

## Lab 2 (Array)

### Problem #1 | No Bracket

This problem will help you to know how arrays can be represented.

We will challenge you to write a program to print out an array in which 'square brackets' are **not allowed** (**NO ANY SQUARE BRACKETS ARE ALLOWED IN YOUR SOURCE FILE**). So, you will need to write the array in a different way.

Then print out the values in the array according to the mode. Which are:

- 0: Print out the even index of the array
- 1: Print out the odd index of the array

The array will start with index '0'. If there's no element to print out, print 'none'

#### Input

The input consists of three lines:

First Line	An integer $n$ in which $1 \leq n \leq 10,000$ as the number of elements in an array.
Second Line	$n$ integers that are elements of the array. Separating each element with a 'whitespace'
Third Line	An integer to determine mode, which can be only two choices, 1 or 0

#### Output

The output consists of only one line, which is the sequence of the array according to mode. Each element is separated by 'whitespace'

#### Examples of input and output

Input	Output
5 1 1 1 2 3 0	1 1 3
10 6 5 8 4 9 7 8 5 1 3 1	5 4 7 5 3
1 2 1	none

**Reminder:** **DO NOT** write array by using square brackets. If you do so, your score will be deducted.

### Problem #2 | No Bracket No Printf

This problem will help you to know how array and function are used together and how we can pass array into functions.

We will challenge you to write a program to print out an array in which 'square brackets' are **not allowed (NO ANY SQUARE BRACKET IS ALLOWED IN YOUR SOURCE FILE)**. Additionally, we will challenge you not to call any `printf()` functions in the **main function**. And finally, you must declare the array in the main function only.

Your task is to write an array and find the largest and the smallest element in the array. And then find their index, starting with index 0. If there is any repetition in the largest or the smallest element, print the leftmost index of them.

#### Input

The input consists of two lines:

First Line	An integer $n$ in which $1 \leq n \leq 10,000$ as the number of elements in an array.
Second Line	$n$ integers that are elements of the array. Separating each element with a 'whitespace'

#### Output

The output consists of two lines:

First Line	The largest element in the array and its index. Separating with a 'whitespace'
Second Line	The smallest element in the array and its index. Separating with a 'whitespace'

#### Examples of input and output

Input	Output
9 1 5 8 7 654 231 48 5 0	654 4 0 8
5 2 8 1 -7 -7	8 1 -7 3

**Hint:** you will have to use the function to print the array. For example,

```
#include<stdio.h>
void your_function(.....){ //function name and the number of functions is your choice
    //write your code here
}

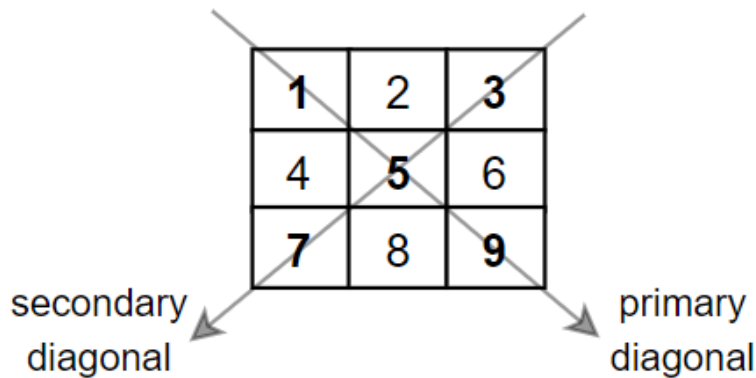
int main(){

    ..... //declare an array without any square brackets.
    return 0;
```

### Problem #3 | Sum of Diagonal Matrix

Write a program in C that takes input for the number of rows and columns, and then input the matrix elements. The program should then calculate and output the sum of the elements along the primary diagonal and the secondary diagonal separately.

Ensure that your program handles the case where the number of rows is **not equal** to the number of columns. In such cases, your program should print "ERROR" and exit program.



#### Input

The input consists of two lines:

First Line	Two integers in which as the row and column.
Second Line	Enter numbers into the matrix.

#### Output

The program calculates the sum of the elements along the primary diagonal and the secondary diagonal.

#### Examples of input and output

Input	Output
3 3 1 5 2 6 4 9 0 3 5	Primary: 10 Secondary: 6
4 4 2 -4 -5 -1 0 3 -2 -9 -3 -1 4 7 7 6 -8 -12	Primary: -3 Secondary: 3
2 3	ERROR

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int rows, cols, primary = 0, secondary = 0;
    //write your code here
    int **matrix = (int **)malloc(rows * sizeof(int *));
    for (int i = 0; i < rows; i++) {
        matrix[i] = (int *)malloc(cols * sizeof(int));
    }
    //write your code here
    for (int i = 0; i < rows && i < cols; i++) {
        primary += matrix[i][i];
    }
    for (int i = 0; i < rows && i < cols; i++) {
        secondary += matrix[i][cols - i - 1];
    }
    //write your code here
    free(matrix);
    return 0;
}
```

## Problem #4 | Dictionary

dictionary is a collection of items that allows us to store data in key: value pairs. Similar to array that stored the data by the index.

Write a program to insert the dictionary data type. Additionally, make the program to edit the value in dictionary by the keys (if no change was made, display No change before the original elements in the dictionary).

Input

The input consists of size + 2 lines:

First Line	An integer <i>size</i> in which $1 \leq size \leq 10$ as the number of elements in an array.
Second to size + 1 <sup>st</sup> lines	Two string, key and value separating between element with a whitespace, for inserting elements in the dictionary. Which their length is $1 \leq length \leq 100$
Size + 2 line (last line)	Two strings, key and value separating between element with a whitespace, for editing the value in the dictionary by the key.

Output

The output consists of size lines:

First Line to size <sup>th</sup> line	The elements of dictionary after edited separated by whitespaces and colon ':'
---------------------------------------	--

If there were no change the output consists of size+1 lines

First Line	"No change" to indicate that no change was made.
Second Line to size + 1 <sup>st</sup> line	The <b>original</b> elements in the dictionary, including keys and values separated by whitespaces and colon ':'.

Example of input and output

Input	Output
3 apple A bird B cat C cat D	apple : A bird : B cat : D
5 name Ken age 20 height 180 gender female size thin gender male	name : Ken age : 20 height : 180 gender : male size : thin

3 University KMUTT Faculty CPE Year 1 name Gunn	No change University : KMUTT Faculty : CPE Year : 1
---	--

Hint: you will need string.h to check the keys

```
#include<stdio.h>
#include<string.h> //hint

struct dict{
    char value[100];
    char key[100];
};

void editdict(){ //for editing the dictionary
    //write your code here
}

void printdict(struct dict *dic, int size){ //for printing the output
    for(int i=0; i<size ; i++){
        printf("%s : %s\n",dic[i].key,dic[i].value);
    }
}

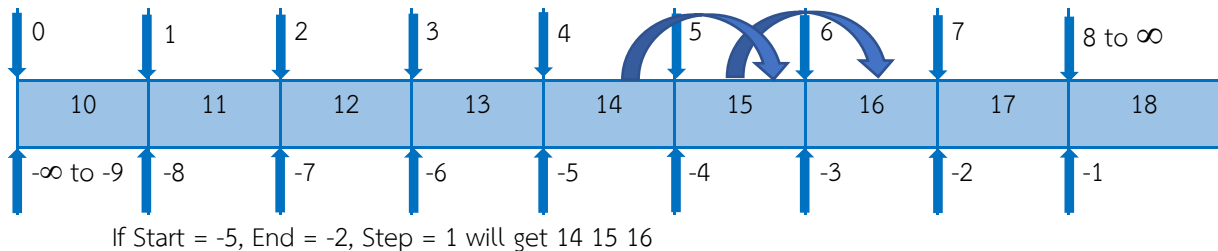
int main(){
    int size;
    scanf("%d",&size);
    struct dict dic[size]; //you can do this to declare structure in the array

    //write your code here
}
```

### Challenge #1 | List Slicing

Write a program to create the structure of the list from given template by entering the number of elements in list then collect the elements from user input and get input which are the initial index, end index and step of index increment. [You can look into more information how the slicing work.](#)

Example.



#### Input

The input consists of two lines:

First Line	A number of elements $1 \leq n \leq 1,000,000$ in list.
Second Line	A sequence of integer (int) separating between element with a whitespace
Third Line	Three integers as start, end and step

#### Output

The output consists of one line, print out all the elements in the sliced list. If there is no element in the sliced list print "empty"

#### Example of input and output

Input	Output
7 1 2 3 4 5 6 7 0 3 1	1 2 3
1 1 0 1 1	1
8 -2 -1 -1 0 4 5 7 9 -1 -7 -2	9 5 0