Experiment No. 1

Aim: Implementation of stack using array for real world application.

Objective: 1. To introduce the concept of data structures and analysis procedure.
2. To conceptualize linear data structures and its implementation for various real-world application.

Theory:

1) Introduction To Data Structure:

Data structure can be defined as the group of data elements which provides an efficient way of storing and organising data in the computer so that it can be used efficiently, some examples of Data structure are arrays, linked list, stack, queue, etc. Data structures are widely used in almost array aspect of computer science is operating system, compiler Design, Artifical Intelligence, Graphics and many more.

Classification of Data Structure
- Primitive data structure
- Non-Primitive data structure

2) Introduction To stack:

Stack is a linear list in which insertion and deletions are allowed only at one end, called top. A stack is an abstract data type (ADT), can be implemented in most of the programming languages. It is named

as stack because it behaves like a real-world stack, for example - piles of plates or deck of cards etc.

3 Various Operations. \ PUSH, POP, PEEP, CHANGE, DISPLAYS etc]

• PUSH

PUSH operation refers to inserting an element in the stack Since there's only one position at which the new element can be inserted - Top of stack, the new element is inserted at the top of the stack.

• POP

POP operation refers to ~~inserting~~ the removal of an element in the stack. Since there's only we have to access the element at the top of the stack, Note:- We can also choose to return the value of popped element stack, its completely at the choice of the programmers to implement this.

• PEEP

PEEP operation allows the user to see the element on the top of the stack. The stack is not modified in any manner in this operation.

• CHANGE

User can change the content of the specific elements

• DISPLAYS

The displays () function displays all the elements in the stack. It uses a for loop to do so. Ip

there are no elements in the stack, then stack is empty is printed.

4 Algorithm:-

1. PUSH (S, TOP, X). This procedure inserts the element X to the top of a stack which is represented by a vector S containing N elements with a pointer TOP denoting the top element in the stack.

i Checks for stack overflow

If TOP > N
then write ('stack overflow')
Return

ii Increment TOP

TOP ← TOP + 1

iii Insert element
S[TOP] ← X

iv Finished
Return

2. POP(S, TOP). This function removes the top element from a stack which is represented by a vector S and return this element. TOP is a pointer to the top element of the stack

i Check for underflow on stack

If top = 0
then Write ('Stack underflow on POP') take action in response to underflow.
Exit

ii) Decrement Pointer
TOP ← TOP - 1

iii) Return former top element of stack.
Return (S[TOP + 1])

3. PEEP(S, TOP, i), Give a vector S (consisting of N elements) representing a sequentially allocated stack, and a pointer TOP denoting the TOP element of the stack, this function returns the value of the ith element from the TOP of the stack. The element is not deleted by this function.

i) Checks for stack underflow
If TOP - i + 1 ≤ 0
then Write ('Stack underflow on PEEP'); take action in response to underflow
Exit

ii) Return i^{th} element from top of stack'
Return (S[TOP - i + 1])

Consider stack S with N = 5

Index of 1st Element from TOP of stack : TOP TOP-1+1 = TOP
Index of 2nd Element from TOP of stack : TOP-1 TOP-2+1 = TOP-1
Index of 3rd Element from TOP of stack : TOP-2 TOP-3+1 = TOP-2
Index of 4th Element from TOP of stack : TOP-3 TOP-4+1 = TOP-3
Index of 5th Element from TOP of stack : TOP(n-1) = TOP-n+1

__/__/____

Conclusion : Thus, we have learnt how to implement the operation of stack (which are PUSH, POP, PEEP using array stack is one of the most important data structure. In stack element can be added or removed from a stack at only on one end (stack follows 'LIFO' structure) We use operators "PUSH" for adding element and "POP" for removing the element Thus we have noticed that the element popped from a stack is always the last one pushed onto it Therefore stack is refer as LIFO list.

5 Examples :

- The stack of trays in cafeteria.
- A stack of plates in cupboard.
- A deriveway that is only one car wide

PROGRAM FOR STACK :[File name: STACKNEW.C]

```
≡ File Edit Search Run Compile Debug Project Options Window Help
[■]                                STACKNEW.C                                5-[↕]
#include <stdio.h>
#include <conio.h>
int STK[100], TOP = -1, i , n, x, choice;
void Push();
void Pop();
void Peep();
void Change();
void Display();

void main()
{
clrscr();
printf("\t Welcome to Implementation of STACK using Array!! \n");
printf("Enter the size of Stack (Maximum size = 100): ");
scanf("%d",&n);

do
{
printf(" \n Stack Operation available: \n");
printf("\t1.Push \t2.Pop \t3.Peep \t4.Display \t5.Change \t6.Exit \n");
printf("\n Enter your Choice: ");
scanf("%d", &choice);
switch(choice)
{
case 1:
     Push();
     break;
case 2:
     Pop();
     break;
case 3:
     Peep();
     break;
case 4:
     Display();
     break;
case 5:
     Change();
     break;
case 6:
     printf("Exit: Program Finished!! ");
     break;
```

```
default:
     printf("Please enter a valid choice: 1,2,3,4,5,6 \n");
}
}
while (choice != 6);
}


// Function to perform PUSH operation
void Push()
{
if (TOP >= n - 1)
{
printf("Stack Overflow \n");
}
else
{
printf(" Enter the element to be pushed: ");
scanf("%d", &x);
TOP++;
STK[TOP] = x;
```

```
}
}

// Function to perform POP element
void Pop()
{
 if (TOP < 0)
 {
 printf("Stack Underflow \n");
 }
 else
 {
 printf("The Popped element is: %d \n", STK[TOP]);
 TOP--;
 }
 }

// Function to perform Peep Operation
void Peep()
{
printf("Enter the Position of the element from the top which you want to peep:
```

```
scanf("%d", &i);
if(TOP - i + 1 < 0 )
{
printf(" Stack underflow on Peep \n");
}
else
{
printf("The %d element from the top is: %d \n",i,STK[TOP - i + 1]);
}
}


// Function to Change the element in the stack.
void Change()
{

  int v1,v2;
  printf("\nEnter Position for change : ");
  scanf("%d",&v1);
  printf("\nEnter the Number for change : ");
  scanf("%d",&v2);
  if(TOP-v1<=-1)
  {
     printf("\nSTACK is overflow !!!");
  }
  else
  {
    STK[TOP-v1]=v2;
    printf("\n Changed Successfull !!!");
  }

}

// Function to Displayy the Stack
void Display()
{
if (TOP < 0)
{
printf("Stact is empty \n");
}
else
{
printf("The elements in the stack are:");
for (i= TOP; i > -1; i--)
{
printf("\n %d \n", STK[i]);
}
}
}
134:73
F1 Help  Alt-F8 Next Msg  Alt-F7 Prev Msg  Alt-F9 Compile  F9 Make  F10 Menu
```

OUTPUT:

1. Underflow (Stack is Empty, So no POP operation will be perform)

```
          Welcome to Imoplementation of STACK using Array!!
 Emter the size of Stack (Maximum size = 100): 5

  Stack Operation avaiable:
         1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

  Enter your Choice: 2
 Stack Underflow

  Stack Operation avaiable:
         1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

  Enter your Choice: _
```

2. Stack is Empty.

```
 Stack Operation avaiable:
        1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

 Enter your Choice: 4
Stact is empty

 Stack Operation avaiable:
        1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

 Enter your Choice: _
```

3. PUSH operation values are 10,20,30,40,50

```
Enter your Choice:  1
Enter the element to be pushed: 10

Stack Operation avaiable:
       1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

Enter your Choice: 1
Enter the element to be pushed: 20

Stack Operation avaiable:
       1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

Enter your Choice: 1
Enter the element to be pushed: 30

Stack Operation avaiable:
       1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

Enter your Choice: 1
Enter the element to be pushed: 40
```

```
Stack Operation avaiable:
       1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

Enter your Choice:    1
Enter the element to be pushed: 50

Stack Operation avaiable:
       1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

Enter your Choice: 4
The elements in the stack are:
50

40

30

20

10

Stack Operation avaiable:
       1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

Enter your Choice: _
```

4. Overflow (Stack size is 5)

```
 Stack Operation avaiable:
        1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

 Enter your Choice: 4
The elements in the stack are:
 50

 40

 30

 20

 10

 Stack Operation avaiable:
        1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

 Enter your Choice: 1
Stack Overflow

 Stack Operation avaiable:
        1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

 Enter your Choice:
```

5. Deleting the value throught (POP) operation.(Value deleted are 50, 40)

```
 Enter your Choice: 2
The Popped element is: 50

 Stack Operation avaiable:
        1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

 Enter your Choice: 2
The Popped element is: 40

 Stack Operation avaiable:
        1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

 Enter your Choice: 4
The elements in the stack are:
 30

 20

 10

 Stack Operation avaiable:
        1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

 Enter your Choice: _
```

6. Performing the Peep Operation.(PEEP)

```
 Stack Operation avaiable:
        1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

 Enter your Choice: 4
The elements in the stack are:
 30

 20

 10

 Stack Operation avaiable:
        1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

 Enter your Choice: 3
Enter the Position of the element from the top which you want to peep:2
The 2 element from the top is: 20
```

7. Change Operation

```
 Stack Operation avaiable:
        1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

 Enter your Choice: 4
The elements in the stack are:
 30

 20

 10

 Stack Operation avaiable:
        1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

 Enter your Choice: 5

Enter Position for change : 1

Eneter the Number for change : 99

 Changed Successfull !!!
 Stack Operation avaiable:
        1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

 Enter your Choice: _
```

```
 Stack Operation avaiable:
        1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

 Enter your Choice: 5

Enter Position for change : 1

Eneter the Number for change : 99

 Changed Successfull !!!
 Stack Operation avaiable:
        1.Push  2.Pop   3.Peep  4.Display      5.Change       6.Exit

 Enter your Choice: 4
The elements in the stack are:
 30

 99

 10
```