**COLLEGE OF COMPUTING AND INFORMATICS**

**CSEB3213/CSEB324/CSNB344 DATA STRUCTURES AND ALGORITHMS**

**SEM 1 2023/2024**

**LAB 4: STACKS**

**Objectives**

Introduction on Stacks concept and operations in both STL and Linked List implementations using C++ programming language.

**Instruction**

1. This is an individual lab exercise.
2. Please answer Question 1 and 2. **Write in the template and submit in Wednesday’s class (13/12/2023)**
3. **Please write the output too**

**Template for Answer (print/write this and submit Wednesday class (13/12/2023))**

**Lab 4: Stack (STL and Linked List Implementation)**

Name: …………………………………….

ID : ……………………………………….

Date : …………………………..

**Question 1: Codes**

**Sample of Output**

**Question 2: Codes**

**Sample of Output**

**LEVEL: EASY**

**Question 1**

Referring to sample of program below:

1. Complete the **main()**. This function should be able to invoke all functions in the program.
2. Write a function named **analysis()**. This function should be able to display:

* type of data for each value (even or odd) in stack.
* total even value in stack.
* total odd value in stack.

1. Write a function named **pop()**. This function should be able to remove all numbers in stack.

|  |
| --- |
| **Sample of Program** |
| #include<iostream>  using namespace std;  struct Data{  int no;  Data \*next;  };  void push(/\*suitable parameter\*/){  Data \*n = new Data;  n->no = val;  n->next = NULL;    /\*insertion process\*/  }    **//Question 1(b)**  **//Question 1(c)**  int main() {  Data \*head = NULL; int size; int val;  cout<<"Enter total value to insert: ";  cin>> size;  for(int i = 1; i<=size; i++){  cout<<"Enter a value : ";  cin>>val;  **//Question 1(a)**  }    **//Question 1(a)**    cout<<"\nEnd of program";  return 0;  } |

|  |
| --- |
| **Sample of Output** |
|  |

**Question 2**

Referring to your solution in **Question 1**, convert the program to **STL Stack** implementation.

**HOMEWORK**

**LEVEL: MODERATE**

**Question 3 (10 marks)**

***Source: Lab Test Semester 1 2020/2021 (Set 1)***

**CASE STUDY**

One of the well-known stack applications is palindrome checker. A palindrome is a word, phrase, number or sequence of words that reads the same backwards as forwards.

*Example: racecar*

Referring to above case study, creates a Palindrome program in C++ using **Stack STL** implementation and the appropriate functions. Refer to sample output below. Your application should therefore be able to identify whether the provided word is a palindrome or not.

|  |  |
| --- | --- |
| SAMPLE OUTPUT |  |
| :: Palindrome Program ::  Enter total character: 5  Enter character : m a d a m  Your word : madam  Reverse order: madam  Palindrome : Yes  Thank you for using this program. | :: Palindrome Program ::  Enter total character: 4  Enter character : f o o d  Your word : food  Reverse order: doof  Palindrome : No  Thank you for using this program. |

**madam**

Word:

**madam**

Word:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **m** | **a** | **d** | **a** | **m** |
| **0** | **1** | **2** | **3** | **4** |

Word:

**food**

Word:

**doof**

|  |  |  |  |
| --- | --- | --- | --- |
| **f** | **o** | **o** | **d** |
| **0** | **1** | **2** | **3** |

**LEVEL: MODERATE (Self-Lab Revision Exercise)**

**Question 4**

Referring to the same case study and procedures, solve the issue using **Stack Linked List** implementation.

Word:

**madam**

**madam**

Word:

****