

# Algorithms and data structures

## Tutorial 3 - Stacks and Queues

Follow the below guide:

- After a labwork, you will have one week (or 7 days) to complete all exercises. All submissions must be sent before 23:59 of the day before the next labwork day.
- Compress all code source files in a zip file and rename it as FULLNAME-ID-TT#no.zip (e.g NguyenVanA-070-TT1.zip). Save your files according to the exercise number i.e Ex1.cpp, Ex2.c, etc.
- Only code source files (.c or .cpp) should be in the zip files. Other files (.exe, .o) MUST be removed from the zip file.
- Send to this email: doan-nhat.quang@usth.edu.vn
- Copy/Paste from any source is not tolerated. Penalty will be applied for late submission.
- **NOTE: You must follow the guide. Incorrect zip file name, zip files containing other files (.exe), copy/paste lead to heavy penalty (no score).**

### Exercise 1: 7 points

Write a program in C/C++ to enter a string of characters i.e. your name, then try to compute this string in reverse order using a Stack structure with the Linked List principle.

For example: "Nguyen Van A"  $\rightarrow$  "A Nav Neyugn"

### Exercise 2: 8 points

Assume that a queue of N passengers waiting to enter a train car

- Only the first person in the queue can enter at a time
- New persons arrive will go to the end of the queue.

Implement basic operations for Queue ADT using the Linked List principle to manage the queue.

**Exercise 3:** 5 points

Suppose that we play a following game. There are a list 10 numbers from 0 to 9 and an empty stack.

- Step 1: We draw randomly a number  $n_1$  from the list and push it into a stack.
- Step 2: We perform randomly a second draw to choose another number  $n_2$  from the list and put it into the stack. If  $n_1 == n_2$ , we must draw again.
- Step 3: While the first two are still unknown. Don't reveal these number then guess a third number  $n_3$ . This number is entered from keyboard.
- Step 4: Verify that the third number is between the first and the second ( $n_1 < n_3 < n_2$  or  $n_2 < n_3 < n_1$ ). If true, you win.
- Step 5: Otherwise, pop up the second number from the stack. Return to the Step 2. If we still guess wrong number this time, your turn is over and clear the stack to play the game again.

Write this game in C/C++ using the Linked List implementation for the stack.

To obtain a random number from 0 to 9, use the following code.

```
#include <time.h>
int main(){
    srand ( time(NULL) );
    int iSecret = rand() % 9 + 1;
    printf("%d", iSecret);
}
```