Lab 8
FIT Staff
9th week

1 Topics

- STL

2 Reading Materials

In Russian

- `http://cppstudio.com/post/8453/`
- `http://cppstudio.com/post/9037/`

In English

- `https://www.geeksforgeeks.org/set-in-cpp-stl/`
- `https://www.geeksforgeeks.org/vector-in-cpp-stl/`

3 Problem Set

4 Lab Contest

All given task are emplaced in automated checker system for **lab8**:
`http://acm.kbtu.kz/cgi-bin/new-register?action=211&contest_id=164`
Feel free to submit your solutions without attempt count penalty.

Problem A. 76433. ASC sort

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

You are given N integers. Sort the N integers by ascending order. Store N integers in a vector.

Input

The first line of input contains the number N - number of integers.
The second line contains a sequence of integers.

Output

Print out sorted integers

Example

standard input	standard output

`4`
`20 3 7 9`
	`3 7 9 20`

Problem B. 76435. Reverse array

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

You are given N integers. Your task is to reverse sequence of integers. Store N integers in a vector.

Input

The first line of input contains the number N - number of integers.
The second line contains a sequence of integers.

Output

Print out reversed integers

Example

standard input	standard output
5 3 4 1 8 6	6 8 1 4 3

Problem C. 76440. Reverse in range

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

You are given n integers. Then index ranges a and b, $(0 \leq a < b \leq n-1)$. Your task is to reverse array elements in a given range ([a...b] — index range bounds inclusively). Store n integers in a vector.

Input

The first line of input contains the number n - the number of integers.
The second line contains a sequence of integers. The third line contains integers a and b

Output

Print out result sequence

Example

standard input	standard output

5
9 -1 2 8 6
1 3
	9 8 2 -1 6

Problem D. 76470. Vector erase

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

You are given n integers. Then you are given an integer k — index in the vector. Your task is to erase value at given index k.

Input

The first line of input contains the number n, $(1 \leq n \leq 100)$ — the number of integers.
The second line contains a sequence of integers.
The third line contains an integer k, $(0 \leq k \leq n-1)$ — index in the vector.

Output

Print out elements of the vector separated by space after erase operation

Example

standard input	standard output

`5`
`3 1 2 9 6`
`3`
	`3 1 2 6`

Problem E. 76471. Erase range

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

You are given n integers. Then index ranges a and b, $(0 \leq a < b \leq n-1)$. Your task is to erase sequence elements in a given range ([a...b] — index range bounds inclusively). Store n integers in a vector.

Input

The first line of input contains the number n - the number of integers.
The second line contains a sequence of integers. The third line contains integers a and b

Output

Print out result sequence after erase operation

Example

`standard input`	`standard output`

`5`
`2 3 9 -1 8`
`2 3`
	`2 3 8`

Problem F. 76472. Add element

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

You are given n integers. Then two integers k, $(0 \leq k \leq n - 1)$ — index in the sequence and a — given value. Your task is to insert number a at index k. Store n integers in a vector.

Input

The first line of input contains the number n - the number of integers. The second line contains a sequence of integers. The third line contains integers k and a.

Output

Print out newly created sequence

Example

standard input	standard output

5
2 3 8 -3 4
1 10
	2 10 3 8 -3 4

Problem G. 76473. Number of primes

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

You are given n, $(1 \leq n \leq 100)$ positive integers, each integer $v[i]$, $(1 \leq v[i] \leq 1000)$. Then one integer number k, $(1 \leq k \leq 200)$. Your task is to show a count of prime numbers in a sequence that are greater than k. Store n integers in a vector. You should create function $isPrime$ to check for the prime.

Input

The first line of input contains the number n - the number of integers. The second line contains a sequence of integers. The third line contains integers k.

Output

Output single number — count result

Example

standard input	standard output

5
3 4 59 13 7
8
	2

Problem H. 76503. K smallest numbers

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

You are given n, $(1 \leq n \leq 100)$ positive integers, each integer $v[i]$, $(1 \leq v[i] \leq 1000)$. Then one integer number k, $(1 \leq k \leq n-1)$. Find k smallest numbers from given sequence.

Input

The first line of input contains the number n - the number of integers. The second line contains a sequence of integers. The third line contains integers k.

Output

Print out k smallest numbers

Example

standard input	standard output

5
100 3 40 143 20
2
	3 20

Problem I. 76504. Find K

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

Input

You are given n, $(1 \le n \le 100)$ positive integers, each integer $v[i]$, $(1 \le v[i] \le 1000)$. Then one integer number k, $(1 \le k \le 1000)$.

Output

Output Yes if k found at least ones in the sequence, No otherwise.

Examples

`standard input`	`standard output`

`4`
`9 30 4 -3`
`4`
	`Yes`

`4`
`2 3 0 1`
`10`
	`No`

Problem J. 76509. Big difference

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

Input

You are given n, $(1 \leq n \leq 100)$ integers, each integer $v[i]$, $(1 \leq v[i] \leq 1000)$.

Output

Output difference between the largest and smallest values in the sequence

Example

standard input	standard output
4 10 2 3 14	12

Problem K. 76512. Sum of K largest

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

You are given n, $(1 \leq n \leq 100)$ positive integers, each integer $v[i]$, $(1 \leq v[i] \leq 1000)$. Then one integer number k, $(1 \leq k \leq n-1)$. Find the sum of k largest numbers from given sequence.

Input

The first line of input contains the number n - the number of integers. The second line contains a sequence of integers. The third line contains integers k.

Output

Print out single number — sum of k largest numbers

Example

`standard input`	`standard output`

`4`
`2 12 4 10`
`2`
	`22`

Problem L. 76513. Unique elements

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

Input

You are given n, $(1 \leq n \leq 100)$ integers, each integer $v[i]$, $(1 \leq v[i] \leq 1000)$.

Output

Print out single number — the number of unique elements

Example

standard input	standard output

`5`
`1 2 1 3 3`
	`3`

Problem M. 76515. Sum of uniques

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

Input

You are given n, $(1 \le n \le 100)$ integers, each integer $v[i]$, $(1 \le v[i] \le 1000)$.

Output

Print out single number — the sum of unique elements.

Example

standard input	standard output

`5`
`1 1 2 3 3`
	`6`

Problem N. 76520. Remove evens

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

Input

You are given n, $(1 \leq n \leq 100)$ integers, each integer $v[i]$, $(1 \leq v[i] \leq 1000)$.
Create new sequence with unique elements. Then remove even numbers from new sequence.

Output

Print out sequence after removing operation

Example

`standard input`	`standard output`

`5`
`1 1 2 3 3`
	`1 3`

Problem O. 76521. Sort letters

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

Input

Single line of input contains one string — s which consists of upper and lower case letter.

Output

In first line output single integer — the number of unique letters from given string
Second line contains lower case each letter in the alphabet separated by space

Examples

standard input	standard output

`testsample`	`7`
	`a e l m p s t`

`SecondTEST`	`7`
	`c d e n o s t`

5 solutions

problem 76433

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

using namespace std;

int main(){
  int n;
  cin >> n;
  vector<int> v;
  for(int i = 0;i < n;i ++) {
       int x;
       cin >> x;
       v.push_back(x);
  }
  sort(v.begin(),v.end());
  for(int i = 0;i < v.size();i ++) cout << v[i] << " ";
  return 0;
}
```

problem 76435

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

using namespace std;

int main(){
  int n;
  cin >> n;
  vector<int> v;
  for(int i = 0;i < n;i ++) {
       int x;
       cin >> x;
       v.push_back(x);
  }
  reverse(v.begin(),v.end());
  for(int i = 0;i < v.size();i ++) cout << v[i] << " ";
  return 0;
}
```

problem 76440

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

using namespace std;

int main(){
  int n, x, y;
  cin >> n;
  vector<int> v;
  for(int i = 0;i < n;i ++) {
    int cur;
    cin >> cur;
    v.push_back(cur);
  }
  cin >> x >> y;

  reverse(v.begin() + x, v.begin() + y + 1);

  for(int i = 0; i < n; i++)
    cout << v[i] << " ";


  return 0;
}
```

problem 76470

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

using namespace std;
```

```
int main(){
  int n, x, a;
  cin >> n;
  vector<int> v;

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

  v.erase(v.begin()+x);

  for(int i = 0; i < v.size(); i++)
    cout << v[i] << " ";


  return 0;
}
```

problem 76471

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

using namespace std;

int main(){
  int n, x, y, a;
  cin >> n;
  vector<int> v;

  for(int i = 0; i < n; i++){
    cin >> a;
    v.push_back(a);
  }
  cin >> x >> y;

  v.erase(v.begin() + x, v.begin() + y + 1);
  for(int i = 0; i < v.size(); i++)
    cout << v[i] << " ";


  return 0;
}
```

problem 76472

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

using namespace std;

int main(){
  int n, x, y, a;
  cin >> n;
  vector<int> v;

  for(int i = 0; i < n; i++){
    cin >> a;
    v.push_back(a);
  }
  cin >> x >> y;

  v.insert(v.begin() + x, y);

  for(int i = 0; i < v.size(); i++)
    cout << v[i] << " ";


  return 0;
}
```

problem 76473

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

```
#include <cmath>

using namespace std;

bool isPrime(int n){
  if(n == 1) return false;
  for(int i = 2; i < sqrt(n); i++){
    if(n % i == 0) return false;
  }
  return true;
}

int main(){
  int n, k, a, cnt = 0;
  cin >> n;
  vector<int> v;

  for(int i = 0; i < n; i++){
    cin >> a;
    v.push_back(a);
  }
  cin >> k;

  for(int i = 0; i < v.size(); i++)
    if(v[i] > k && isPrime(v[i]))
      cnt++;

  cout << cnt;


  return 0;
}
```

problem 76503

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

```
using namespace std;


int main(){
  int n, k, a;
  cin >> n;
  vector<int> v;

  for(int i = 0; i < n; i++){
    cin >> a;
    v.push_back(a);
  }
  cin >> k;
  sort(v.begin(), v.end());


  for(int i = 0; i < k; i++)
    cout << v[i] << " ";

  cout << endl;


  return 0;
}
```

problem 76504

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

using namespace std;


int main(){
  int n, k, a, cnt = 0;
  cin >> n;
  vector<int> v;
  for(int i = 0; i < n; i++){
    cin >> a;
    v.push_back(a);
  }
  cin >> k;

  if(find(v.begin(), v.end(), k) != v.end())
    cout << "Yes";
  else
    cout << "No";


  return 0;
}
```

problem 76509

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

using namespace std;


int main(){
  int n, a;
  cin >> n;
  vector<int> v;

  for(int i = 0; i < n; i++){
    cin >> a;
    v.push_back(a);
  }
  sort(v.begin(), v.end());

  cout << v[v.size() - 1] - v[0] << endl;

  return 0;
}
```

problem 76512

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

using namespace std;

bool cmp(int a, int b){
  return a > b;
}

int main(){
  int n, a, k, sum = 0;
  cin >> n;
  vector<int> v;

  for(int i = 0; i < n; i++){
    cin >> a;
    v.push_back(a);
  }
  cin >> k;
  sort(v.begin(), v.end(), cmp);

  for(int i = 0; i < k; i++)
    sum += v[i];

  cout << sum << "\n";

  return 0;
}
```

problem 76513

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

```
using namespace std;

int main(){
  int n, a;
  cin >> n;
  set<int> v;

  for(int i = 0; i < n; i++){
    cin >> a;
    v.insert(a);
  }

  cout << v.size() << "\n";

  return 0;
}
```

problem 76515

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


using namespace std;

int main(){
  int n, a, sum = 0;
  cin >> n;
  set<int> v;

  for(int i = 0; i < n; i++){
    cin >> a;
    v.insert(a);
  }

  for(set<int>::iterator it = v.begin(); it != v.end(); it++)
```

```
    sum += *it;

  cout << sum << endl;

  return 0;
}
```

problem 76520

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


using namespace std;

int main(){
  int n, a;
  cin >> n;
  set<int> v;

  for(int i = 0; i < n; i++){
    cin >> a;
    v.insert(a);
  }

  for(set<int>::iterator it = v.begin(); it != v.end(); it++)
    if((*it) % 2 != 0)
      cout << *it << " ";

  cout << endl;

  return 0;

}
```

problem 76521

```
#include <iostream>
```

```
#include <algorithm>
#include <vector>
#include <cmath>
#include <set>


using namespace std;

int main(){
  string s;
  cin >> s;
  set<int> v;

  for(int i = 0; i < s.size(); i++){
    v.insert(tolower(s[i]));
  }

  cout << v.size() << endl;

  for(set<int>::iterator it = v.begin(); it != v.end(); it++)
    cout << (char)*it << " ";

  cout << endl;

  return 0;

}
```