Department of Information Systems and Technologies

CTIS 152 – Data Structures and Algorithms

Spring 2024 - 2025

Lab Guide #15 – Week 10-2

OBJECTIVE : Stack Library and Stack Exercises

Instructors : Serpil TIN
Assistants : Berk ÖNDER & Hatice Zehra YILMAZ

Copy the stack_int.h file from Moodle to each of your local project folders and modify it when necessary!

Q1. Write a C program that reads a number bigger than 30 and prints all of its digits via a stack.

Example Run:

```
Enter a number: 5
Enter a number: 6
Enter a number: 7
Enter a number: 32
Digits of 32 are 3 2
```

Project Name: LG15_Q1
File Name: Q1.cpp

Q2. Make the necessary changes in the header file to create a **string** stack.
Write a C program that reads several names from a file named **"players.txt"** into a stack and displays the given names in reverse order.

Example Run:

```
PLAYERS IN REVERSE ORDER
***********************
Semih Kilicsoy
Milot Rashica
Gabriel Paulista
Mert Gunok
Gedson Fernandes
Rafa Silva
Ciro Immobile
```

players.txt

```
Ciro Immobile
Rafa Silva
Gedson Fernandes
Mert Gunok
Gabriel Paulista
Milot Rashica
Semih Kilicsoy
```

Project Name: LG15_Q2
File Name: Q2.cpp

Q3. Make the necessary changes in the header file to create a structure stack.
Write a C program that gets information about several computer competitions from the binary file named "**competitions.bin**". The binary file contains the competition name (50 chars), start time (24 chars), and end time (24 chars for all competitions. The program will display the information on the competitions on the screen in reverse order using a STACK.

Example Run:

```
Competition Name                                   Start Time              End Time
************************************************************************************************
Brute Force 4.0 - Cybersecurity                    Apr 03, 2021 03:30 PM   Apr 04, 2021 03:30 PM
ThoughtWorks Data Engineer Hiring Challenge        Mar 19, 2021 02:30 PM   Mar 28, 2021 09:25 PM
AGEING BETTER WITH ICTs                            Mar 15, 2021 11:00 AM   Apr 16, 2021 12:59 AM
Data Structures and Algorithms Coding Contest      Mar 13, 2021 06:00 AM   Mar 13, 2021 07:30 AM
Peak Senior Software Engineer Hiring Challenge     Mar 05, 2021 02:30 PM   Mar 14, 2021 08:25 PM
Organic Food Supply Chain Hackathon                Mar 04, 2021 08:30 PM   Mar 28, 2021 09:25 PM
```

Project Name: LG15_Q3
File Name: Q3.cpp

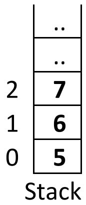
Q4. Write a C program that reads an octal number (base 8) and finds the number's decimal equivalent using a stack.
Write the following function;

- `convertDecimal` that takes a number as a parameter, finds and returns the decimal equivalent of the number using an integer stack.

Project_name: LG15_Q4
File_name: Q4.cpp

HINT: You can use the built-in function **pow** by including the math header file.

Octal num: **765**,
Decimal equivalent: $\mathbf{(7 \times 8^2) + (6 \times 8^1) + (5 \times 8^0) = 501}$



Example Run #1:

```
Enter an octal number: 247
Decimal equivalent of the number is: 167
```

Example Run #2:

```
Enter an octal number: 473
Decimal equivalent of the number is: 315
```

ADDITIONAL QUESTIONS

AQ1.

You are supposed to make some additions to stack_int.h by writing some functions in it, which are:

DisplayStack : Displays the stack,
CountStack : Counts the elements of the stack (Stack content does not change!),
RemMaxStack : Removes the Maximum element from the stack,
SendNthToEnd : The nth element from the top is sent to the bottom of the stack.

Write a C program that gets numbers from the user to fill the stack until a sentinel value is entered (-9 for instance), then displays a menu, and calls the appropriate STACK function according to the user's choice. Examine well the example run.

Project Name: LG15_AQ1
File Name: AQ1.cpp

Example Run:

```
Enter a number: 23
Enter a number: 42
Enter a number: 56
Enter a number: 87
Enter a number: 33
Enter a number: -9

       MENU
************************
1) Count Stack
2) Remove Maximum Element
3) Send Nth To End
4) Exit
Enter your choice: 1

STACK CONTENT
33
87
56
42
23

Number of elements in the stack: 5

       MENU
************************
1) Count Stack
2) Remove Maximum Element
3) Send Nth To End
4) Exit
Enter your choice: 2

STACK CONTENT
33
56
42
23
       MENU
```

```
************************
1) Count Stack
2) Remove Maximum Element
3) Send Nth To End
4) Exit
Enter your choice: 3

Enter N: 2

STACK CONTENT
33
42
23
56
       MENU
************************
1) Count Stack
2) Remove Maximum Element
3) Send Nth To End
4) Exit
Enter your choice: 4
```

AQ2.

Write a C program that evaluates a postfix expression such as
`6 2 + 5 * 8 4 / -`

The program should read a postfix expression consisting of positive digits and operators into a character array and will be evaluated as follows:

`(((6+2) *5) - (8/4))`

⟶

from left to right

Postfix Expression : Op1 Op2 operator
Ex : 6 2 +

Infix Expression : Op1 operator Op2
Ex : 6 + 2

For this reason, use a stack where values should be pushed to the stack in the order (op1,op2, …).
The arithmetic operations allowed in an expression are:
+ addition
- subtraction
* multiplication
/ division
Assume input data is a VALID postfix expression.

Attention: The expression is given as a string. To convert any number in character format to a number, you need the following example code;

```
If (exp[i] >= '0' && exp[i]<= '9')
 num = exp[i] – ‘0’;
 a + b = b + a
 a * b = b * a
```

but

```
 a – b ≠ b – a
 a / b ≠ b / a
```

Example Run #1:

```
Enter an expression: 4 8 + 6 5 - * 3 2 - 3 9 + * /
```

```
The result is: 1
```

Example Run #2:

```
Enter an expression: 9 8 * 3 + 1 6 4 * + /
The result is: 3
```

Project Name: LG15_AQ2
File Name: AQ2.cpp