Exercises

5 – Arrays and Pointers

Department of Computer Science
University of Pisa
Largo B. Pontecorvo 3
56127 Pisa

Exercise 1

Write a C program, that contains a function `diff_abs` that takes as arguments 2 float (`a` and `b`) and whose invokation is such that the variable corresponding to the first argument contains the value $a - b$, while the variable corresponding to the second argument contains $b - a$. Print the content of the 2 variables after the execution of the function, with a precision up to the second decimal digit.

Input

```
3.14
-2.71
```

Output

```
5.85
-5.85
```

Exercise 2

Write a C program that reads 3 integers from the standard input and executes function `ordered_swap(·,·,·)` to the three variables that store the three input values. The read values have to be passed to the function in the same order they are acquired from the standard input.
Implement the function `ordered_swap`, that swaps the content of the three input variables so that the variable corresponding to the first argument contains that smallest integer, the second variable contains the median integer, and the third variable the largest integer.
Print the content of the three variables after the execution of the function.

Input

```
14
-1
7
```

Output

```
-1
7
14
```

Exercise 3

Write a C program that reads 1 integer *N* from the keyboard and passes it to a function that allocates an array of size *N* and fills it with the values of the first *N* multiples of 5. In main, print the content of the array returned by the function.
Assumption - *N* is a > 0 integer.

Input

```
5
```

Output

```
5
10
15
20
25
```

Exercise 4

Write a C program that reads 6 integers from the keyboard and assigns the first 5 values at the first 5 positions of an array; store the sixth value in a variable N. Write a function that, given input the array initialized eith the first 5 values from the keyboard and the integer N, returns the array resized to contain $5 + N$ elements, such that each one of the new N elements corresponds to the sum of the numbers before it in the array, i.e.

$$arr[i] = \sum_{j=0}^{i-1} arr[j].$$

In main, print the content of the array returned by the function.

Input

```
-1
2
3
0
4
3
```

Output

```
-1
2
3
0
4
8
16
32
```