**AIM:**

To implement a stack using Array and linked List and execute the following operation on stack.

(i) Push an element into a stack

(ii) Pop an element from a stack

(iii) Return the Top most element from a stack

(iv) Display the elements in a stack

**ARRAY IMPLEMENTATION OF STACK**

**ALGORITHM:**

**STACK FULL :**

IsFull()

**Step 1 :** Start.

**Step 2 :** If top = MAX - 1 goto Step 3 else goto Step 4.

**Step 3 :** Return 1 and Stop.

**Step 4 :** Return 0.

**Step 5 :** Stop.

**STACK EMPTY :**

IsEmpty()

**Step 1 :** Start.

**Step 2 :** If top = -1 goto Step 3 else goto Step 4.

**Step 3 :** Return 1 and Stop.

**Step 4 :** Return 0.

**Step 5 :** Stop.

**PUSH :**

Push(ele)

**Step 1 :** Start.

**Step 2 :** If IsFull() = True goto Step 3 else goto Step 4.

**Step 3 :** Display message “Stack Overflow…!” and goto Step 6.

**Step 4 :** Set Top = Top + 1.

**Step 5 :** Set Stack[Top] = ele.

**Step 6 :** Stop.

**POP :**

Pop()

**Step 1 :** Start.

**Step 2 :** If IsEmpty() = True goto Step 3 else goto Step 4.

**Step 3 :** Display message “Stack Underflow…!” and goto Step 6.

**Step 4 :** Display Stack[Top].

**Step 5 :** Set Top = Top – 1.

**Step 6 :** Stop.

**PEEK**

Top()

**Step 1 :** Start.

**Step 2 :** If IsEmpty() = True goto Step 3 else goto Step 4.

**Step 3 :** Display message “Stack Underflow…!” and goto Step 5.

**Step 4 :** Display Stack[Top].

**Step 5 :** Stop.

**DISPLAY**

Display()

**Step 1 :** Start.

**Step 2 :** If IsEmpty() = True goto Step 3 else goto Step 4.

**Step 3 :** Display message “Stack Underflow…!” and goto Step 8.

**Step 4 :** Set i = top.

**Step 5 :** Repeat Steps 6 to 7 while i >= 0.

**Step 6 :** Display Stack[i].

**Step 7 :** Set i = i – 1.

**Step 8 :** Stop.

**PROGRAM**

#include <stdio.h>

#define MAX 5

int Stack[MAX], top = -1;

int IsFull();

int IsEmpty();

void Push(int ele);

void Pop();

void Top();

void Display();

int main()

{

int ch, e;

do

{

printf("1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT");

printf("\nEnter your choice : ");

scanf("%d", &ch);

switch(ch)

{

case 1:

printf("Enter the element : ");

scanf("%d", &e);

Push(e);

break;

case 2:

Pop();

break;

case 3:

Top();

break;

case 4:

Display();

break;

}

} while(ch <= 4);

return 0;

}

int IsFull()

{

if(top == MAX - 1)

return 1;

else

return 0;

}

int IsEmpty()

{

if(top == -1)

return 1;

else

return 0;

}

void Push(int ele)

{

if(IsFull())

printf("Stack Overflow...!\n");

else

{

top = top + 1;

Stack[top] = ele;

}

}

void Pop()

{

If(IsEmpty())

printf("Stack Underflow...!\n");

else

{

printf("%d\n", Stack[top]);

top = top - 1;

}

}

void Top()

{

if(IsEmpty())

printf("Stack Underflow...!\n");

else

printf("%d\n", Stack[top]);

}

void Display()

{

int i;

if(IsEmpty())

printf("Stack Underflow...!\n");

else

{

for(i = top; i >= 0; i--)

printf("%d\t", Stack[i]);

printf("\n");

}

}

**OUTPUT:**

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 1

Enter the element : 10

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 1

Enter the element : 20

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 1

Enter the element : 30

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 1

Enter the element : 40

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 1

Enter the element : 50

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 1

Enter the element : 60

Stack Overflow...!

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 4

50 40 30 20 10

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 3

50

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 2

50

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 2

40

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 2 B.BHUVANESWARAN | AP (SG) | CSE | Rajalakshmi Engineering College

13

30

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 2

20

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 2

10

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 2

Stack Underflow...!

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 5

**LINKED LIST IMPLEMENTATION OF STACK**

**ALGORITHM:**

**EMPTY LIST**

IsEmpty()

**Step 1 :** Start.

**Step 2 :** If List = NULL goto Step 3 else goto Step 4.

**Step 3 :** Return 1 and Stop.

**Step 4 :** Return 0 and Stop.

**Step 5 :** Stop.

**PUSH**

Push(e)

**Step 1 :** Start.

**Step 2 :** Set NewNode = addressof(Stack).

**Step 3 :** Set NewNodeElement = e.

**Step 4 :** If IsEmpty() = True, then goto Step 5 else goto Step 6.

**Step 5 :** Set NewNodeNext = NULL and goto Step 7.

**Step 6 :** Set NewNodeNext = List.

**Step 7 :** Set List = NewNode.

**Step 8 :** Stop.

**POP**

Pop()

**Step 1 :** Start.

**Step 2 :** If IsEmpty() = True, then goto Step 3 else goto Step 4.

**Step 3 :** Display “Stack is Underflow…!” and goto Step 8.

**Step 4 :** Set TempNode = List.

**Step 5 :** Set List = ListNext.

**Step 6 :** Display the TempNodeElement.

**Step 7 :** Delete TempNode.

**Step 8 :** Stop.

**TOP**

Top(List)

**Step 1 :** Start.

**Step 2 :** If IsEmpty() = True, then goto Step 3 else goto Step 4.

**Step 3 :** Display “Stack is Underflow…!” and goto Step 5.

**Step 4 :** Display the List Element.

**Step 5 :** Stop.

**DISPLAY**

Display()

**Step 1 :** Start.

**Step 2 :** If IsEmpty() = True goto Step 3 else goto Step 4.

**Step 3 :** Display “Stack is Underflow…!” and goto Step 8.

**Step 4 :** Set Position = List.

**Step 5 :** Repeat the Steps 6-7 until Position != NULL.

**Step 6 :** Display PositionElement.

**Step 7 :** Set Position = PositionNext.

**Step 8 :** Stop.

**PROGRAM**

/\* Implementation of stack using linked list - STACKLL.C \*/

#include <stdio.h>

#include <stdlib.h>

struct node

{

int Element;

struct node \*Next;

}\*List = NULL;

typedef struct node Stack;

int IsEmpty();

void Push(int e);

void Pop();

void Top();

void Display();

int main()

{

int ch, e;

do

{

printf("1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT");

printf("\nEnter your choice : ");

scanf("%d", &ch);

switch(ch)

{

case 1:

printf("Enter the element : ");

scanf("%d", &e);

Push(e);

break;

case 2:

Pop();

break;

case 3:

Top();

break;

case 4:

Display();

break;

}

} while(ch <= 4);

return 0;

}

int IsEmpty()

{

if(List == NULL)

return 1;

else

return 0;

}

void Push(int e)

{

Stack \*NewNode = malloc(sizeof(Stack));

NewNode->Element = e;

if(IsEmpty())

NewNode->Next = NULL;

else

NewNode->Next = List;

List = NewNode;

void Pop()

{

if(IsEmpty())

printf("Stack is Underflow...!\n");

else

{

Stack \*TempNode;

TempNode = List;

List = List->Next;

printf("%d\n", TempNode->Element);

free(TempNode);

}

}

void Top()

{

if(IsEmpty())

printf("Stack is Underflow...!\n");

else

printf("%d\n", List->Element);

}

void Display()

{

If(IsEmpty())

printf("Stack is Underflow...!\n");

else

{

Stack \*Position;

Position = List;

while(Position != NULL)

{

printf("%d\t", Position->Element);

Position = Position->Next;

}

printf("\n");

}

}

**OUTPUT:**

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 1

Enter the element : 10

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 1

Enter the element : 20

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 1

Enter the element : 30

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT 22

B.BHUVANESWARAN | AP (SG) | CSE | Rajalakshmi Engineering College

Enter your choice : 1

Enter the element : 40

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 1

Enter the element : 50

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 4

50 40 30 20 10

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 3

50

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 2

50

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 2

40

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 2

30

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 2

20

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 2

10

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 2

Stack is Underflow...!

1.PUSH 2.POP 3.TOP 4.DISPLAY 5.EXIT

Enter your choice : 5

**RESULT:**

Hence, implemented stack structures with arrays and linked lists