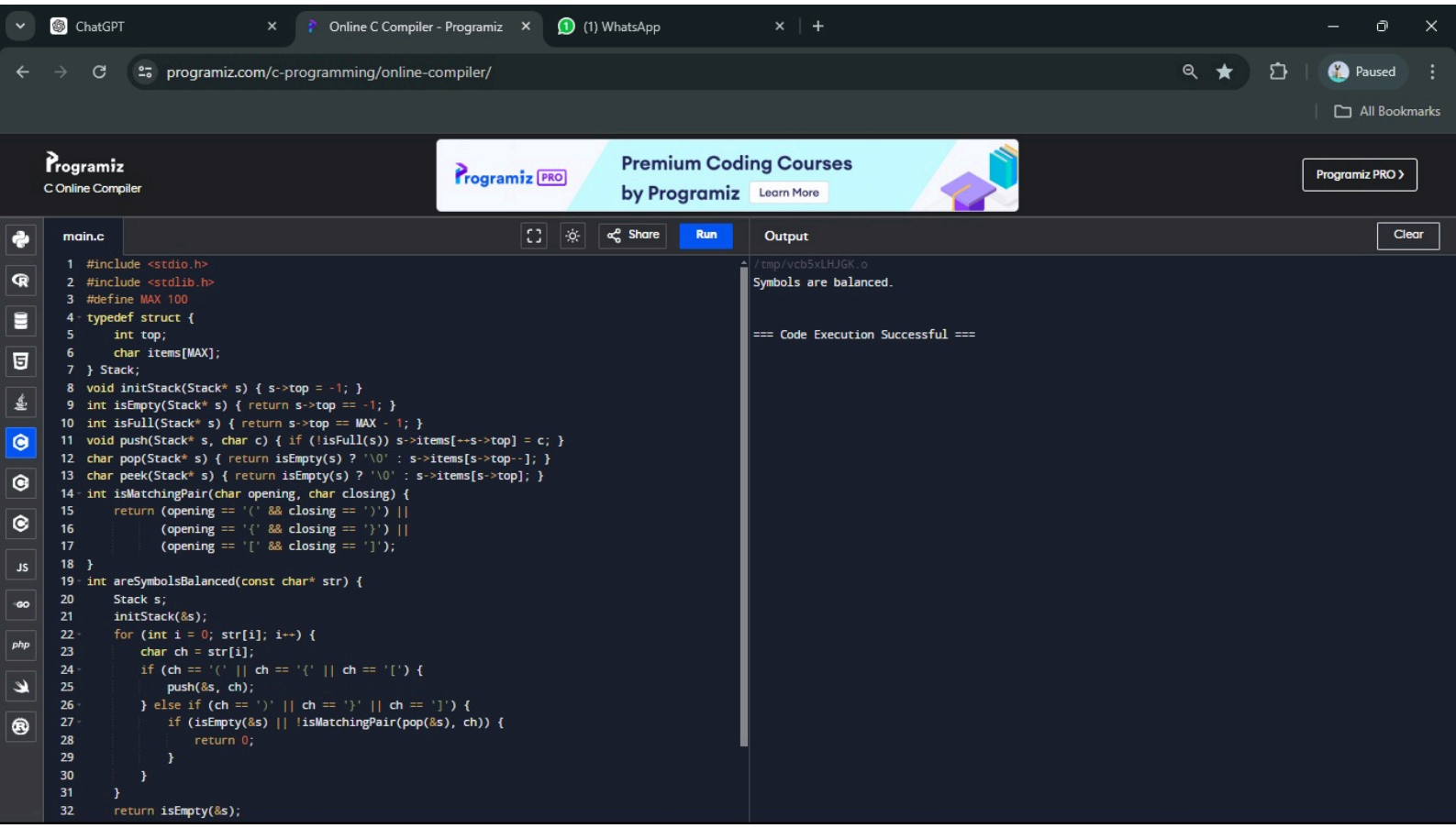
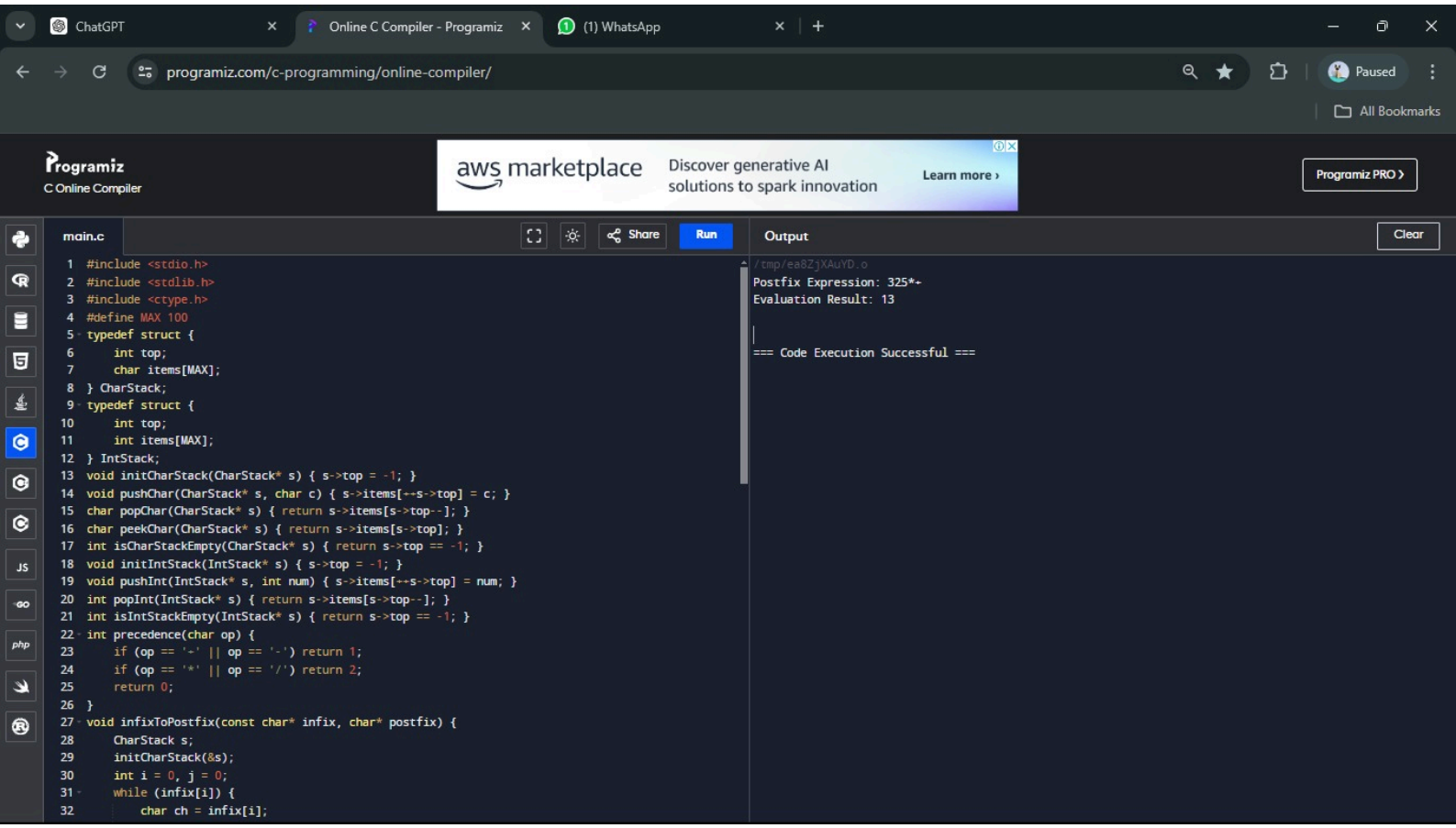
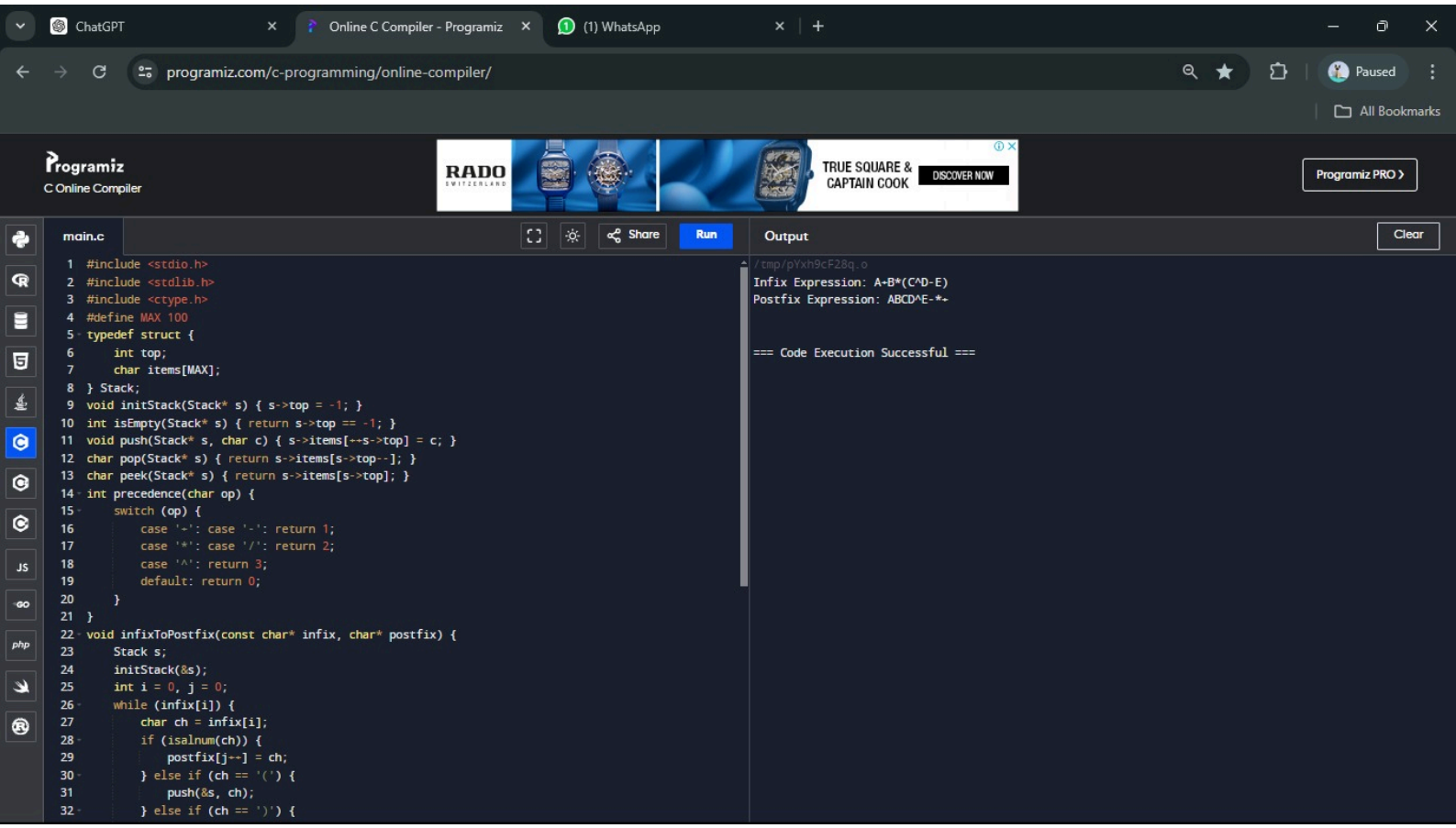

ChatGPT
Online C Compiler - Programiz
(1) WhatsApp
programiz.com/c-programming/online-compiler/
Paused
All Bookmarks
Programiz
C Online Compiler
Programiz PRO
Premium Coding Courses
by Programiz
Learn More
Programiz PRO
main.c
Share
Run
Output
Clear
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
typedef struct {
int top;
char items[MAX];
} Stack;
void initStack(Stack* s) { s->top = -1; }
int isEmpty(Stack* s) { return s->top == -1; }
int isFull(Stack* s) { return s->top == MAX - 1; }
void push(Stack* s, char c) { if (!isFull(s)) s->items[++s->top] = c; }
char pop(Stack* s) { return isEmpty(s) ? '\0' : s->items[s->top--]; }
char peek(Stack* s) { return isEmpty(s) ? '\0' : s->items[s->top]; }
int isMatchingPair(char opening, char closing) {
return (opening == '(' && closing == ')') ||
(opening == '{' && closing == '}') ||
(opening == '[' && closing == ']');
}
int areSymbolsBalanced(const char* str) {
Stack s;
initStack(&s);
for (int i = 0; str[i]; i++) {
char ch = str[i];
if (ch == '(' || ch == '{' || ch == '[') {
push(&s, ch);
} else if (ch == ')' || ch == '}' || ch == ']') {
if (isEmpty(&s) || !isMatchingPair(pop(&s), ch)) {
return 0;
}
}
}
return isEmpty(&s);
/tmp/vcb5xLHJGK.o
Symbols are balanced.
=== Code Execution Successful ===


ChatGPT
Online C Compiler - Programiz
(1) WhatsApp
programiz.com/c-programming/online-compiler/
Paused
All Bookmarks
Programiz
C Online Compiler
aws marketplace
Discover generative AI solutions to spark innovation
Learn more ›
Programiz PRO ›
main.c
Share
Run
Output
Clear
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#define MAX 100
typedef struct {
int top;
char items[MAX];
} CharStack;
typedef struct {
int top;
int items[MAX];
} IntStack;
void initCharStack(CharStack* s) { s->top = -1; }
void pushChar(CharStack* s, char c) { s->items[++s->top] = c; }
char popChar(CharStack* s) { return s->items[s->top--]; }
char peekChar(CharStack* s) { return s->items[s->top]; }
int isCharStackEmpty(CharStack* s) { return s->top == -1; }
void initIntStack(IntStack* s) { s->top = -1; }
void pushInt(IntStack* s, int num) { s->items[++s->top] = num; }
int popInt(IntStack* s) { return s->items[s->top--]; }
int isIntStackEmpty(IntStack* s) { return s->top == -1; }
int precedence(char op) {
if (op == '+' || op == '-') return 1;
if (op == '*' || op == '/') return 2;
return 0;
}
void infixToPostfix(const char* infix, char* postfix) {
CharStack s;
initCharStack(&s);
int i = 0, j = 0;
while (infix[i]) {
char ch = infix[i];
/tmp/ea8ZjXAuYD.o
Postfix Expression: 325*+
Evaluation Result: 13
=== Code Execution Successful ===




```c
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#define MAX 100
typedef struct {
    int top;
    char items[MAX];
} Stack;
void initStack(Stack* s) { s->top = -1; }
int isEmpty(Stack* s) { return s->top == -1; }
void push(Stack* s, char c) { s->items[++s->top] = c; }
char pop(Stack* s) { return s->items[s->top--]; }
char peek(Stack* s) { return s->items[s->top]; }
int precedence(char op) {
    switch (op) {
        case '+': case '-': return 1;
        case '*': case '/': return 2;
        case '^': return 3;
        default: return 0;
    }
}
void infixToPostfix(const char* infix, char* postfix) {
    Stack s;
    initStack(&s);
    int i = 0, j = 0;
    while (infix[i]) {
        char ch = infix[i];
        if (isalnum(ch)) {
            postfix[j++] = ch;
        } else if (ch == '(') {
            push(&s, ch);
        } else if (ch == ')') {
```