```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
# define SIZE 25
struct Stack {
   int top;
   int arr[SIZE];
};
void push(struct Stack *s, int ss);
int pop(struct Stack *s);
void main()
{
   struct Stack s;
   s.top = 0;
   char aa[SIZE];
   printf("Enter postfix: ");
   scanf("%s", aa);
   int length = strlen(aa);
   int dc = 0;
   for(int i = 0; i<length; i++){
      if (isdigit(aa[i])){
      dc++;
   }
   }
   if (length-dc != dc-1){
      printf("ERROR");
      return;
   }
```

```
    for(int i = 0; i<length;i++){
        if (isdigit(aa[i])){
            push(&s, aa[i] - '0');
        } else {
            if (s.top < 1){
                printf("ERROR");
                return;
            }
            int t1 = pop(&s);
            int t2 = pop(&s);
            if (aa[i] == '+'){
                push(&s, (t1+t2));
            } else if (aa[i] == '-') {
                push(&s, (t2-t1));
            } else if (aa[i] == '*') {
                push(&s, (t2*t1));
            } else if (aa[i] == '/') {
                push(&s, (t2/t1));
            }
        }
    }
    printf("Result = %d", pop(&s));
}
void push(struct Stack *s, int ss)
{
    // Checks if the stack is full
    if (s->top < SIZE)
    {
```

```
      s->arr[s->top] = ss;
      s->top++;
   }
   else
   {
      printf("STACK IS FULL\n");
   }
}
int pop(struct Stack *s)
{
   if (s->top > 0)
   {
      s->top--;
      return s->arr[s->top];
   }
   else
   {
      printf("NO ELEMENTS IN STACK\n");
      return -1;
   }
}
```