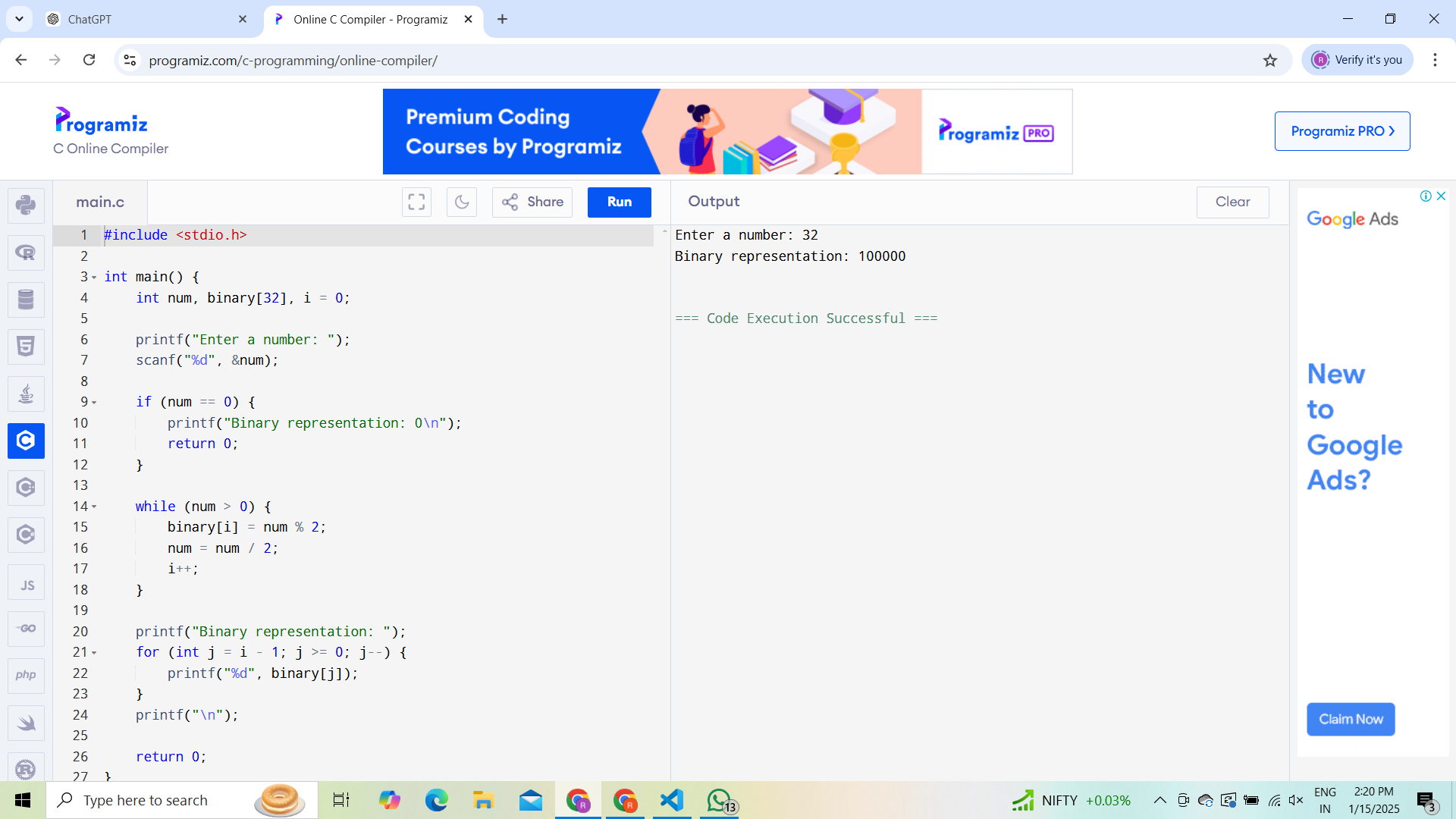
1. **WAP to print the binary representation of a number using loops:**
2. #include <stdio.h>
3. int main() {
4. int num, binary[32], i = 0;
5. printf("Enter a number: ");
6. scanf("%d", &num);
7. if (num == 0) {
8. printf("Binary representation: 0\n");
9. return 0;
10. }
11. while (num > 0) {
12. binary[i] = num % 2;
13. num = num / 2;
14. i++;
15. }
16. printf("Binary representation: ");
17. for (int j = i - 1; j >= 0; j--) {
18. printf("%d", binary[j]);
19. }
20. printf("\n");
21. return 0;
22. }

**OUTPUT:**



**2)WAP to count the no. of vowels and consonants in a string using pointers:**

1. #include <stdio.h>
2. #include <ctype.h>
3. int main() {
4. char str[100], \*ptr;
5. int vowels = 0, consonants = 0;
6. printf("Enter a string: ");
7. fgets(str, 100, stdin);
8. ptr = str;
9. while (\*ptr != '\0') {
10. char ch = tolower(\*ptr);
11. if (ch >= 'a' && ch <= 'z') {
12. if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

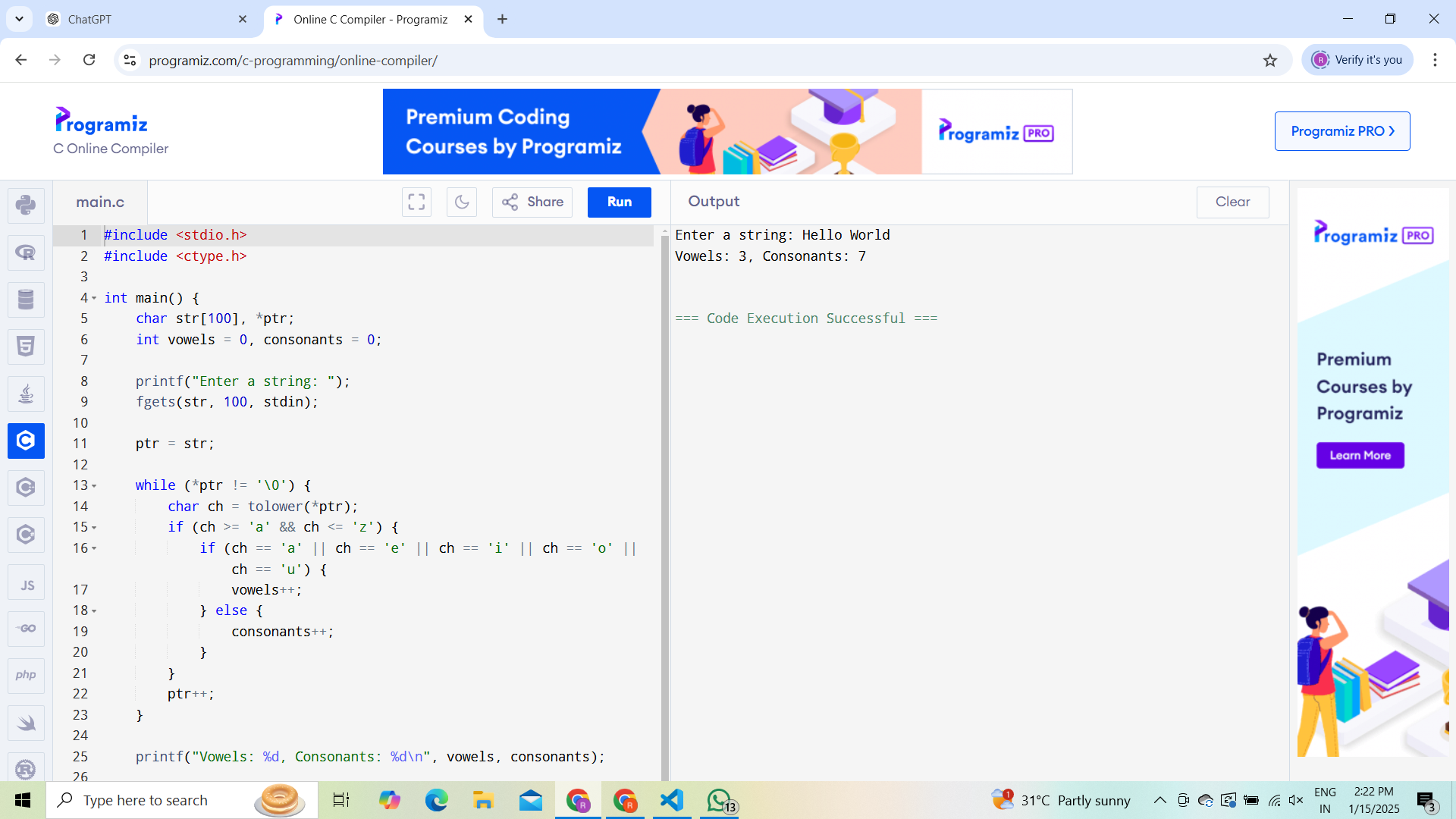
vowels++;

1. } else {

consonants++;

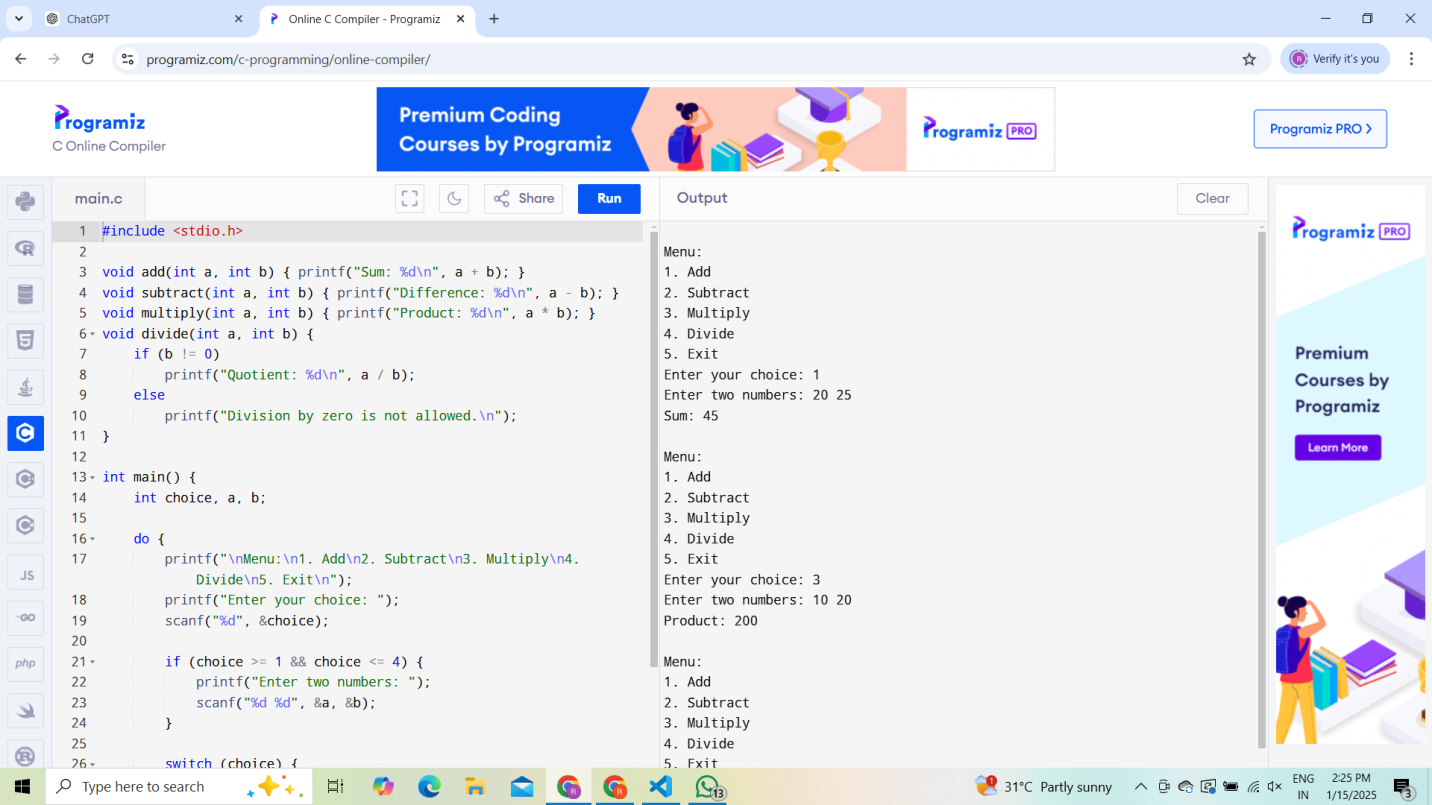
1. }
2. }
3. ptr++;
4. }
5. printf("Vowels: %d, Consonants: %d\n", vowels, consonants);
6. return 0;
7. }

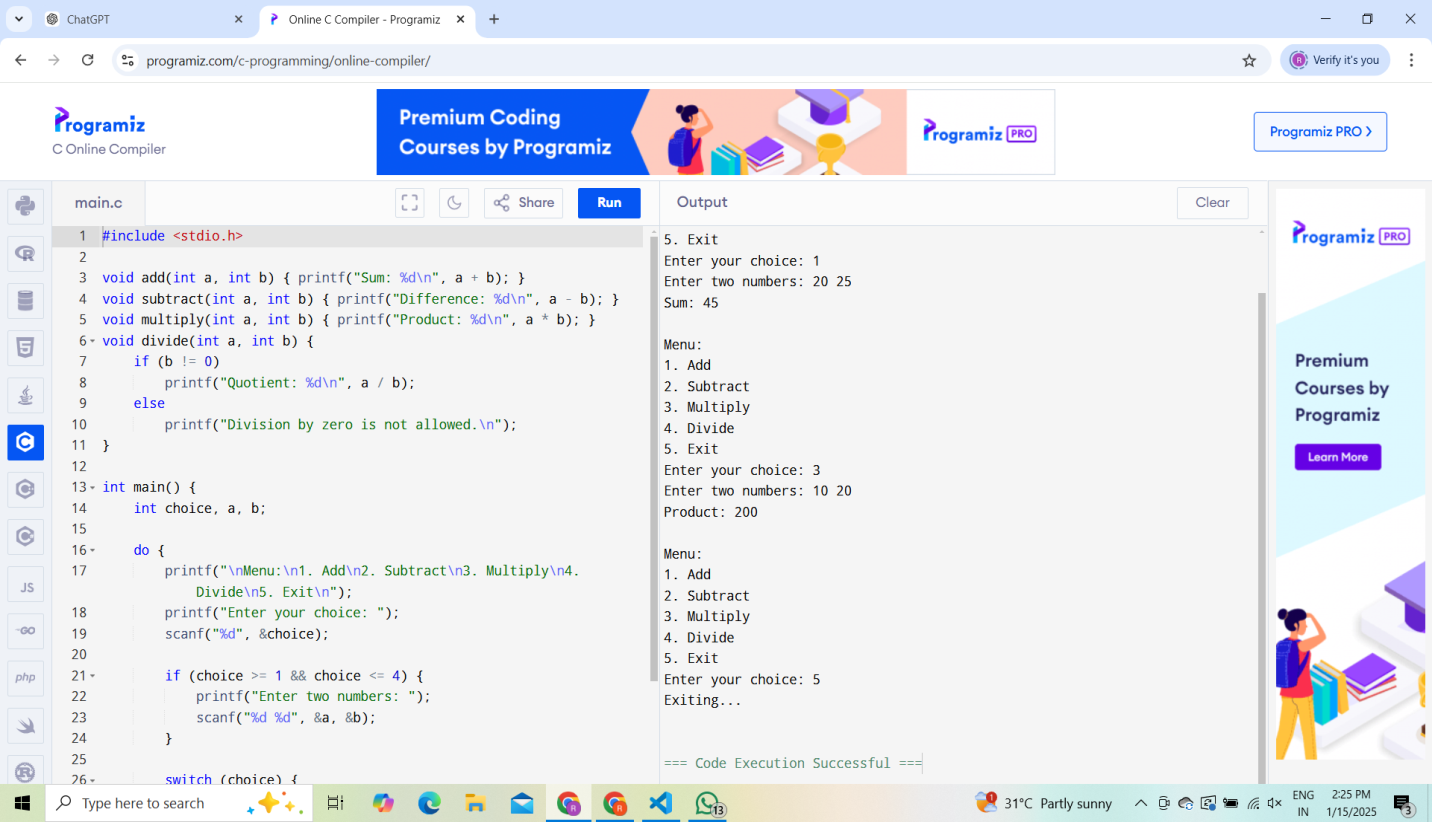
**OUTPUT:**



1. **Write separate functions for addition, subtraction, multiplication and division. Call these functions from a menu driven program.**
2. #include <stdio.h>
3. void add(int a, int b) { printf("Sum: %d\n", a + b); }
4. void subtract(int a, int b) { printf("Difference: %d\n", a - b); }
5. void multiply(int a, int b) { printf("Product: %d\n", a \* b); }
6. void divide(int a, int b) {
7. if (b != 0)
8. printf("Quotient: %d\n", a / b);
9. else
10. printf("Division by zero is not allowed.\n");
11. }
12. int main() {
13. int choice, a, b;
14. do {
15. printf("\nMenu:\n1. Add\n2. Subtract\n3. Multiply\n4. Divide\n5. Exit\n");
16. printf("Enter your choice: ");
17. scanf("%d", &choice);
18. if (choice >= 1 && choice <= 4) {
19. printf("Enter two numbers: ");
20. scanf("%d %d", &a, &b);
21. }
22. switch (choice) {
23. case 1: add(a, b); break;
24. case 2: subtract(a, b); break;
25. case 3: multiply(a, b); break;
26. case 4: divide(a, b); break;
27. case 5: printf("Exiting...\n"); break;
28. default: printf("Invalid choice!\n");
29. }
30. } while (choice != 5);
31. return 0;
32. }

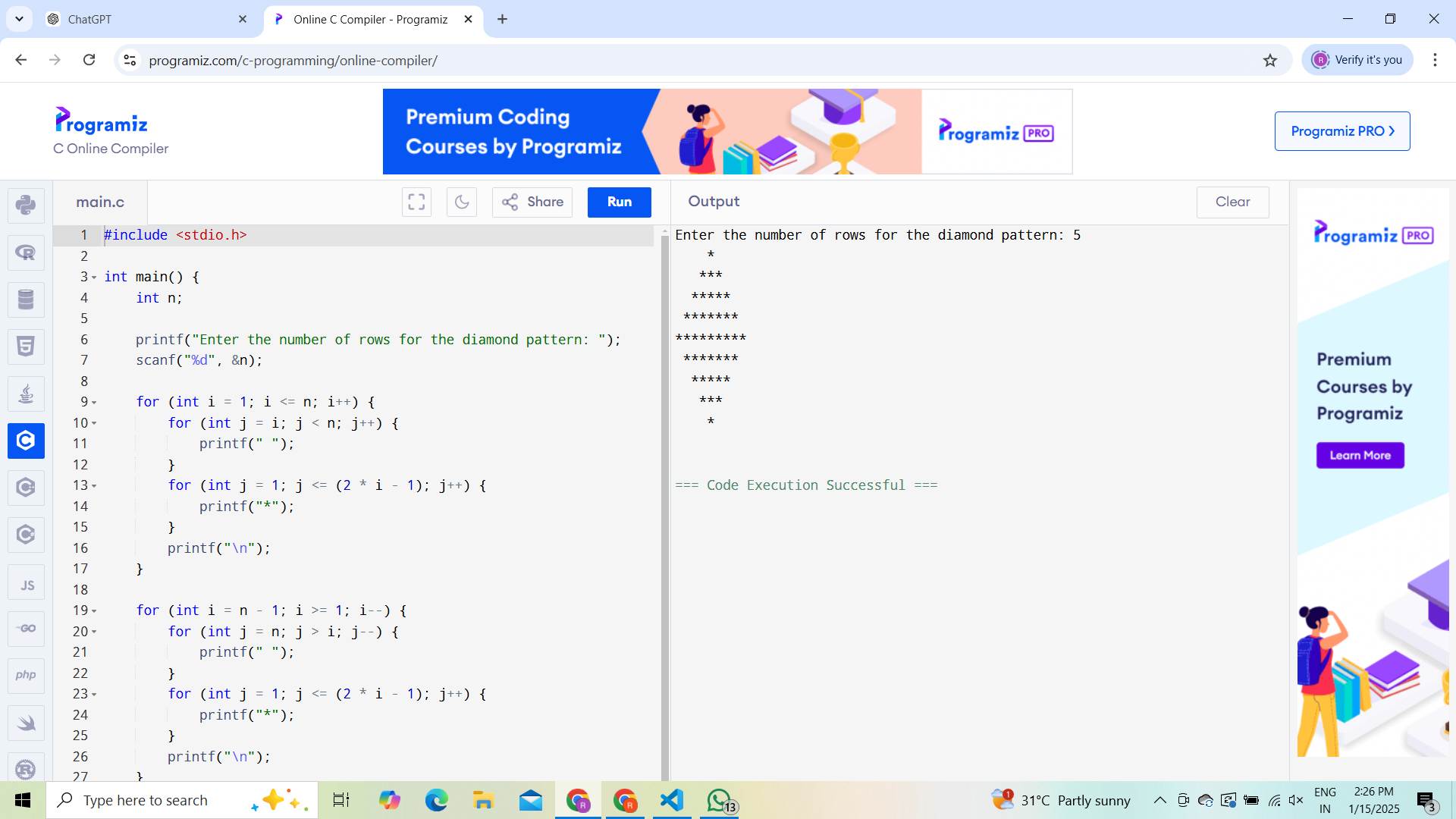
**OUTPUT**

****

****

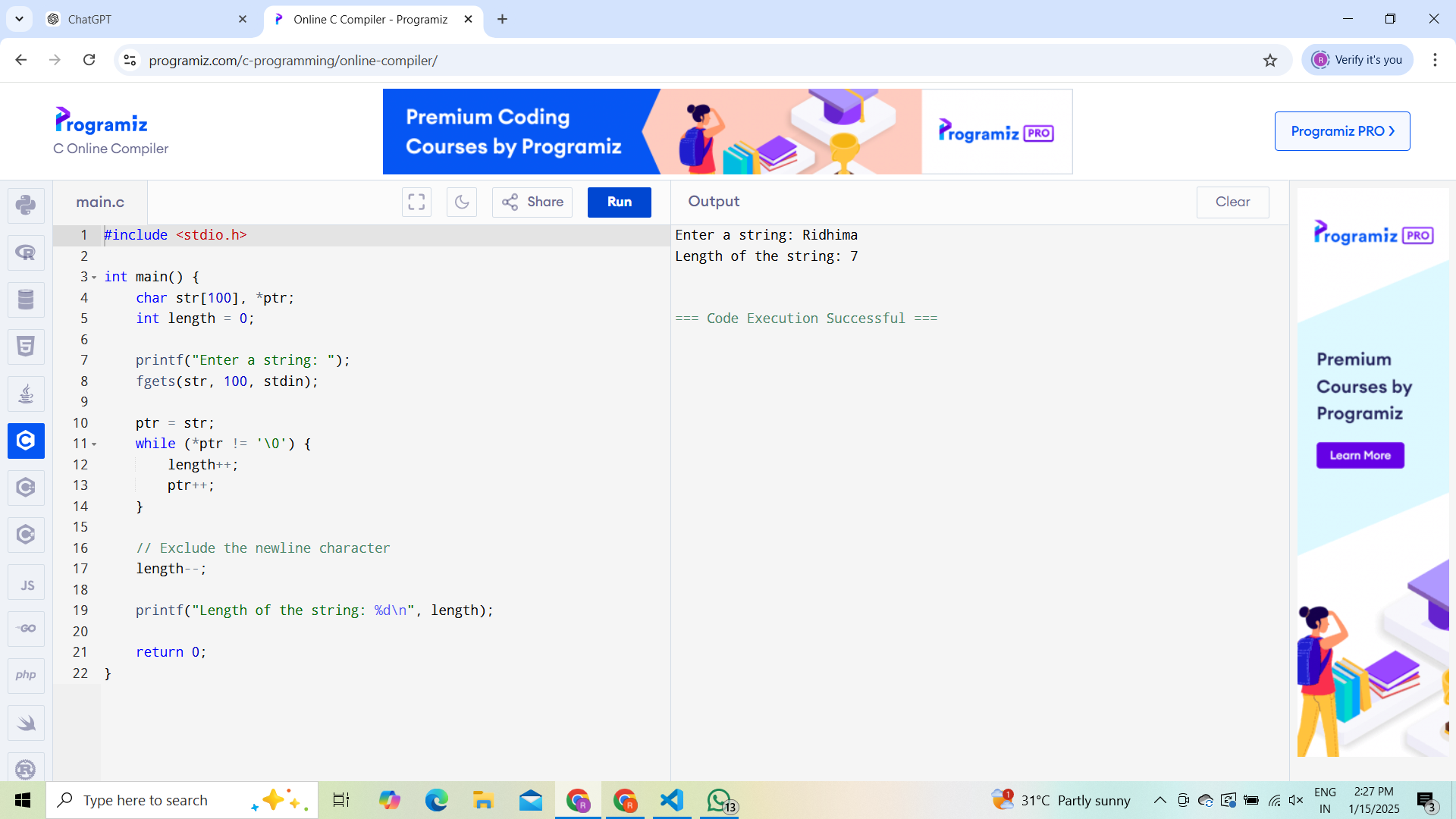
1. **WAP to print a diamond shaped pattern using loops:**
2. #include <stdio.h>
3. int main() {
4. int n;
5. printf("Enter the number of rows for the diamond pattern: ");
6. scanf("%d", &n);
7. for (int i = 1; i <= n; i++) {
8. for (int j = i; j < n; j++) {
9. printf(" ");
10. }
11. for (int j = 1; j <= (2 \* i - 1); j++) {
12. printf("\*");
13. }
14. printf("\n");
15. }
16. for (int i = n - 1; i >= 1; i--) {
17. for (int j = n; j > i; j--) {
18. printf(" ");
19. }
20. for (int j = 1; j <= (2 \* i - 1); j++) {
21. printf("\*");
22. }
23. printf("\n");
24. }
25. return 0;
26. }

**OUTPUT:**

****

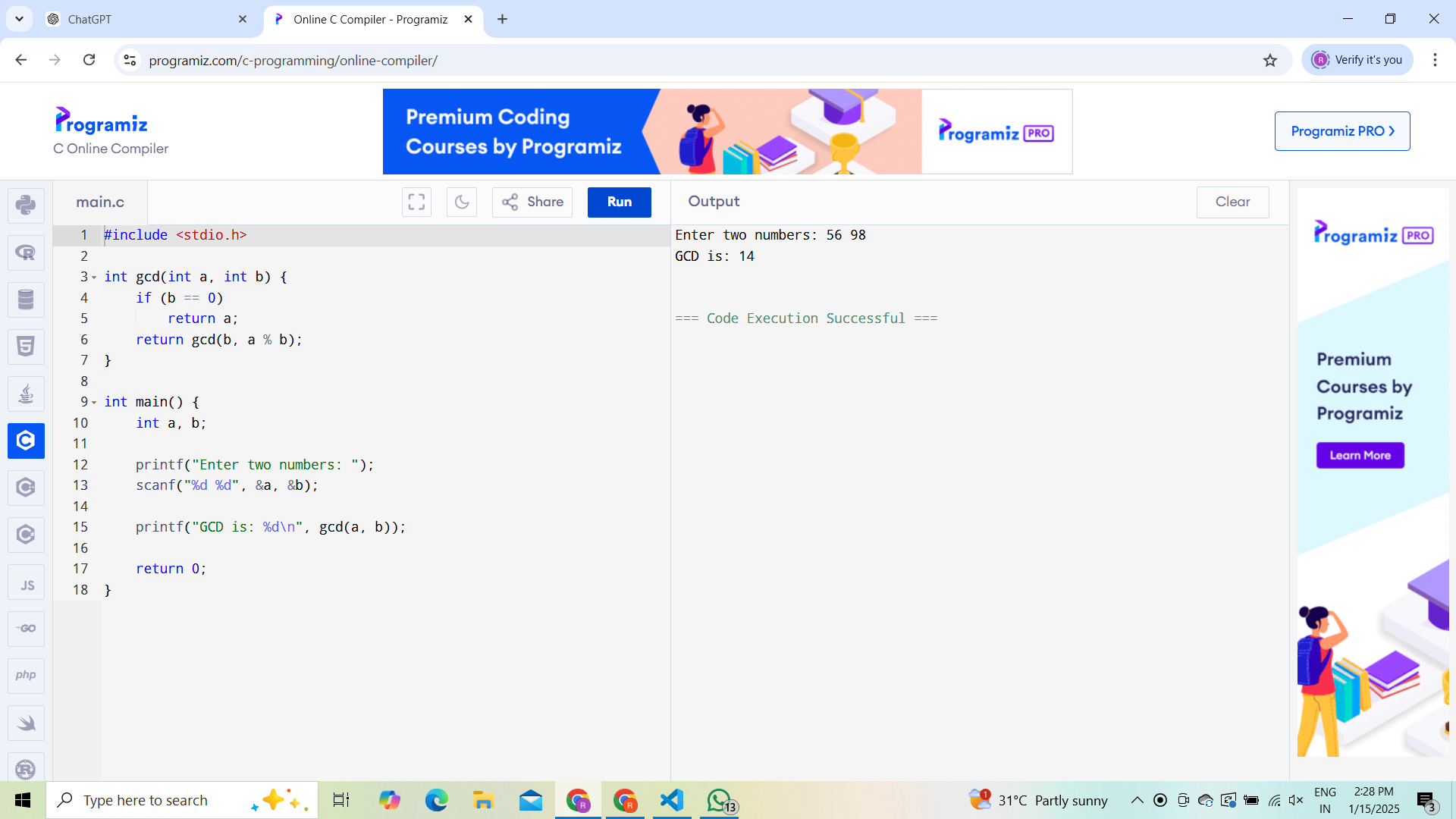
1. **WAP to find the length of the string using pointers:**
2. #include <stdio.h>
3. int main() {
4. char str[100], \*ptr;
5. int length = 0;
6. printf("Enter a string: ");
7. fgets(str, 100, stdin);
8. ptr = str;
9. while (\*ptr != '\0') {
10. length++;
11. ptr++;
12. }
13. // Exclude the newline character
14. Length--;
15. printf("Length of the string: %d\n", length);
16. return 0;
17. }

**OUTPUT:**

****

1. **Write a function to calculate the GCD of two numbers using recursion:**
2. #include <stdio.h>
3. int gcd(int a, int b) {
4. if (b == 0)
5. return a;
6. return gcd(b, a % b);
7. }
8. int main() {
9. int a, b;
10. printf("Enter two numbers: ");
11. scanf("%d %d", &a, &b);
12. printf("GCD is: %d\n", gcd(a, b));
13. return 0;
14. }

**OUTPUT:**

****