Lecture 8 - Implementation of Stack

Implementation of stack using an array.

```
#include<stdio.h>
#include<stdlib.h>

int n, top = -1, *stack;

void push(int x){
        if(top==n) return;
        stack[++top]=x;
}

int pop(){
        if(top==-1) return -1;
        return stack[top--];
}

int peek(){
        if(top==-1) return -1;
        return stack[top];
}

void display(){
        for(int i=top ; i>-1 ; i--) printf("%d ",stack[i]);
        printf("\n\n");
}

int main(){

        n = 10;

        printf("Initializing the stack with size 10\n\n");

        stack = (int*)malloc(n*sizeof(int));

        printf("Pushing elements into the stack\n1\n2\n3\n\n");

        push(1);
        push(2);
        push(3);
```

```
        printf("Displaying elements of the stack -\n");
        display();
        printf("The top of the stack = %d\n\n",peek());
        printf("Pop the top of the stack = %d\n\n",pop());
        printf("Pop the top of the stack = %d\n\n",pop());
        printf("Displaying elements of the stack -\n");
        display();
        return 0;
}
```

Output

```
Initializing the stack with size 10

Pushing elements into the stack
1
2
3

Displaying elements of the stack -
3 2 1

The top of the stack = 3

Pop the top of the stack = 3

Pop the top of the stack = 2

Displaying elements of the stack -
1
```

Exercise

1. **Evaluation of Postfix Expressions Using Stack.**
2. **C Program to Convert Infix to Postfix Expression using Stack**