Academic Year	**2024**		
Semester	☐ Fall	☒ Winter	☐ Summer
Course Code - Name	CSCI 3310 – System Programming		
Instructor	Dr. Razi Iqbal		
Assessment	Lab 3		
Deadline	Submit during the lab		

Lab 3

The main purpose of this lab is to test your knowledge of Threads.

Instructions:

- You are required to submit C file (.c).
- Students having exactly similar code will get a straight 0.
- You are required to complete these questions using any Linux Distro.
- The soft deadline for submission of this lab is during the lab session and the hard deadline is 2 days after your lab session. If for some reason, you are not able to complete this lab in the lab session, please inform the TAs.

Question

You are required to create a child thread from a parent thread which is expected to populate an integer array which is then displayed by the parent thread.

The program is expected to take a number from the command line arguments which indicates the number of elements to be added to an integer array starting from 0 to that number (exclusive). For example, if 10 is provided from the command line arguments, it means that array should have numbers from 0 to 9 (total of 10 numbers). Below are the responsibilities of both Parent and the Child threads:

Parent Thread:

This thread is responsible for

- Displaying its thread ID
- Creating a child thread and passing it the number of elements to be added to the array received through command line arguments along with a function which would run for the child thread
- Joining the threads
- Finally, displaying all the elements inserted in the integer array

Child Thread:

This thread is responsible for

- Displaying its thread ID
- Creating an integer array to store integer values (size is received from the parent thread which basically is the only command line argument passed)
- Returning the integer array to the parent thread so that parent thread can display the elements of this array
- Finally, exiting the thread

Note the following:

- You need to create a separate function that would run in the child thread.
- You might want to look into `malloc` function in C that helps in creating dynamic arrays in C. For example, below is the line of code you can use to create a dynamic array in the heap memory:

  ```
  int *numbers = malloc(size* sizeof(int));
  ```

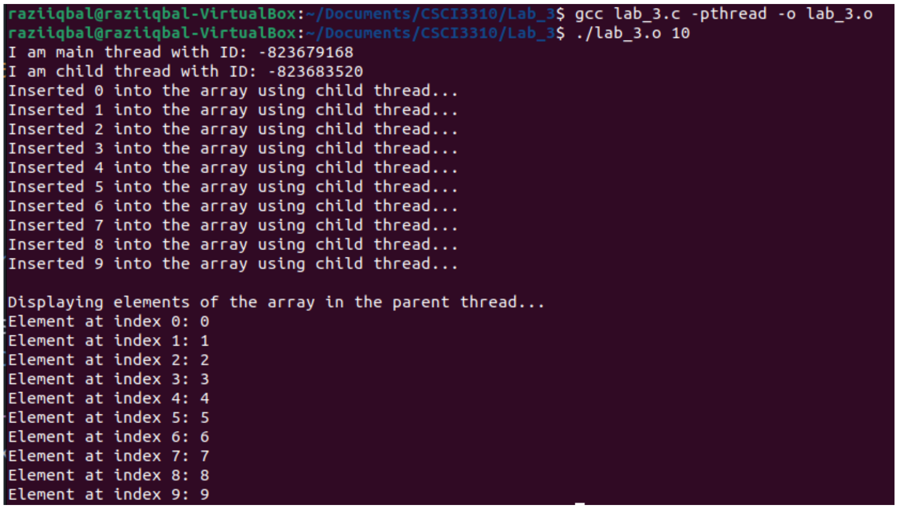
- Do not forget to release the memory allocated to dynamic array by using `free` function. For example, below is the line of code you can use to release the memory occupied using `malloc` (result is a pointer pointing to that dynamic memory in the main method returned by thread function):

  ```
  free(result);
  ```

- Do not use global variables.

Hint: use `malloc` in the function created for the child thread and `free` in the main function.

Below is the expected output of the program:



```
raziiqbal@raziiqbal-VirtualBox:~/Documents/CSCI3310/Lab_3$ gcc lab_3.c -pthread -o lab_3.o
raziiqbal@raziiqbal-VirtualBox:~/Documents/CSCI3310/Lab_3$ ./lab_3.o 10
I am main thread with ID: -823679168
I am child thread with ID: -823683520
Inserted 0 into the array using child thread...
Inserted 1 into the array using child thread...
Inserted 2 into the array using child thread...
Inserted 3 into the array using child thread...
Inserted 4 into the array using child thread...
Inserted 5 into the array using child thread...
Inserted 6 into the array using child thread...
Inserted 7 into the array using child thread...
Inserted 8 into the array using child thread...
Inserted 9 into the array using child thread...

Displaying elements of the array in the parent thread...
Element at index 0: 0
Element at index 1: 1
Element at index 2: 2
Element at index 3: 3
Element at index 4: 4
Element at index 5: 5
Element at index 6: 6
Element at index 7: 7
Element at index 8: 8
Element at index 9: 9
```