```c
#include <stdio.h>
#include <ctype.h>
# define SIZE 25

struct Stack {
    int top;
    int arr[SIZE];
};

void push(struct Stack *s, int ss);
int pop(struct Stack *s);
void display(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){
```

```c
        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));

}


/***
```

```c
   Adds elements to the top of the stack
***/
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");
   }
}


/***
Remove the last element from the stack
***/
int pop(struct Stack *s)
{
   //Checks if the stack is not empty
   if (s->top > 0)
   {
      return s->arr[s->top--];
   }
   else
   {
```

```c
        printf("NO ELEMENTS IN STACK\n");

    }

}
/***

Prints all the elements in the stack

***/

void display(struct Stack *s)

{
    // Checks if the stack is empty

    if (s-> top == 0)

    {
        printf("NO ELEMENTS IN STACK\n");

        return;

    }
    for (int j = 0; j<s->top; j++)

    {
        printf("%d, " , s->arr[j]);

    }
    printf("\n");

}
```
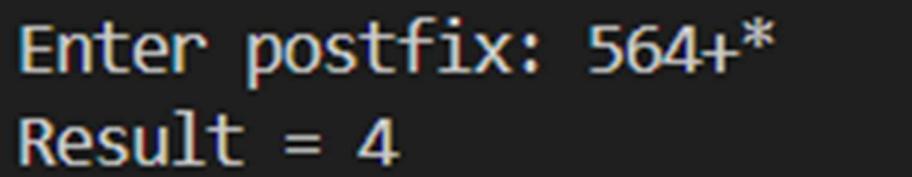
```
Enter postfix: 564+*
Result = 4
```