

3. ARRAY IMPLEMENTATION OF STACK ADT

Preamble

A stack is a linear data structure where elements are stored in the LIFO (Last In First Out) principle where the last element inserted would be the first element to be deleted.

A stack is an Abstract Data Type (ADT) that is popularly used in most programming languages. It is named stack because it has the similar operations as the real-world stacks, for example: a pack of cards or a pile of plates etc.

Steps

Step 1: Define a stack size.

Step 2: Read the stack operation.

Step 3: Read the stack element.

Step 4: Check the stack operation is Push or Pop.

Step 5: If operation is push then check the stack status.

- i. If stack status is over flow we can't push the element in to the stack.
- ii. Otherwise we can add the data into stack
- iii. Move top to next position.

Implementation in C

```
#include<stdio.h>

int stack[100],choice,n,top,x,i;

void push(void);
void pop(void);
void display(void);

int main()
{
    //clrscr();

    top=-1;

    printf("\n Enter the size of STACK[MAX=100]:");
    scanf("%d",&n);

    printf("\n\t STACK OPERATIONS USING ARRAY");
```

```

printf("\n\t-----");
printf("\n\t 1.PUSH\n\t 2.POP\n\t 3.DISPLAY\n\t 4.EXIT");
do
{
    printf("\n Enter the Choice:");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1:
        {
            push();
            break;
        }
        case 2:
        {
            pop();
            break;
        }
        case 3:
        {
            display();
            break;
        }
        case 4:
        {
            printf("\n\t EXIT POINT ");
            break;
        }
        default:
        {
            printf ("\n\t Please Enter a Valid Choice(1/2/3/4)");
        }
    }
}

```

```

        }
    }while(choice!=4);
    return 0;
}

void push()
{
    if(top>=n-1)
    {
        printf("\n\tSTACK is over flow");
    }
    else
    {
        printf(" Enter a value to be pushed:");
        scanf("%d",&x);
        top++;
        stack[top]=x;
    }
}

void pop()
{
    if(top<=-1)
    {
        printf("\n\t Stack is under flow");
    }
    else
    {
        printf("\n\t The popped elements is %d",stack[top]);
        top--;
    }
}

void display()
{

```

```

        if(top>=0)
        {
            printf("\n The elements in STACK \n");
            for(i=top; i>=0; i--)
                printf("\n%d",stack[i]);
            printf("\n Press Next Choice");
        }
        else
        {
            printf("\n The STACK is empty");
        }
    }
}

```

Sample Input and Output

```

STACK OPERATIONS USING ARRAY
-----
1.PUSH
2.POP
3.DISPLAY
4.EXIT
Enter the Choice:1
Enter a value to be pushed:10

Enter the Choice:1
Enter a value to be pushed:34

Enter the Choice:1
Enter a value to be pushed:25

Enter the Choice:1
Enter a value to be pushed:90

Enter the Choice:1
Enter a value to be pushed:90

Enter the Choice:3

The elements in STACK

90
90
25
34
10
Press Next Choice
Enter the Choice:2

The popped elements is 90
Enter the Choice:2

The popped elements is 90
Enter the Choice:3

The elements in STACK

25
34
10
Press Next Choice
Enter the Choice:

```