Data Structures

Experiment no. 2

Develop code to implement stack Application (Infix to Postfix Conversion) & Postfix Expression Evaluation using Stack

Q. WAP in C to convert Infix expression to Postfix expression.

Code:

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
#define MAX 100

char stack[MAX];
int tos=-1;

void push(char ch);
char pop();
int precedence(char ch);

void main()
{
        char infix[MAX];
        char ch;
        int i=0;
        clrscr();
        printf("\nEnter Infix Expression : ");
        fflush(stdin);
        gets(infix);

        printf("\nPostfix Expression : ");

        while(infix[i] != '\0')
        {
                if(isalnum(infix[i]))
                        printf("%c ",infix[i]);
                else if(infix[i]=='(')
                        push(infix[i]);
                else if(infix[i]==')')
                {
                        while((ch=pop())!='(')
                        {
```

```
                                        printf("%c ",ch);
                                }
                        }
                        else
                        {
                                while(precedence(stack[tos]) >= precedence(infix[i]))
                                {
                                        printf("%c ",pop());
                                }
                                push(infix[i]);
                        }
                        i++;
                }

                while(tos != -1)
                {
                        printf("%c ",pop());
                }

                getch();
}

void push(char ch)
{
                tos++;
                stack[tos]=ch;
}

char pop()
{
                if(stack[tos]==-1)
                        return -1;
                else
                        return stack[tos--];
}

int precedence(char ch)
{
                if(ch=='(')
                        return 0;
                else if(ch=='+' || ch=='-')
                        return 1;
                else if(ch=='*' || ch=='/')
                        return 2;
                return 0;
}
```

Output:

```
Enter Infix Expression : (a/b+(c*d/e)-(f/g)*h)

Postfix Expression : a b / c d * e / + f g / h * -
```

Q. WAP in C to evaluate postfix expression.

Code:

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
#define MAX 100

int stack[MAX];
int tos=-1;

void push(int n);
int pop();

void main()
{
	char postfix[MAX];
	int i=0,a,b;
	clrscr();

	printf("\nEnter a Postfix expression to evaluate : ");
	fflush(stdin);
	gets(postfix);

	while(postfix[i] != '\0')
	{
		if(isdigit(postfix[i]))
		{
			push(postfix[i]-'0');
		}
		else
		{
			a=pop();
			tos--;
			b=pop();
			tos--;
			switch(postfix[i])
			{
				case '+':
					push(b+a);
					break;

				case '-':
					push(b-a);
					break;

				case '*':
```

```
                                        push(b*a);
                                        break;

                                case '/':
                                        push(b/a);
                                        break;
                        }
                }
                i++;
        }

        printf("\nValue of the entered postfix expression is : %d",pop());
        printf("\n\n\t %s  =  %d\n",postfix,pop());

        getch();
}

void push(int n)
{
        tos++;
        stack[tos]=n;
}

int pop()
{
        return (stack[tos]);
}
```

Output:

```
Enter a Postfix expression to evaluate : 6523+8*+3+*

Value of the entered postfix expression is : 288

          6523+8*+3+*  =  288
```