NAME: Arnav Thakare
Section and roll no.: B-29

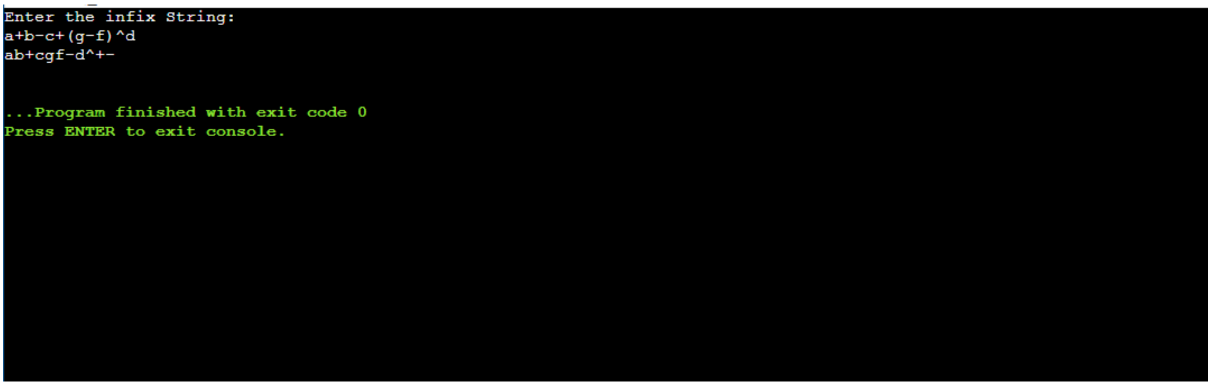
INFIX TO POSTFIX CONVERSION

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct Stack{
    char stack[100];
    int top;
};
void push(struct Stack *s,char x){
    if(s->top==99){
    return;
    }
    s->top++;
    s->stack[s->top]=x;
}
void peep(struct Stack *s){
    if(s->top==-1){
            return;
    }
    s->top--;
}
char pop(struct Stack *s){
    char temp=s->stack[s->top];
    s->top--;
    return temp;
}
int precedence(char c){
    switch(c){

    case '^':
            return 6;
    case '*':
            return 5;
    case '/':
            return 4;
    case '+':
            return 3;
    case '-':
            return 2;
    }
}
```

```
void infixtoPostfix(char *infix,struct Stack *s){
    char postfix[30];
    int i=0;
    int j=0;
    while(infix[i]!='\0'){
            if((infix[i]>='a' && infix[i]<='z') || (infix[i]>='A' && infix[i]<='Z')){
                        postfix[j]=infix[i];
                        j++;
                        i++;
            }
            else if(infix[i]=='('){
                 push(s,infix[i]);
                 i++;
            }
            else if(infix[i]==')'){
                        while(s->stack[s->top]!='('){
                                    postfix[j]=pop(s);
                                    j++;
                        }
                        peep(s);
                        i++;

            }
            else if(infix[i]=='+' || infix[i]=='-' || infix[i]=='*' || infix[i]=='^' || infix[i]=='/'){
                        if(s->top==-1 || s->stack[s->top]=='('){
                        push(s,infix[i]);
                        i++;
                        }
                        else{
                                    while((s->top!=-1) && (s->stack[s->top]!='(') &&
(precedence(infix[i])<=precedence(s->stack[s->top]))){
                                    postfix[j]=pop(s);
                                    j++;
                                    }
                                    push(s,infix[i]);
                                    i++;
                        }
            }
    }
    while(s->top!=-1){
            postfix[j]=pop(s);
            j++;
    }
    for(int i=0;i<j;i++){
            printf("%c",postfix[i]);
    }
    printf("\n");
}
```

```
int main(){
   struct Stack *s;
   s=(struct Stack *)malloc(sizeof(struct Stack));
   s->top=-1;
   char infix[30];
   printf("Enter the infix String:\n");
   scanf("%s",infix);
   infixtoPostfix(infix,s);
return 0;
}
```

OUTPUT:



```
Enter the infix String:
a+b-c+(g-f)^d
ab+cgf-d^+-


...Program finished with exit code 0
Press ENTER to exit console.
```