; i++){
        cin >> t;
        a.push_back(t);
    }
    cin >> x;
    for(int i = 0; i < a.size(); i++){
        if(a[i] < x){
            cout << i + 1;
            return 0;
        }
    }
    cout << a.size() + 1;
    return 0;
}
```

Как только находим элемент меньший x выводим результат и заканчиваем программу.

Если такой элемент не найден выводим $n+1$

1219. Двойной переворот

(Время: 1 сек. Память: 16 Мб Сложность: 19%)

Дана последовательность натуральных чисел 1, 2, 3, ..., N. Необходимо сначала расположить в обратном порядке часть этой последовательности от элемента с номером A до элемента с номером B, а затем от C до D.

Входные данные

Входной файл INPUT.TXT содержит 5 натуральных чисел N, A, B, C и D ($A \leq B$; $C \leq D$; $1 \leq A, B, C, D \leq N \leq 1000$).

Выходные данные

В выходной файл OUTPUT.TXT выведите полученную последовательность, полученную в результате двойного переворота.

```
#include<iostream>

using namespace std;

int main()
{
    int x[1001], N, A, B, C, D;
    cin >> N >> A >> B >> C >> D;

    for (int i = 1; i <= N; i++) x[i] = i;

    for (int i = A; i <= (A + B) / 2; i++) swap(x[i], x[B + A - i]);

    for (int i = C; i <= (C + D) / 2; i++) swap(x[i], x[D + C - i]);

    for (int i = 1; i <= N; i++)
        cout << x[i] << ' ';

    return 0;
}
```