# Lab 3
# FIT Staff
# 3rd week

## 1 Topics

- Arrays

- Loop operators: for, while, do while

## 2 Reading Materials

In Russian

- `https://informatics.msk.ru/mod/book/view.php?id=559`

- `https://informatics.msk.ru/mod/book/view.php?id=559&chapterid=297`

In English

- `http://www.cplusplus.com/doc/tutorial/arrays/`

## 3 Problem Set

# 72477. Oddlover

| | |
|---|---|
| Input file: | standard input |
| Output file: | standard output |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given an array consisting of integers. Write a program, which outputs all odd elements of array.

## Input

The first line contains integer $n$ ($1 \leqslant n \leqslant 10^5$) — array size. The next line contains $n$ integers $a_i$ ($1 \leqslant a_i \leqslant 10^9$) — elements of array.

## Output

Print all odd elements.

## Examples

| standard input | standard output |
|---|---|
| 3<br>4 7 6 | 7 |
| 5<br>4 4 6 9 5 | 9 5 |
| 5<br>17 13 15 25 2 | 17 13 15 25 |
| 5<br>13 2 5 7 17 | 13 5 7 17 |
| 4<br>20 8 18 23 | 23 |

# 72485. Positive numbers

| | |
|---|---|
| Input file: | standard input |
| Output file: | standard output |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given an array consisting of integers. Write a program, which outputs count of positive numbers in array.

## Input

The first line contains integer $n$ $(1 \leqslant n \leqslant 10^5)$ — array size. The next line contains $n$ integers $a_i$ $(-10^9 \leqslant a_i \leqslant 10^9)$ — elements of array.

## Output

Single integer, positive numbers count.

## Examples

| standard input | standard output |
|---|---|
| 3<br>-8 -14 2 | 1 |
| 1<br>14 | 1 |
| 4<br>8 -13 -9 7 | 2 |
| 2<br>8 -2 | 1 |
| 3<br>-3 -12 -15 | 0 |

# 72568. Maxmimum in array

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given an array consisting of integers. Write a program, which outputs maximum in array.

## Input

The first line contains integer $n$ ($1 \leqslant n \leqslant 10^5$) — array size. The next line contains $n$ integers $a_i$ ($-10^9 \leqslant a_i \leqslant 10^9$) — elements of array.

## Output

Single integer, maximum in array.

## Examples

| standard input | standard output |
|---|---|
| 4<br>2 7 3 3 | 7 |
| 1<br>3 | 3 |
| 3<br>-1 9 5 | 9 |
| 4<br>-10 -2 -5 -8 | -2 |
| 2<br>1 8 | 8 |

# 72569. Position of maximum

| Input file: | standard input |
|---|---|
| Output file: | standard output |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given an array consisting of integers. Write a program, which finds position of maximum element of array. If maximums is two or more you should output first position.

## Input

The first line contains integer $n$ $(1 \leqslant n \leqslant 10^5)$ — array size. The next line contains $n$ integers $a_i$ $(-10^9 \leqslant a_i \leqslant 10^9)$ — elements of array.

## Output

Single integer, position of maximum in array.

## Examples

| standard input | standard output |
|---|---|
| 4<br>2 7 3 3 | 2 |
| 1<br>3 | 1 |
| 3<br>-1 9 5 | 2 |
| 4<br>-10 -2 -5 -8 | 2 |
| 2<br>1 8 | 2 |

# 72763. Sum of array

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given an array consisting of integers. Write a program, which finds sum of all elements.

## Input

The first line contains integer $n$ $(1 \leqslant n \leqslant 10^5)$ — array size. The next line contains $n$ integers $a_i$ $(-10^9 \leqslant a_i \leqslant 10^9)$ — elements of array.

## Output

Single number, sum of elements.

## Examples

| standard input | standard output |
|---|---|
| 4<br>2 7 3 3 | 15 |
| 1<br>3 | 3 |
| 3<br>-1 9 5 | 13 |
| 4<br>-10 -2 -5 -8 | -25 |
| 2<br>1 8 | 9 |

## Note

Sum can be big number use your mind.)

# 72764. Positive Minimum

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given an array consisting of integers. Print minimal positive number in array.

## Input

The first line contains integer $n$ ($1 \leqslant n \leqslant 10^5$) — array size. The next line contains $n$ integers $a_i$ ($-10^9 \leqslant a_i \leqslant 10^9$) — elements of array.

## Output

One number, minimal positive number in array.

## Examples

| standard input | standard output |
|---|---|
| 3<br>13 9 -2 | 9 |
| 5<br>-13 7 6 -7 -3 | 6 |
| 3<br>-4 -4 2 | 2 |
| 5<br>8 4 15 -15 -7 | 4 |
| 3<br>4 -5 -12 | 4 |

# 72768. Reverse

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given an array consisting of integers. Print all array in reverse order.

## Input

The first line contains integer $n$ $(1 \leqslant n \leqslant 10^5)$ — array size. The next line contains $n$ integers $a_i$ $(-10^9 \leqslant a_i \leqslant 10^9)$ — elements of array.

## Output

Reversed array.

## Examples

| standard input | standard output |
|---|---|
| 4<br>-13 5 -7 -15 | -15 -7 5 -13 |
| 5<br>-1 -2 6 14 11 | 11 14 6 -2 -1 |
| 2<br>8 14 | 14 8 |
| 5<br>9 1 -14 13 -8 | -8 13 -14 1 9 |
| 5<br>-11 -13 14 -2 7 | 7 -2 14 -13 -11 |

# 72780. MaxToMin

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given an array consisting of integers. Write a program, which will change all maximal elements to minimal elements of the array. Look to sample to better understand the conditions.

## Input

The first line contains integer $n$ $(1 \leqslant n \leqslant 10^5)$ — array size. The next line contains $n$ integers $a_i$ $(1 \leqslant a_i \leqslant 10^9)$ — elements of array.

## Output

Array with changed elements.

## Examples

| standard input | standard output |
|---|---|
| 3<br>9 5 5 | 5 5 5 |
| 4<br>1 4 14 6 | 1 4 1 6 |
| 5<br>3 1 2 2 5 | 3 1 2 2 1 |
| 5<br>1 7 6 15 5 | 1 7 6 1 5 |
| 3<br>1 14 7 | 1 1 7 |

## 72783. Sum l...r

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given an array consisting of integers. Write program, which will get sum of elements a[i], where $i$ $(l \leqslant i \leqslant r)$.

### Input

The first line contains three integers $n$, $l$, $r$, $(1 \leqslant n \leqslant 10^5, 1 \leqslant l \leqslant r \leqslant n)$ — array size and integrals. The next line contains $n$ integers $a_i$ $(1 \leqslant a_i \leqslant 10^9)$ — elements of array.

### Output

Sum of elements.

### Examples

| standard input | standard output |
|---|---|
| 1 1 1<br>12 | 12 |
| 2 2 2<br>12 15 | 15 |
| 5 4 5<br>7 5 8 11 7 | 18 |
| 3 3 3<br>15 13 5 | 5 |
| 5 3 5<br>15 12 13 10 3 | 26 |

## 72871. Reverse elements on l…r

| | |
|---|---|
| Input file: | standard input |
| Output file: | standard output |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given an array consisting of integers. Write program, which will reverse elements in interval $l$, $r$.

### Input

The first line contains three integers $n$, $l$, $r$, $(1 \leqslant n \leqslant 10^5, 1 \leqslant l \leqslant r \leqslant n)$ — array size and integrals. The next line contains $n$ integers $a_i$ $(1 \leqslant a_i \leqslant 10^9)$ — elements of array.

### Output

Array with changed elements.

### Examples

| standard input | standard output |
|---|---|
| 5 4 5<br>2 8 10 5 12 | 2 8 10 12 5 |
| 6 4 6<br>8 12 8 12 2 2 | 8 12 8 2 2 12 |
| 6 4 5<br>12 13 15 5 12 13 | 12 13 15 12 5 13 |
| 9 8 9<br>5 12 10 11 14 3 14 1 11 | 5 12 10 11 14 3 14 11 1 |
| 5 1 5<br>11 2 13 9 1 | 1 9 13 2 11 |

# 72901. How many times?

| | |
|---|---|
| Input file: | standard input |
| Output file: | standard output |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

You have an array of characters. Write a program, which will count how many times character appears in array.

## Input

The first line contains one integer $n$ $(1 \leqslant n \leqslant 10^5)$ — array size. The next line contains $n$ characters $a_i$ — elements of array is guaranteed they are lowercase Latin letters.

## Output

Each line must contains Latin letter in alphabet order and how many times it appears in the array. Look at the sample to better understand the problem.

## Examples

| standard input | standard output |
|---|---|
| 10<br>m b d a v q i s o r | a 1<br>b 1<br>d 1<br>i 1<br>m 1<br>o 1<br>q 1<br>r 1<br>s 1<br>v 1 |
| 2<br>n m | m 1<br>n 1 |
| 3<br>f q h | f 1<br>h 1<br>q 1 |
| 6<br>b w a k k b | a 1<br>b 2<br>k 2<br>w 1 |

# 73239. Upper bound -1?

| | |
|---|---|
| Input file: | standard input |
| Output file: | standard output |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given a sorted array and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.

You may assume no duplicates in the array.

## Input

The first line contains two integers $n$, $m$ $(1 \leqslant n \leqslant 10^5, 1 \leqslant m \leqslant 10^9)$ — array size and target value. The next line contains $n$ integers $a_i$ $(1 \leqslant a_i \leqslant 10^9)$ — elements of array.

## Output

Single integer, answer of problem.

## Examples

| standard input | standard output |
|---|---|
| 5 15<br>1 3 5 12 14 | 5 |
| 2 13<br>6 12 | 2 |
| 2 12<br>7 11 | 2 |
| 5 12<br>1 6 7 12 13 | 4 |
| 3 9<br>7 10 11 | 1 |

# 73242. Duplicates

| | |
|---|---|
| Input file: | standard input |
| Output file: | standard output |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given a sorted array, write a program which will delete all duplicates of elements and just save one of them.

## Input

The first line contains one integer $n$ $(1 \leqslant n \leqslant 10^5)$ — array size. The next line contains $n$ integers $a_i$ $(1 \leqslant a_i \leqslant 10^5)$ — elements of array.

## Output

Array of the unique and sorted elements

## Examples

| standard input | standard output |
|---|---|
| 3<br>1 1 4 | 1 4 |
| 2<br>1 5 | 1 5 |
| 5<br>1 2 3 4 5 | 1 2 3 4 5 |
| 3<br>2 3 4 | 2 3 4 |
| 3<br>1 4 4 | 1 4 |

## 73242. Duplicates

| | |
|---|---|
| Input file: | standard input |
| Output file: | standard output |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given a sorted array, write a program which will delete all duplicates of elements and just save one of them.

### Input

The first line contains one integer $n$ ($1 \leqslant n \leqslant 10^5$) — array size. The next line contains $n$ integers $a_i$ ($1 \leqslant a_i \leqslant 10^5$) — elements of array.

### Output

Array of the unique and sorted elements

### Examples

| standard input | standard output |
|---|---|
| 3<br>1 1 4 | 1 4 |
| 2<br>1 5 | 1 5 |
| 5<br>1 2 3 4 5 | 1 2 3 4 5 |
| 3<br>2 3 4 | 2 3 4 |
| 3<br>1 4 4 | 1 4 |

## 73247. Merge two arrays

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given two sorted integer arrays $a$ and $b$, merge $a$ into $b$ as one sorted array.

### Input

The first line contains one integer $n$ ($1 \leqslant n \leqslant 10^5$) — $a$ size. The second line contains $n$ integers $a_i$ ($1 \leqslant a_i \leqslant 10^5$) — elements of array. The third line contains one integer $m$ ($1 \leqslant m \leqslant 10^5$) — $b$ size. The next line contains $m$ integers $b_i$ ($1 \leqslant b_i \leqslant 10^5$) — elements of array.

### Output

Array B into sorted order.

### Examples

| standard input | standard output |
|---|---|
| 5<br>4 5 6 7 10<br>5<br>2 4 6 7 8 | 2 4 4 5 6 6 7 7 8 10 |
| 3<br>1 4 7<br>1<br>6 | 1 4 6 7 |
| 4<br>2 5 7 8<br>3<br>2 4 9 | 2 2 4 5 7 8 9 |
| 2<br>3 10<br>4<br>1 3 6 8 | 1 3 3 6 8 10 |
| 1<br>4<br>4<br>1 1 5 6 | 1 1 4 5 6 |

# 73257. Infinite?

| | |
|---|---|
| Input file: | standard input |
| Output file: | standard output |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

You are given an array, find the sum of all elements. But you don't know size of the array.

## Input

In the first line elements of array $a_i$ $(1 \leqslant a_i \leqslant 10^5)$.

## Output

Sum of all elements.

## Examples

| standard input | standard output |
|---|---|
| 50 18 81 | 149 |
| 27 12 | 39 |
| 16 24 7 72 | 119 |
| 10 66 91 | 167 |
| 34 61 90 24 44 46 3 73 37 34 4 7 14 76 20 48 58 80 17 71 | 841 |

# 73261. Square

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given an array, output the square value of each element.

## Input

The first line contains integer $n$ $(1 \leqslant n \leqslant 10^5)$ — array size. The next line contains $n$ integers $a_i$ $(1 \leqslant a_i \leqslant 10^4)$ — elements of array.

## Output

Changed array.

## Examples

| standard input | standard output |
|---|---|
| 5<br>6 7 15 6 11 | 36 49 225 36 121 |
| 4<br>7 14 9 3 | 49 196 81 9 |
| 5<br>3 4 3 13 5 | 9 16 9 169 25 |
| 2<br>3 5 | 9 25 |
| 3<br>6 12 4 | 36 144 16 |

# 73266. Sort

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given an array. Sort it. And reverse it.

## Input

The first line contains integer $n$ ($1 \leqslant n \leqslant 10^3$) — array size. The next line contains $n$ integers $a_i$ ($1 \leqslant a_i \leqslant 10^4$) — elements of array.

## Output

Sorted array in reversed order.

## Examples

| standard input | standard output |
|---|---|
| 9<br>4 11 1 7 5 1 6 10 8 | 11 10 8 7 6 5 4 1 1 |
| 4<br>10 10 11 13 | 13 11 10 10 |
| 6<br>2 15 3 13 5 13 | 15 13 13 5 3 2 |
| 4<br>3 4 3 6 | 6 4 3 3 |
| 3<br>6 3 2 | 6 3 2 |

# 73270. Is it square?

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given a single number $x$ check is it perfect square or not? Perfect square is numbers like 4, 9, 25.

## Input

Single integer $x$ $(1 \leqslant x \leqslant 10^9)$.

## Output

"Yes" if is it perfect square, "No" in other case.

## Examples

| standard input | standard output |
|---|---|
| 525 | No |
| 273 | No |
| 101 | No |
| 36 | Yes |
| 4 | Yes |

# 73273. Is it Prime?

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Given a single number $x$ check if it is prime number or not. A prime number (or a prime) is a natural number greater than 1 that cannot be formed by multiplying two smaller natural numbers.

## Input

Single integer $x$ $(1 \leqslant x \leqslant 10^6)$.

## Output

"Yes"if it is prime. "No"in other cases

## Examples

| standard input | standard output |
|---|---|
| 48 | No |
| 61 | Yes |
| 93 | No |
| 19 | Yes |
| 59 | Yes |
| 67 | Yes |

# 4 Lab Contest

All given task are emplaced in automated checker system for **lab3**:
http://acm.kbtu.kz/cgi-bin/new-register?action=211&contest_id=127
Feel free to submit your solutions without attempt count penalty.

# 5 Solutions

**Problem 72477**

```cpp
#include <iostream>
using namespace std;
int main () {
    int n;
    cin >> n;
    int a[500500];
    for (int i = 1; i <= n; ++ i) {
        cin >> a[i];
    }
    for (int i = 1; i <= n; ++ i) {
        if (a[i] % 2 == 1)
            cout << a[i] << ' ';
    }
    return 0;
}
```

**Problem 72485**

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

int main () {
    int n;
    cin >> n;
    int a[500500];
    for (int i = 1; i <= n; ++ i) {
        cin >> a[i];
    }
    int cnt = 0;
    for (int i = 1; i <= n; ++ i) {
```

```cpp
        if (a[i] > 0)
            ++ cnt;
    }
    cout << cnt;
    return 0;
}
```

**Problem 72568**

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

int main () {
    int n;
    cin >> n;
    int a[500500];
    for (int i = 1; i <= n; ++ i) {
        cin >> a[i];
    }
    int x = -1e9 - 17;
    for (int i = 1; i <= n; ++ i) {
        if (a[i] > x)
            x = a[i];
    }
    cout << x;
    return 0;
}
```

**Problem 72569**

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

int main () {
    int n;
    cin >> n;
    int a[500500];
    for (int i = 1; i <= n; ++ i) {
        cin >> a[i];
    }
    int x = -1e9 - 17, y = 0;
```

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

## Problem 72763

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

int main () {
    int n;
    cin >> n;
    int a[500500];
    long long sum = 0;
    for (int i = 1; i <= n; ++ i) {
        cin >> a[i];
    }
    for (int i = 1; i <= n; ++ i) {
        sum += a[i];
    }
    cout << sum;
    return 0;
}
```

## Problem 72764

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

int main () {
    int n;
    cin >> n;
    int a[500500];
    int x = 1e9 + 17;
    for (int i = 1; i <= n; ++ i) {
        cin >> a[i];
    }
```

```cpp
    for (int i = 1; i <= n; ++ i) {
        if (a[i] > 0)
            x = min (x, a[i]);
    }
    cout << x;
    return 0;
}
```

## Problem 72768

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

int main () {
    int n;
    cin >> n;
    int a[500500];
    for (int i = 1; i <= n; ++ i) {
        cin >> a[i];
    }
    for (int i = n; i >= 1; -- i) {
        cout << a[i] << ' ';
    }
    return 0;
}
```

## Problem 72780

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

int main () {
    int n;
    cin >> n;
    int a[500500];
    int mx = 0, mn = 1e9 + 17;
    for (int i = 1; i <= n; ++ i) {
        cin >> a[i];
    }
    for (int i = 1; i <= n; ++ i) {
        if (a[i] > mx)
```

```cpp
                mx = a[i];
            if (a[i] < mn)
                mn = a[i];
        }
        for (int i = 1; i <= n; ++ i)
            if (a[i] == mx)
                a[i] = mn;
        for (int i = 1; i <= n; ++ i)
            cout << a[i] << ' ';
        return 0;
}
```

**Problem 72783**

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

int main () {
    int n, l, r;
    cin >> n >> l >> r;
    long long ans = 0;
    int a[500500];
    for (int i = 1; i <= n; ++ i) {
        cin >> a[i];
    }
    for (int i = l; i <= r; ++ i) {
        ans += a[i];
    }
    cout << ans;

    return 0;
}
```

**Problem 72871**

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

int main () {
    int n, l, r;
    cin >> n >> l >> r;
```

```
    long long ans = 0;
    int a[500500];
    for (int i = 1; i <= n; ++ i) {
        cin >> a[i];
    }
    for (int i = 1; i < l; ++ i)
        cout << a[i] << ' ';
    for (int i = r; i >= l; -- i)
        cout << a[i] << ' ';
    for (int i = r + 1; i <= n; ++ i)
        cout << a[i] << ' ';

    return 0;
}
```

## Problem 72901

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

int cnt[26];

int main () {
    int n;
    cin >> n;
    for (int i = 1; i <= n; ++ i) {
        char x;
        cin >> x;
        cnt[x - 'a'] ++;
    }
    for (int i = 0; i < 26; ++ i) {
        if (cnt[i] == 0)
            continue;
        cout << char (i + 'a') << ' ' << cnt[i] << endl;
    }

    return 0;
}
```

## Problem 73239

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

int main () {
    int n, m, k;
    cin >> n >> m;
    int a[500500];
    for (int i = 1; i <= n; ++ i) {
        cin >> a[i];
    }
    for (int i = 1; i <= n; ++ i) {
        if (a[i] <= m)
            k = i;
    }
    cout << k;
    return 0;
}
```

## Problem 73242

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

int main () {
    int n, m, k;
    cin >> n;
    int a[500500];
    for (int i = 1; i <= n; ++ i) {
        cin >> a[i];
    }
    for (int i = 1; i <= n; ++ i) {
        if (a[i] != a[i - 1])
            cout << a[i] << ' ';
    }
    return 0;
}
```

## Problem 73247

```cpp
#include <iostream>
```

```cpp
using namespace std;

int main () {
    int n, m, k;
    cin >> n;
    int a[500500], b[500500], c[500500];
    for (int i = 1; i <= n; ++ i) {
        cin >> a[i];
    }
    cin >> m;
    for (int i = 1; i <= m; ++ i) {
        cin >> b[i];
    }
    int x = 1, y = 1;
    a[n + 1] = 1e9 + 17;
    b[m + 1] = 1e9 + 17;
    for (int i = 1; i <= n + m; ++ i) {
        if (a[x] >= b[y]) {
            c[i] = b[y];
            ++ y;
        }
        else {
            c[i] = a[x];
            ++ x;
        }
    }
    for (int i = 1; i <= n + m; ++ i) {
        cout << c[i] << ' ';
    }
    return 0;
}
```

**Problem 73257**

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

int main () {
    int n, m, k = 0;
    while (cin >> n) {
```

```cpp
        k += n;
    }
    cout << k;
    return 0;
}
```

## Problem 73261

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

int main () {
    int n, a[500500];
    cin >> n;
    for (int i = 1; i <= n; ++ i) {
        cin >> a[i];
    }
    for (int i = 1; i <= n; ++ i) {
        a[i] = a[i] * a[i];
    }
    for (int i = 1; i <= n; ++ i) {
        cout << a[i] << ' ';
    }
    return 0;
}
```

## Problem 73266

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

int main () {
    int n, a[500500];
    cin >> n;
    for (int i = 1; i <= n; ++ i) {
        cin >> a[i];
    }
    for (int i = 1; i <= n; ++ i) {
        for (int j = 1; j <= n; ++ j) {
            if (a[i] >= a[j])
                swap (a[i], a[j]);
```

```
        }
    }
    for (int i = 1; i <= n; ++ i) {
        cout << a[i] << ' ';
    }
    return 0;
}
```

## Problem 73270

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

int main () {
    int n;
    cin >> n;
    double x = sqrt (n);
    int y = x;
    if (y * y == n) {
        cout << "Yes";
    }
    else {
        cout << "No";
    }
    return 0;
}
```

## Problem 73273

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

int main () {
    int n;
    cin >> n;
    if (n == 1) {
        cout << "No";
        return 0;
    }
```

```cpp
    for (int i = 2; i * i <= n; ++ i) {
        if (n % i == 0) {
            cout << "No";
            return 0;
        }
    }
    cout << "Yes";
    return 0;
}
```

# 6   Additional tasks for this lab

You can solve problems starting from A to J from the link below:
https://informatics.msk.ru/mod/statements/view.php?id=208
*note: statements are in russian*