Homework Assignment

Class:	CS202	**Semester:**	Fall 2019
Assignment type:	Homework assignment	**Due date:**	12/1/19
Assignment topic:	Stack	**Assignment no.**	8b
Delivery:	WebCampus – cpp files		

Goal

Practice the use of linked lists and stacks

General remarks

- Keep all your testing code in submitted **`cpp`** files
- For all the problems, ensure/add the proper memory allocation/deallocation (all instructions about memory are not necessarily mentioned in the instruction).
- For all the problems, please use **`valgrind`** tool to confirm the proper memory management. Use the command:

```
valgrind --tool=memcheck --leak-check=yes --show-reachable=yes
--num-callers=20 --track-fds=yes ./01.o
```

where `01.o` is the name of tested binary file

Problem II. Stack class template (30p)

Implement template class for the stack. The stack is of the size of *n* elements, and *n* is given as parameter to the constructor. Implement the following class *myStack*:

```
public:
  void push(<Type>) // puts the integer element onto the stack
  <Type> pop()      // retrieves the element from the top of the
stack
  void disp()       // prints the entire stack
private:
  int stackPointer  // points to the top free spot in the stack.
  <Type> *elements
```

Use a dynamic array. Throw exceptions when:

- **`pop`** function was used when stack is empty
- **`push`** function was used when stack is full

Catch exceptions in **`main()`** function, rethrow in **`pop()`** and **`push()`**

Prepare a menu (see sample output). Clear screen at the beginning of each iteration (before printing the menu). To clear the screen on *bobby* (this might not work on other systems) use the following:

```
#include <stdlib.h>
… // some code...
system("clear");
```

Sample output/operation:

```
Stack:
Menu:
1. push element
2. pop element
3. exit
Enter: 1
Enter value: 5

Stack: 5
Menu:
1. push element
2. pop element
3. exit
Enter: 1
Enter value: 7

Stack: 5,7
Menu:
1. push element
2. pop element
3. exit
Enter: 1
Enter value: 3
Stack: 5,7,3
Menu:
1. push element
2. pop element
3. exit
Enter: 1
Enter value: 11

Stack: 5,7,3,11
Menu:
1. push element
2. pop element
3. exit
Enter: 2

Stack: 5,7,3
Menu:
1. push element
2. pop element
3. exit
Enter: 2
```

```
Stack: 5,7
Menu:
1. push element
2. pop element
3. exit
Enter: 2
Popped element: 7

Stack: 5
Menu:
1. push element
2. pop element
3. exit
Enter: 2

Stack: 5
Menu:
1. push element
2. pop element
3. exit
Enter: 2
can't pop from empty stack
error operating the stack at position 0

Stack:
Menu:
1. push element
2. pop element
3. exit
Enter: 3
```

Submission:

Include the following elements in your submission: (**rid** = your rebel id)

Problem	Element	File
Problem I	Code of your program (for problem 1)	rid_1.cpp file
	Summary of the submission	
	Summary: 1 cpp file, submit them to the WebCampus (add all the files as the single submission). Remember about proper names of the files!	