Mihir sahu

1nt21is093

8/12/22

Algorithm of. Stack implementation using array:

START

We begin with a class named Stack

Create a pointer top which we will use to carry out all operations

Initialize an array in which we'll be storing our data

Initialize a constructor as top = -1 indicating that stack is empty

Push() - function, we use this function to insert data into the stack, so first we check if top==full i.e stack is full and data cannot be inserted. Else increment the top pointer and insert the data.

Pop() - function, this function is used to remove data from the stack, first we check if top==-1 i.e if stack is empty nothing is there to delete. Else delete the element in top pointer and decrement it.

isEmpty() - function check whether stack is empty i.e to p== - 1 or top < 0

display() - function to display the contents of the stack, we iterate through the array from the beginning till we reach the top pointer.

END

Code:

#include <stdio.h>

int stack[100],i,j,choice=0,n,top=-1;

void push();

void pop();

void show();  //dedfining all the functions

void main ()

{

    printf("Enter the number of elements in the stack ");

    scanf("%d",&n);  //write the no of ele

    printf("\*\*\*\*\*\*\*\*\*Stack operations using array\*\*\*\*\*\*\*\*\*");

printf("\n----------------------------------------------\n");

    while(choice != 4)

    {

        printf("Chose one from the below options...\n");

        printf("\n1.Push\n2.Pop\n3.Show\n4.Exit");

        printf("\n Enter your choice \n");

        scanf("%d",&choice);  //asking user for choice

        switch(choice)

        {

            case 1:

            {

                push();  //for push operation

                break;

            }

            case 2:

            {

                pop();  //now for pop

                break;

            }

            case 3:

            {

                show();  //to show

                break;

            }

            case 4:

            {

                printf("Exiting....");

                break;

            }

            default:  //if none of the above

            {

                printf("Please Enter valid choice ");

            }

        };

    }

}

void push ()  // push case

{

    int val;

    if (top == n )   // checking the overflow condition

    printf("\n Overflow");

    else

    {

        printf("Enter the value?");  // ask the value to be pushed

        scanf("%d",&val);

        top = top +1;   // after pushing increase top by 1

        stack[top] = val;

    }

}

void pop ()

{

    if(top == -1)   // checking for underflow condition

    printf("Underflow");

    else

    top = top -1;   // now dec top by -1

}

void show()

{

    for (i=top;i>=0;i--)  // to show all the ele in stack

    {

        printf("%d\n",stack[i]);

    }

    if(top == -1)

    {

        printf("Stack is empty");  // check for empty condition

    }

}

O/P:



