

4(a).Implementation of Stack using Array

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

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

4(b).Implementation of Stack using Linked List

Program :

```
#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
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
```