Department of ICT
Faculty of Technology
University of Ruhuna

Data Structures and Algorithms – ICT2113 **Level 2- Semester 1**

Laboratory Assignment 2 **| 2023**

Objectives

Main objective of this lab session is to get hands on experiences of Stack.

What is Stack

- It is type of **linear data structure**.
- It follows **LIFO** (Last In First Out) property.
- It has only one pointer **TOP** that points the last or top most element of Stack.
- Insertion and Deletion in stack can only be done from top only.
- Insertion in stack is also known as a **PUSH operation**.
- Deletion from stack is also known as **POP operation** in stack.

1. Following program is written to implement the menu-base stack using structure and array. Type the program and compile it. If you get some compilation errors, try to fix those as much as you can. Compile and run the program. Observe the output by entering different values.

```
#include<stdio.h>
#include<stdlib.h>
# define Max 5

struct stack
{
        int arr[Max];
        int top;
}st;

void display();

int main()
{
        int choice;
        st.top= -1;
        do {

          printf("\n\n\t 1. push an element into stack :- ");
```

```
            printf("\n\n\t 2. pop an element from stack :- ");
            printf("\n\n\t 3. display the elements of the  stack :- ");
            printf("\n\n\t 4. Exit from the program");
            printf("\n\n\t enter your choice:-  ") ;
            scanf("%d", choice);

            switch (choice)
            {
                    case 1:
                              push();
                              break;
                    case 2:
                              pop();

                    case 3:

                              break;
                    case 4:
                              exit(0);
                              break;
                    default:

                              printf("\n\n\t wrong entry try again");
                              break;
      }

          }while(choice!=4);


return 0;
}
void push()
{
          int item;
          if(st.top== Max -1)
            printf("\n\n\t the stack is full/overflow....");
          else
          {   printf("\n\n\t Enter the element to be pushed into to the stack");
              scanf("%d", &item);
              st.top++;
                    st.arr[st.top]=item;

          }
}
void pop()
{
        if (st.top==-1)
          printf("\n\n\t Stack is Empty/Underflow ......");
        else
        {
                int item;
                item= st.arr[st.top];
                st.top--;
                printf("\n\n\t  the pop elelmen is:-  %d",item);
          }
}
void display()
{
        if (st.top== -1)
            printf("\n\n\t The stack is empty ......");

        else
        {   printf("\n\n\t  The contents of the stack are......");
                for (int i=st.top; i>=0; i--)
                  printf("%d\t" , st.arr[i]);
         }
}
```

2. Implement the stack using structure and pointers.

   a) Declare the stack structure as given bellow

```
 #include<stdio.h>
 #define MAX 3


typedef struct
{
   int TOP;
   int ele[MAX];

}Stack;
```

   b) Initialize the stack

```
void init(Stack *s)
{
s->TOP = -1;  //(s->TOP  can be written as (*s).TOP)
}
```

   c) Check stack is full or not

```
int isFull(Stack *s)
```

```
{
if(s->TOP == MAX-1)
return 0;
else
  return -1;
}
```

d) Check stack is empty or not

```
int isEmpty(Stack *s)
{
    if(s->TOP == -1)
        return 0;
   else
    return -1;
}
```

e) Insert an element to the stack

```
void push(Stack *s, int item)
{
    if( !isFull(s) )
    {
        printf("\nStack is full");
        return;
    }
    s->TOP = s->TOP + 1;
    s->ele[s->TOP] = item;
}
```

f) Remove en element from the stack

```
int pop(Stack *s, int *item)
{
    if(!isEmpty(s))
    {
        printf("\nStack is empty");
        return -1;
    }
    *item = s->ele[s->TOP];
    s->TOP = s->TOP - 1;

    return 0;

}

int main()
{
    Stack s;
    int item;
init(&s);

push(&s,10);
push(&s,20);
push(&s,30);
 pop(&s,&item);
printf("\nPoped Item : %d",item);
 pop(&s,&item);
printf("\nPoped Item : %d",item);
 pop(&s,&item);
printf("\nPoped Item : %d",item);
 return 0;}
```

3. C Program to convert decimal to binary using an array based Stack.
   Decimal Number= 27   binaries=1101

4. Write a C program to reverse a stack data structure using recursion

Input Stack	Output Stack
2 <--- Top	9 <--- Top
4	8
8	4
9	2

5. C program to Reverse a String using an array based stack

   The logic behind to implement this program:
   - Read a string.
   - Push all characters until NULL is not found - Characters will be stored in stack variable.
   - Pop all characters until NULL is not found - As we know stack is a LIFO technique, so last character will be pushed first and finally we will get reversed string in a variable in which we store inputted string.