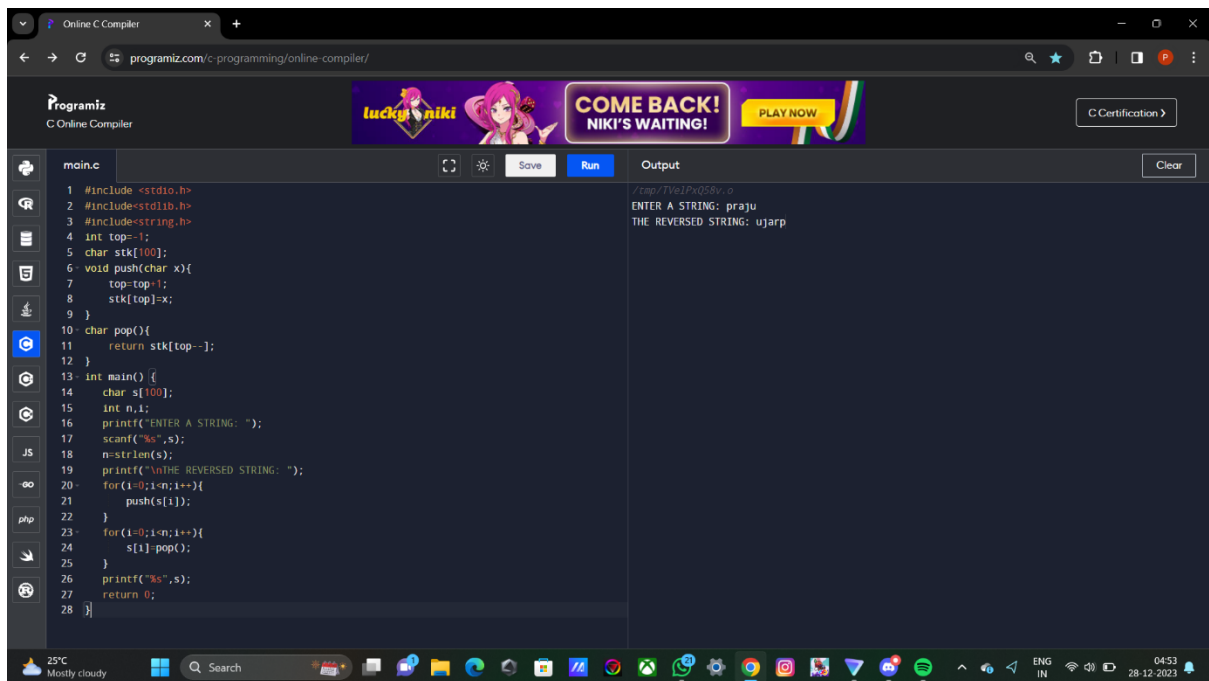


## 1.STRING REVERSAL USING STACK



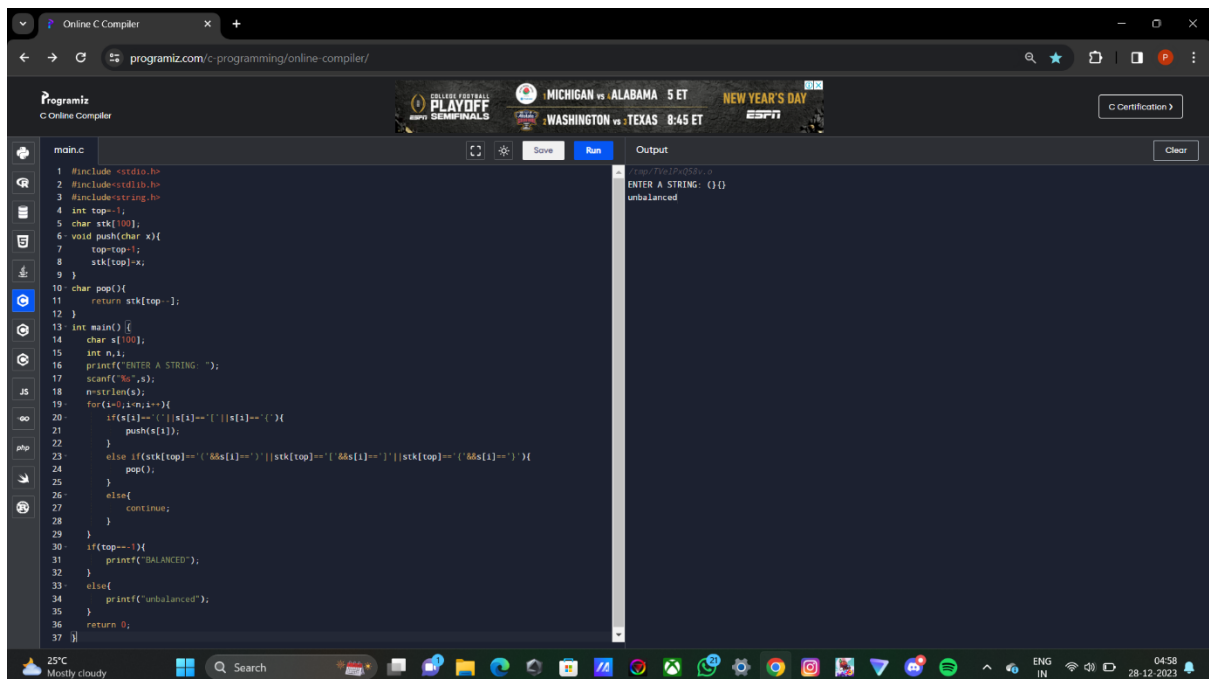
The screenshot shows a web browser window with the URL `programiz.com/c-programming/online-compiler/`. The page features a banner for 'Programiz C Online Compiler' and a 'COME BACK! NIKI'S WAITING!' message. The main area displays a C program for string reversal using a stack. The code is as follows:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 int top=-1;
5 char stk[100];
6 void push(char x){
7     top=top+1;
8     stk[top]=x;
9 }
10 char pop(){
11     return stk[top--];
12 }
13 int main() {
14     char s[100];
15     int n,i;
16     printf("ENTER A STRING: ");
17     scanf("%s",s);
18     n=strlen(s);
19     printf("\nTHE REVERSED STRING: ");
20     for(i=0;i<n;i++){
21         push(s[i]);
22     }
23     for(i=0;i<n;i++){
24         s[i]=pop();
25     }
26     printf("%s",s);
27     return 0;
28 }
```

The output of the program is shown on the right:

```
ENTER A STRING: praju
THE REVERSED STRING: ujarp
```

## 2. BALANCING BRACKETS USING STACK



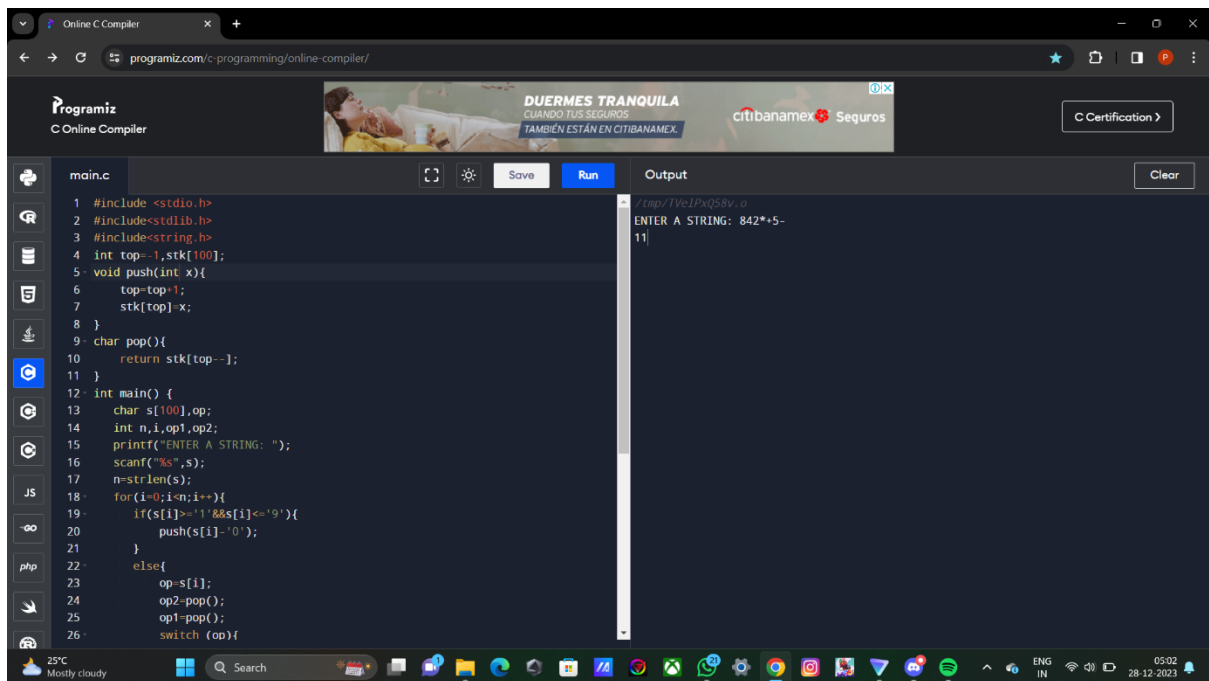
The screenshot shows the same online C compiler interface. The C program for balancing brackets using a stack is as follows:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 int top=-1;
5 char stk[100];
6 void push(char x){
7     top=top+1;
8     stk[top]=x;
9 }
10 char pop(){
11     return stk[top--];
12 }
13 int main() {
14     char s[100];
15     int n,i;
16     printf("ENTER A STRING: ");
17     scanf("%s",s);
18     n=strlen(s);
19     for(i=0;i<n;i++){
20         if(s[i]=='{'||s[i]=='['||s[i]=='('){
21             push(s[i]);
22         }
23         else if(stk[top]=='{'&&s[i]=='}'||stk[top]=='['&&s[i]==']'||stk[top]=='('&&s[i]==')'){
24             pop();
25         }
26         else{
27             continue;
28         }
29     }
30     if(top==0){
31         printf("BALANCED");
32     }
33     else{
34         printf("unbalanced");
35     }
36     return 0;
37 }
```

The output of the program is shown on the right:

```
ENTER A STRING: {}{}
unbalanced
```

### 3. POSTFIX EXPRESSION EVALUATION

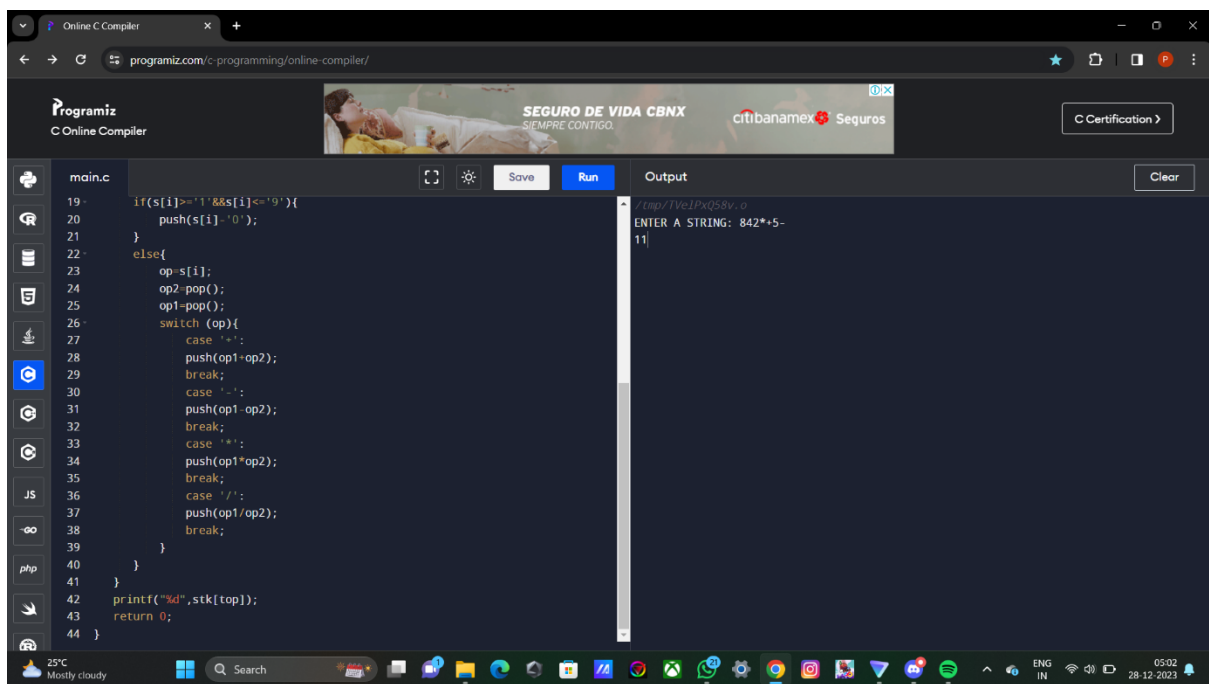


The screenshot shows the Programiz Online C Compiler interface. The code in `main.c` is as follows:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 int top=-1,stk[100];
5 void push(int x){
6     top=top+1;
7     stk[top]=x;
8 }
9 char pop(){
10     return stk[top--];
11 }
12 int main() {
13     char s[100],op;
14     int n,i,op1,op2;
15     printf("ENTER A STRING: ");
16     scanf("%s",s);
17     n=strlen(s);
18     for(i=0;i<n;i++){
19         if(s[i]>='1'&&s[i]<='9'){
20             push(s[i]-'0');
21         }
22         else{
23             op=s[i];
24             op2=pop();
25             op1=pop();
26             switch (op){
```

The output window shows the following text:

```
/tmp/IVeIPxQ58v.o
ENTER A STRING: 842*+5-
11
```



The screenshot shows the continuation of the C program in the Programiz Online C Compiler. The code in `main.c` is as follows:

```
27         case '+':
28             push(op1+op2);
29             break;
30         case '-':
31             push(op1-op2);
32             break;
33         case '*':
34             push(op1*op2);
35             break;
36         case '/':
37             push(op1/op2);
38             break;
39     }
40 }
41 }
42 printf("%d",stk[top]);
43 return 0;
44 }
```

The output window shows the following text:

```
/tmp/IVeIPxQ58v.o
ENTER A STRING: 842*+5-
11
```