

CSC 215

Procedural Programming with C

Lab #4

Headers and Source Files, and Make file

Tutorial Section

- Header and Source files.
 - You need to include the header file in the source file and main file.
 - In the Header file, write the definitions of the functions.
 - In the Source file, implement the functions in the header file.
- Make file.
 - Use gedit to create a file with the name **makefile**.
 - Type the following into the file.

```
CC=gcc
CFLAGS=-I.
DEPS = HEADER_FILE_NAME.h
OBJ = MAIN_FILE_NAME.o SOURCE_FILE_NAME.o

%.o: %.c $(DEPS)
    $(CC) -c -o $@ $< $(CFLAGS)

PROGRAM_NAME: $(OBJ)
    gcc -o $@ $^ $(CFLAGS)

.PHONY: clean

clean:
    rm -f *.o
```

- Replace "*HEADER_FILE_NAME*" with the name of the header file.
- Replace "*SOURCE_FILE_NAME*" with the name of the source file.
- Replace "*MAIN_FILE_NAME*" with the name of the main file.
- Replace "*PROGRAM_NAME*" with the any name you want your program to be called.
- If you are using windows, replace **rm** with **del**.
- To compile the program just type **make**.
- To remove the files created by compiling type **make clean**.

Lab Section

- Create the header file IntArray.h and the source file IntArray.c that have the following functions:
 - A function called **max**. It will take an array of integers and its size as a parameter. Then returns the **index** of the maximum integer. If there can't be a maximum number, it should return -1. The methods signature should look like this:
 - `int max(int arr[], int size)`
 - A function called **secondMax**. It will take an array of integers and its size as a parameter. Then returns the **index** of the second maximum integer. If there can't be a second maximum number, it should return -1. The methods signature should look like this:
 - `int secondMax(int arr[], int size)`
 - A function called **min**. It will take an array of integers and its size as a parameter. Then returns the **index** of the minimum integer. If there can't be a minimum number, it should return -1. The methods signature should look like this:
 - `int min(int arr[], int size)`
 - A function called **secondMin**. It will take an array of integers and its size as a parameter. Then returns the **index** of the second minimum integer. If there can't be a second minimum number, it should return -1. The methods signature should look like this:
 - `int secondMin(int arr[], int size)`
 - A function called **average**. It will take an array of integers and its size as a parameter. Then returns the average of the numbers. If there were no elements in the array, it should return zero. The methods signature should look like this:
 - `float average(int arr[], int size)`
- The main function should do the following:
 - Create an array of integers with the size of 10.
 - Read integer numbers from the user until he types 0 (Zero) or reaches 10. Zero should not be stored in the array.
 - Show the following:
 1. The maximum number and its index in the array.
 2. The second maximum number and its index in the array.
 3. The minimum number and its index in the array.
 4. The second minimum number and its index in the array.
 5. The average of the numbers in the array. Show 2 digits after the floating point.
- Show your program to the instructor. Then upload it to LMS under Lab3 Homework.

- Example run:

```

$ ./lab4
Type numbers [0 to exit]:-
43 32 5 56 8 2 65 -324 23 41
=====
The Maximum number is 65 and its index is 6.
The Second Maximum number is 56 and its index is 3.
The Minimum number is -324 and its index is 7.
The Second Minimum number is 2 and its index is 5.
The Average of the numbers is -4.90
=====
$ ./lab4
Type numbers [0 to exit]:-
4 6 0
=====
The Maximum number is 6 and its index is 1.
The Second Maximum number is 4 and its index is 0.
The Minimum number is 4 and its index is 0.
The Second Minimum number is 6 and its index is 1.
The Average of the numbers is 5.00
=====
$ ./lab4
Type numbers [0 to exit]:-
0
=====
Can't find a maximum number.
Can't find a second maximum number.
Can't find a minimum number.
Can't find a second minimum number.
The Average of the numbers is 0.00
=====
$ ./lab4
Type numbers [0 to exit]:-
10 0
=====
The Maximum number is 10 and its index is 0.
Can't find a second maximum number.
The Minimum number is 10 and its index is 0.
Can't find a second minimum number.
The Average of the numbers is 10.00
=====

```

SUBMIT POLICY: -

- Use the follow naming convention: Lab04_ID_FirstName_LastName.zip
 - **Which must contain: Lab4.c, IntArray.c, IntArray.h, makefile**
 - **Example:** Lab04_123456789_Marwan_Almaymoni.zip
- Use a comment to write your name and ID at the beginning of the code.
- The Deadline is: 16/03/2015 right before the Lab starts.
- Late submissions will not be accepted.
- Email submissions will not be accepted.
- **-1 Point** for not following the naming convention.
- **-1 Point** for not writing your name and ID in the code inside a comment.
- **-8 Points** if the submitted program didn't work due to syntax errors.
- **-10 Points** for cheating and helping others cheat.