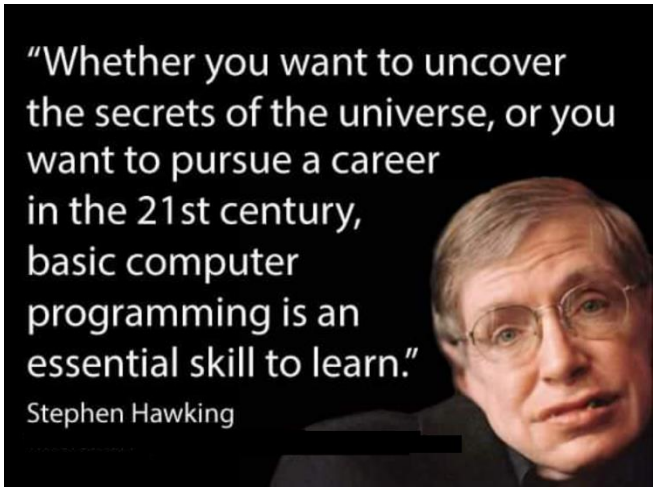

Al Akhawayn University in Ifrane

School of Science and Engineering

CSC2302 Project 2

Fall2022

By: Hanaa Talei

This is your second CSC2302 project that counts 5% of your overall grade

☛ You need to submit your project **Monday November 21st** before midnight

- 1 group member will have to send .c file as an email attachment.
- Name your file as follow: StudentName1_StudentName2_Project2.c
- Email to use: H.Talei@aui.ma
- You need to CC your project partner
- An email received Tuesday November 22nd starting 00:00 will not be accepted: **You get a zero**

☛ Assume the responsibility for the work you will submit

- Any form of plagiarism will imply a WF as course grade

☛ The main goal of this project is to make you work with queues implemented using linked lists

---

Some applications require the use of what we call a **circular queue** adopting an algorithm called round robin. In round robin, time slices are assigned to each process in the queue in equal portions and in a circular order without priority.

☛ The queue still has a front and a rear and follows FIFO!

When I explained the tower of Hanoi game in class, some of you expressed their interest to get the toy from me and play with it. I will use a circular queue to manage your requests using the following menu:

1. Add a student to a queue
2. Time is up
3. Print the content of the queue
4. Student is no more interested
5. I need my toy back

Read the following instructions to know what to implement:

- Each student will be represented with a name and an ID
- **Option1**: is used to add an already created node to the queue
    - You are using enqueue() but your queue must be circular
- **Option2:** The time allocated for the first student in the queue is over
    - Queue might be empty!
    - The first student in the queue has the choice to remain in the queue (as he/she still want to play more with toy) or leave the queue
        - If the student decide to stop playing: dequeue()
        - Otherwise, decide what to do…..
            - Remember front should point to the node to serve and rear must point to the last element in the queue!
- **Option 3:** You need to print the list of the students interested to have the toy from me
    - Probably no student is interested so far
- **Option 4:** A student X wants to leave the queue as he/she is not interested in getting the toy from me. Let's design a function that will remove a student with ID Y from the queue.
    - The student is probably not in the queue!
- **Option 5:** I decided to have my toy back! Let's destroy the queue!!
    - Remember this is not serving!!

- Make sure that you free any node that you don't need anymore
- In addition to the functions stated in this project description, you can add more functions as needed.
- Make sure that you avoid getting input or printing results in functions

Good Luck!