

# 1 Allocate array

---

## Description

---

Write a program that allocates an array of N integers, initializes each element by its index multiplied by 5 and then prints the obtained array on the console.

## Input

---

- On the only line you will receive the number N

## Output

---

- Print the obtained array on the console.
  - Each number should be on a new line

## Constraints

---

- $1 \leq N \leq 20$
- N will always be a valid integer number
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

---

Input	Output
5	0 5 10 15 20

## 2 Compare arrays

---

### Description

---

Write a program that reads two integer arrays of size N from the console and compares them element by element.

### Input

---

- On the first line you will receive the number N
- On the next N lines the numbers of the first array will be given
- On the next N lines the numbers of the second array will be given

### Output

---

- Print `Equal` if the two arrays are the same and `Not equal` if they are not

### Constraints

---

- $1 \leq N \leq 20$
- N will always be a valid integer number
- Time limit: 0.1s
- Memory limit: 16MB

### Sample tests

---

Input	Output
3 1 2 3 1 2 3	Equal

3	Not equal
1	
1	
1	
2	
2	
2	

## 3 Compare char arrays

---

Write a program that compares two `char` arrays lexicographically (letter by letter).

### Input

---

- On the first line you will receive the first `char` array as a string
- On the second line you will receive the second `char` array as a string

### Output

---

- Print `<` if the first array is lexicographically smaller
- Print `>` if the second array is lexicographically smaller
- Print `=` if the arrays are equal

### Constraints

---

- $1 \leq \text{size of arrays} \leq 128$
- Time limit: 0.1s
- Memory limit: 16MB

### Sample tests

---

Input	Output
-------	--------

hello halo	>
food food	=

## 4 Maximal sequence

---

### Description

---

Write a program that finds the length of the maximal sequence of equal elements in an array of N integers.

### Input

---

- On the first line you will receive the number N
- On the next N lines the numbers of the array will be given

### Output

---

- Print the length of the maximal sequence

### Constraints

---

- $1 \leq N \leq 1024$
- Time limit: 0.1s
- Memory limit: 16MB

### Sample tests

---

Input	Output
-------	--------

10	3
2	
1	
1	
2	
3	
3	
2	
2	
2	
1	

## 5 Maximal increasing sequence

---

### Description

---

Write a program that finds the length of the maximal increasing sequence in an array of N integers.

### Input

---

- On the first line you will receive the number N
- On the next N lines the numbers of the array will be given

### Output

---

- Print the length of the maximal increasing sequence

### Constraints

---

- $1 \leq N \leq 1024$
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

---

Input	Output
8 7 3 2 3 4 2 2 4	3

## 6 Maximal K sum

---

### Description

---

Write a program that reads two integer numbers N and K and an array of N elements from the console. Find the maximal sum of K elements in the array.

### Input

---

- On the first line you will receive the number N
- On the second line you will receive the number K
- On the next N lines the numbers of the array will be given

### Output

---

- Print the maximal sum of K elements in the array

### Constraints

---

- $1 \leq N \leq 1024$

- $1 \leq K \leq N$
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

---

Input	Output
8 3 3 2 3 -2 5 4 2 7	16

## 7 Selection sort

---

### Description

---

Sorting an array means to arrange its elements in increasing order. Write a program to sort an array. Use the [Selection sort](#) algorithm: Find the smallest element, move it at the first position, find the smallest from the rest, move it at the second position, etc.

### Input

---

- On the first line you will receive the number  $N$
- On the next  $N$  lines the numbers of the array will be given

### Output

---

- Print the sorted array
  - Each number should be on a new line

## Constraints

---

- $1 \leq N \leq 1024$
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

---

Input	Output
6	1
3	2
4	3
1	4
5	5
2	6
6	
10	1
36	10
10	20
1	27
34	28
28	30
38	31
31	34
27	36
30	38
20	

## 8 Maximal sum

---



## Description

---

Write a program that finds the maximal sum of consecutive elements in a given array of N numbers.

- *Can you do it with only one loop (with single scan through the elements of the array)?*

## Input

---

- On the first line you will receive the number N
- On the next N lines the numbers of the array will be given

## Output

---

- Print the maximal sum of consecutive numbers

## Constraints

---

- $1 \leq N \leq 1024$
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

---

Input	Output
10 2 3 -6 -1 2 -1 6 4 -8 8	11

# 9 Frequent number

---

## Description

---

Write a program that finds the most frequent number in an array of N elements.

## Input

---

- On the first line you will receive the number N
- On the next N lines the numbers of the array will be given

## Output

---

- Print the most frequent number and how many time it is repeated
  - Output should be REPEATING\_NUMBER (REPEATED\_TIMES times)

## Constraints

---

- $1 \leq N \leq 1024$
- $0 \leq \text{each number in the array} \leq 10000$
- There will be only one most frequent number
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

---

Input	Output
-------	--------

13	4 (5
4	times)
1	
1	
4	
2	
3	
4	
4	
1	
2	
4	
9	
3	

## 10 Find sum in array

---

### Description

---

Write a program that finds in a given array of integers with length N a sequence of given sum s (if present).

### Input

---

- On the first line you will receive the number N
- On the second line you will receive the number S
- On the next N lines the numbers of the array will be given

### Output

---

- Print the sequence. Each element on a new line
- Print "No Subset Adds To Sum" if there is no subset that adds up to the requested sum

# Constraints

---

- 1 <= N <= 1024
- 0 <= each number in the array <= 10000
- Only 1 sequence in the array should add up to the required sum
- Time limit: 0.1s
- Memory limit: 16MB

# Sample tests

---

array	result
7 11 4 3 1 4 2 5 8	4 2 5
7 11 6 3 1 3 2 4 8	No Subset Adds To Sum

# 11 Binary search

---

## Description

---

Write a program that finds the index of given element X in a sorted array of N integers by using the [Binary search](#) algorithm.

## Input

---

- On the first line you will receive the number N
- On the next N lines the numbers of the array will be given
- On the last line you will receive the number X

## Output

---

- Print the index where X is in the array
  - If there is more than one occurrence print the first one
  - If there are no occurrences print -1

## Constraints

---

- $1 \leq N \leq 1024$
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

---

Input	Output
-------	--------

10	6
1	
2	
4	
8	
16	
31	
32	
64	
77	
99	
32	

## 12 Index of letters

---

### Description

---

Write a program that creates an array containing all letters from the alphabet (a-z).  
Read a word from the console and print the index of each of its letters in the array.

### Input

---

- On the first line you will receive the word

### Output

---

- Print the index of each of the word's letters in the array
  - Each index should be on a new line

### Constraints

---

- $1 \leq \text{word length} \leq 128$
- Word is consisted of lowercase english letters
- Time limit: 0.1s

- Memory limit: 16MB

## Sample tests

---

Input	Output
telerikacademy	19 4 11 4 17 8 10 0 2 0 3 4 12 24

## 13 Merge sort

---

### Description

---

Write a program that sorts an array of integers using the [Merge sort](#) algorithm.

### Input

---

- On the first line you will receive the number N
- On the next N lines the numbers of the array will be given

### Output

---

- Print the sorted array
  - Each number should be on a new line

## Constraints

---

- $1 \leq N \leq 1024$
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

---

Input	Output
6	1
3	2
4	3
1	4
5	5
2	6
6	
10	1
36	10
10	20
1	27
34	28
28	30
38	31
31	34
27	36
30	38
20	

## 14 Quick sort

---



## Description

---

Write a program that sorts an array of integers using the [Quick sort](#) algorithm.

## Input

---

- On the first line you will receive the number N
- On the next N lines the numbers of the array will be given

## Output

---

- Print the sorted array
  - Each number should be on a new line

## Constraints

---

- $1 \leq N \leq 1024$
- Time limit: 0.1s
- Memory limit: 16MB

## Sample tests

---

Input	Output
6	1
3	2
4	3
1	4
5	5
2	6
6	

10	1
36	10
10	20
1	27
34	28
28	30
38	31
31	34
27	36
30	38
20	

# 15 Prime numbers

---

## Description

---

Write a program that finds all prime numbers in the range  $[1 \dots N]$ . Use the [Sieve of Eratosthenes](#) algorithm. The program should print the biggest prime number which is  $\leq N$ .

## Input

---

- On the first line you will receive the number  $N$

## Output

---

- Print the biggest prime number which is  $\leq N$

## Constraints

---

- $2 \leq N \leq 10\,000\,000$
- Time limit: 0.3s
- Memory limit: 64MB

## Sample tests

---

Input	Output
13	13
126	113
26	23

# 16 Remove elements from array

---

## Description

---

Write a program that reads an array of integers and removes from it a minimal number of elements in such a way that the remaining array is sorted in increasing order. Print the minimal number of elements that need to be removed in order for the array to become sorted.

## Input

---

- On the first line you will receive the number  $N$
- On the next  $N$  lines the numbers of the array will be given

## Output

---

- Print the minimal number of elements that need to be removed

## Constraints

---

- $1 \leq N \leq 1024$
- Time limit: 0.1s
- Memory limit: 16MB

# Sample tests

---

Input	Output
8 1 4 3 3 6 3 2 3	3
10 14 2 9 8 3 13 17 19 30 1	4