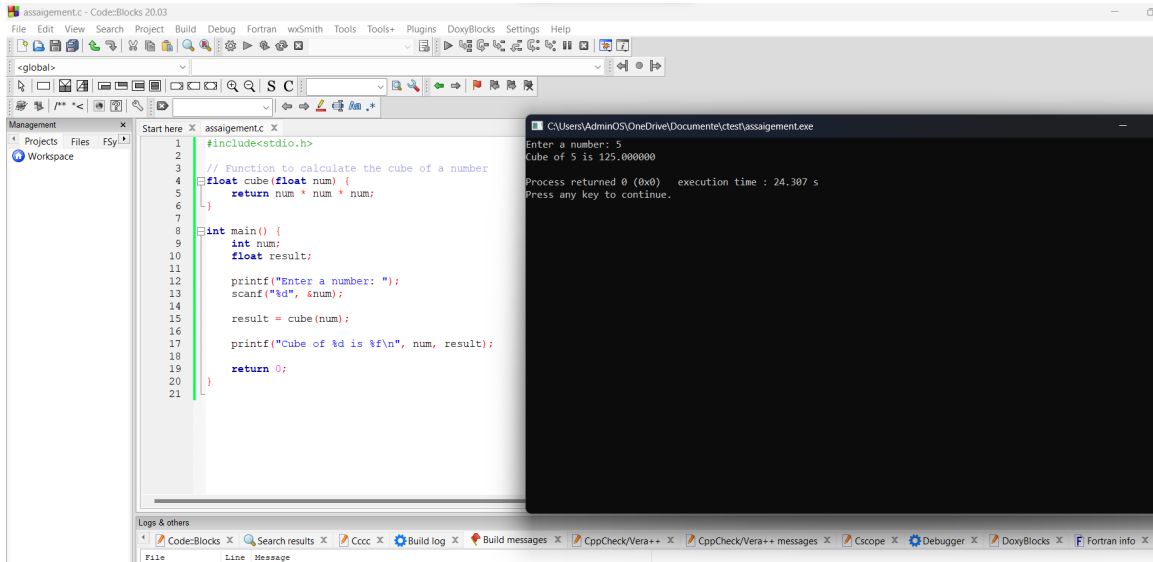


Week 5 tasks :

Prob 1 :



```
#include <stdio.h>

// Function to calculate the cube of a number
float cube(float num) {
    return num * num * num;
}

int main() {
    int num;
    float result;

    printf("Enter a number: ");
    scanf("%d", &num);

    result = cube(num);

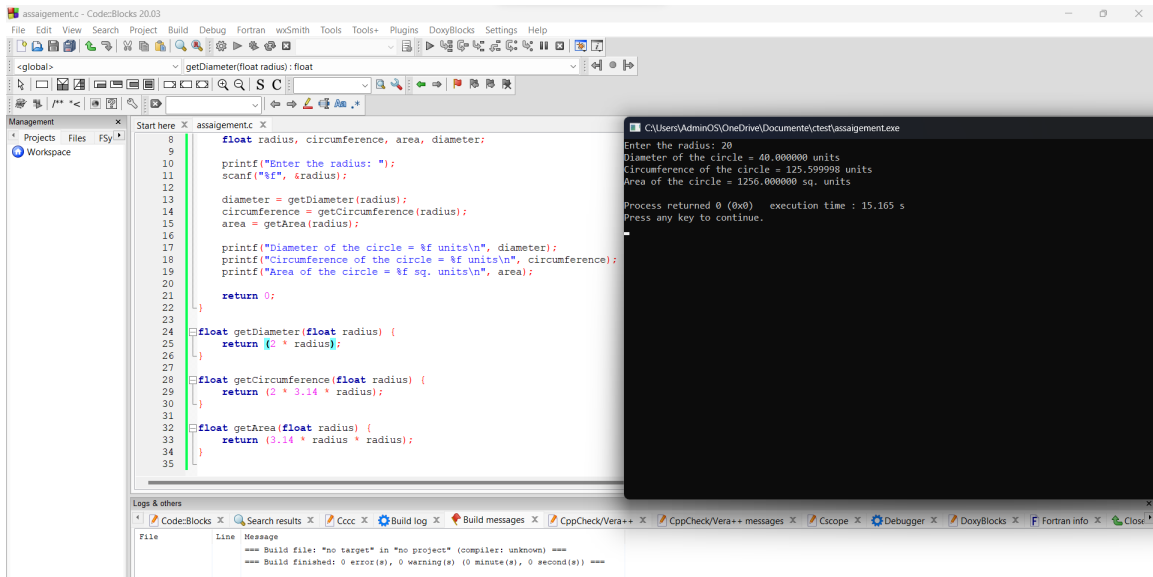
    printf("Cube of %d is %f\n", num, result);

    return 0;
}
```

Terminal Output:

```
Enter a number: 5
Cube of 5 is 125.000000
Process returned 0 (0x0)   execution time : 24.307 s
Press any key to continue.
```

Prob 2 :



```
float radius, circumference, area, diameter;

printf("Enter the radius: ");
scanf("%f", &radius);

diameter = getDiameter(radius);
circumference = getCircumference(radius);
area = getArea(radius);

printf("Diameter of the circle = %f units\n", diameter);
printf("Circumference of the circle = %f units\n", circumference);
printf("Area of the circle = %f sq. units\n", area);

return 0;

float getDiameter(float radius) {
    return (2 * radius);
}

float getCircumference(float radius) {
    return (2 * 3.14 * radius);
}

float getArea(float radius) {
    return (3.14 * radius * radius);
}
```

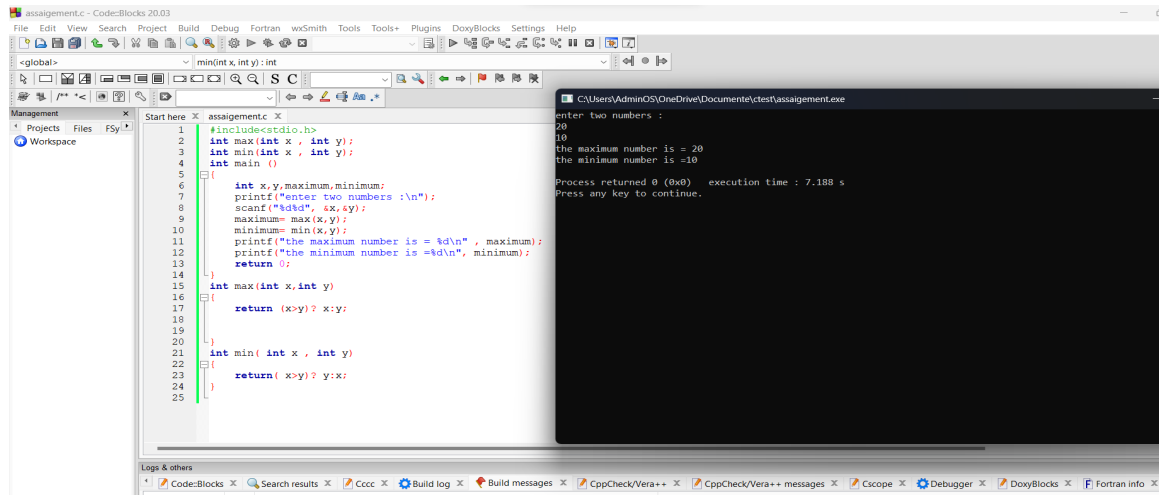
Terminal Output:

```
Enter the radius: 20
Diameter of the circle = 40.000000 units
Circumference of the circle = 125.599998 units
Area of the circle = 1256.000000 sq. units
Process returned 0 (0x0)   execution time : 15.165 s
Press any key to continue.
```

Build Messages:

```
Build file: "no target" in "no project" (compiler: unknown) ==
Build finished: 0 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ==
```

Prob 3 :

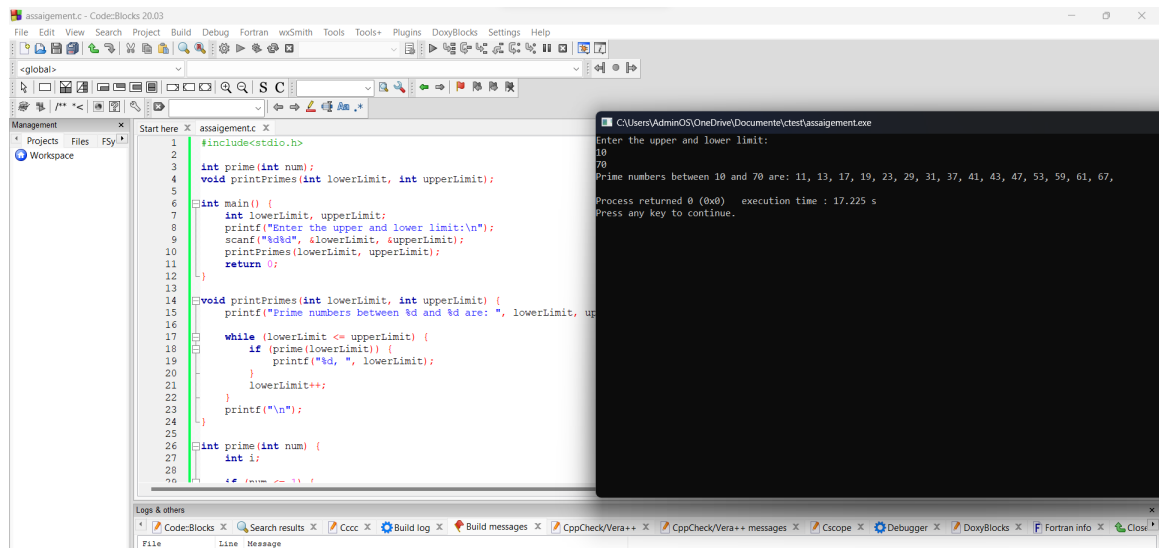


```
1 #include<stdio.h>
2 int max(int x , int y);
3 int min(int x , int y);
4 int main ()
5 {
6     int x,y,maximum,minimum;
7     printf("enter two numbers :\n");
8     scanf("%d%d", &x,&y);
9     maximum= max(x,y);
10    minimum= min(x,y);
11    printf("the maximum number is = %d\n" , maximum);
12    printf("the minimum number is =%d\n", minimum);
13    return 0;
14 }
15 int max(int x,int y)
16 {
17     return (x>y)? x:y;
18 }
19
20 int min( int x , int y)
21 {
22     return (x>y)? y:x;
23 }
24
25
```

Output window content:

```
enter two numbers :
20
10
the maximum number is = 20
the minimum number is =10
Process returned 0 (0x0)   execution time : 7.188 s
Press any key to continue.
```

Prob 6 :

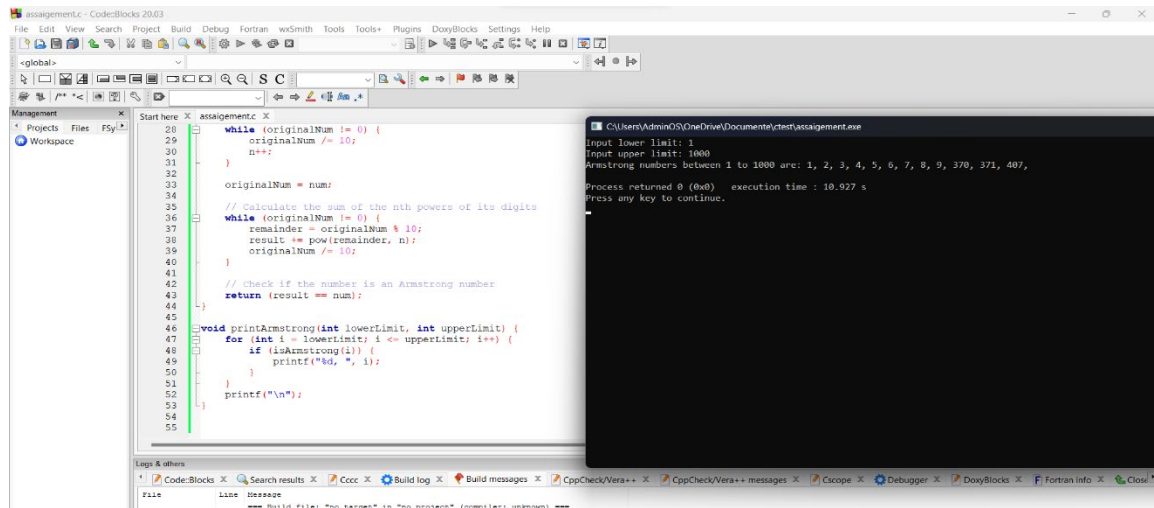


```
1 #include<stdio.h>
2 int prime(int num);
3 void printPrimes(int lowerLimit, int upperLimit);
4
5
6 int main() {
7     int lowerLimit, upperLimit;
8     printf("Enter the upper and lower limit:\n");
9     scanf("%d%d", &lowerLimit, &upperLimit);
10    printPrimes(lowerLimit, upperLimit);
11    return 0;
12 }
13
14 void printPrimes(int lowerLimit, int upperLimit) {
15     printf("Prime numbers between %d and %d are: ", lowerLimit, upperLimit);
16
17     while (lowerLimit <= upperLimit) {
18         if (prime(lowerLimit)) {
19             printf("%d, ", lowerLimit);
20         }
21         lowerLimit++;
22     }
23     printf("\n");
24 }
25
26 int prime(int num) {
27     int i;
28     for (i = 2; i <= num; i++) {
29         if (num % i == 0) {
30             return 0;
31         }
32     }
33     return 1;
34 }
35
```

Output window content:

```
Enter the upper and lower limit:
10
70
Prime numbers between 10 and 70 are: 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67,
Process returned 0 (0x0)   execution time : 17.225 s
Press any key to continue.
```

Prob 8 :

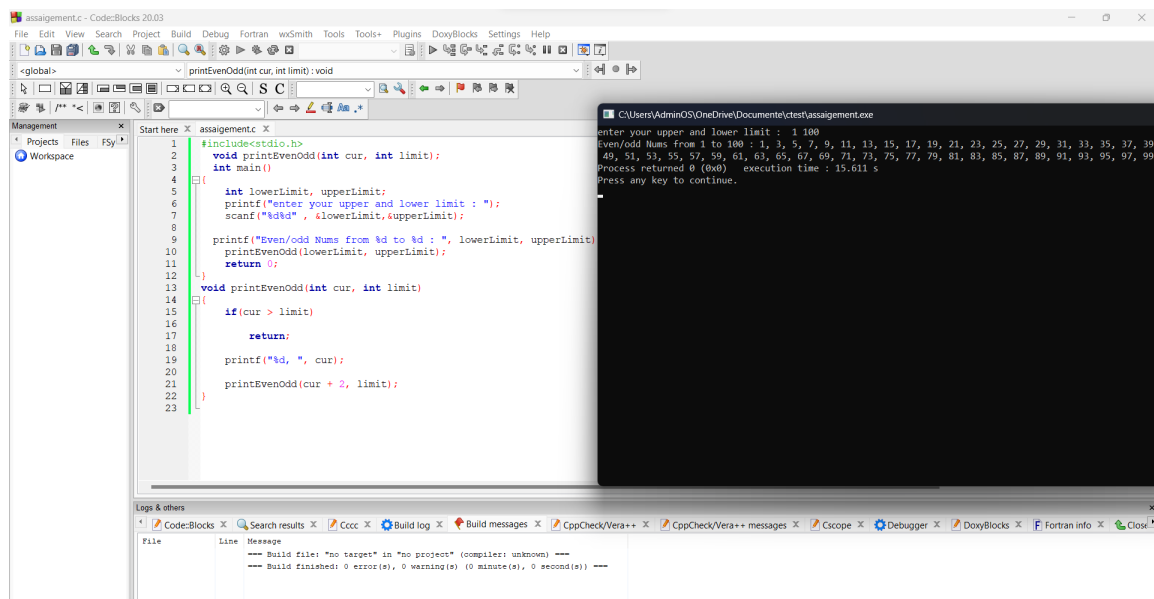


```
26 while (originalNum != 0) {
27     originalNum /= 10;
28     n++;
29 }
30
31 originalNum = num;
32
33 // Calculate the sum of the nth powers of its digits
34 while (originalNum != 0) {
35     remainder = originalNum % 10;
36     result += pow(remainder, n);
37     originalNum /= 10;
38 }
39
40 // Check if the number is an Armstrong number
41 return (result == num);
42 }
43
44 void printArmstrong(int lowerLimit, int upperLimit) {
45     for (int i = lowerLimit; i <= upperLimit; i++) {
46         if (isArmstrong(i)) {
47             printf("%d ", i);
48         }
49     }
50     printf("\n");
51 }
52
53
54
55
```

Terminal Output:

```
C:\Users\AdminOS\OneDrive\Documents\test\assignment.exe
Input lower limit: 1
Input upper limit: 1000
Armstrong numbers between 1 to 1000 are: 1, 2, 3, 4, 5, 6, 7, 8, 9, 370, 371, 407,
Process returned 0 (0x0)   execution time : 10.927 s
Press any key to continue.
```

prob 12 :

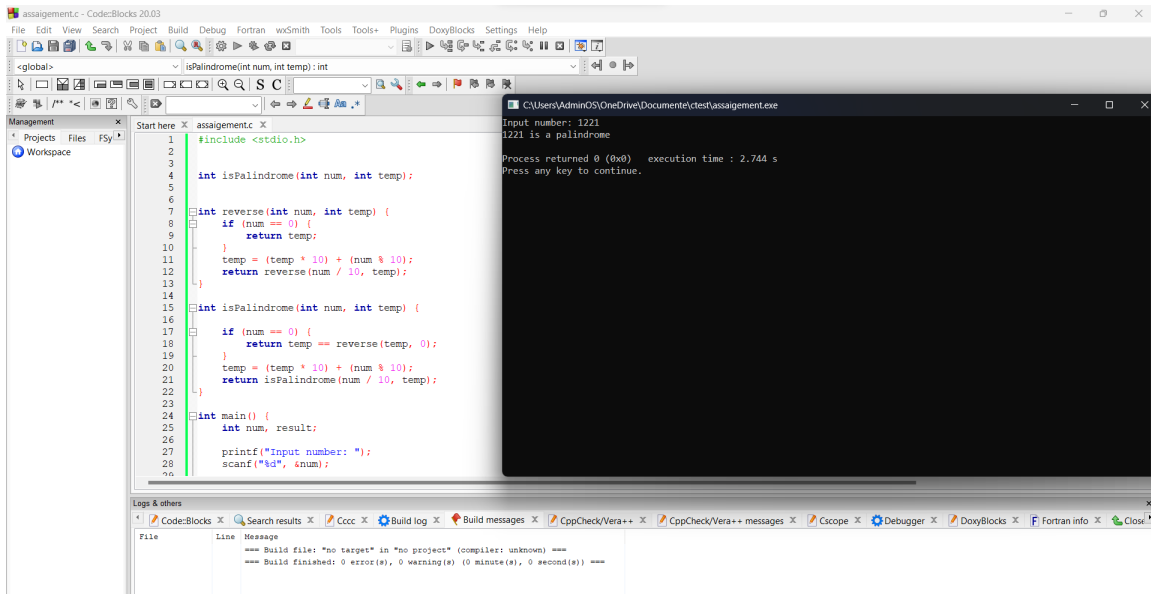


```
1 #include<stdio.h>
2 void printEvenOdd(int cur, int limit);
3 int main()
4 {
5     int lowerLimit, upperLimit;
6     printf("Enter your upper and lower limit : ");
7     scanf("%d%d", &lowerLimit, &upperLimit);
8
9     printf("Even/odd Nums from %d to %d : ", lowerLimit, upperLimit);
10    printEvenOdd(lowerLimit, upperLimit);
11    return 0;
12 }
13 void printEvenOdd(int cur, int limit)
14 {
15     if (cur > limit)
16         return;
17
18     printf("%d ", cur);
19     printEvenOdd(cur + 2, limit);
20 }
21
22
23
```

Terminal Output:

```
C:\Users\AdminOS\OneDrive\Documents\test\assignment.exe
Enter your upper and lower limit : 1 100
Even/odd Nums from 1 to 100 : 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39,
41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99,
Process returned 0 (0x0)   execution time : 15.611 s
Press any key to continue.
```

Prob 16 :



```
#include <stdio.h>

int isPalindrome(int num, int temp);

int reverse(int num, int temp) {
    if (num == 0) {
        return temp;
    }
    temp = (temp * 10) + (num % 10);
    return reverse(num / 10, temp);
}

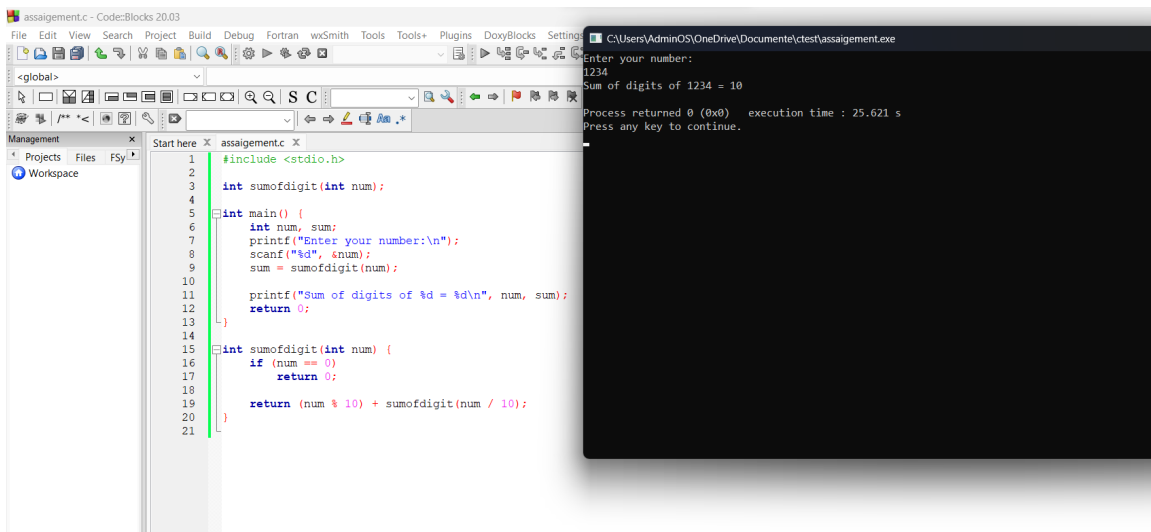
int isPalindrome(int num, int temp) {
    if (num == 0) {
        return temp == reverse(temp, 0);
    }
    temp = (temp * 10) + (num % 10);
    return isPalindrome(num / 10, temp);
}

int main() {
    int num, result;
    printf("Input number: ");
    scanf("%d", &num);
}
```

Output window output:

```
Input number: 1221
1221 is a palindrome
Process returned 0 (0x0)   execution time : 2.744 s
Press any key to continue.
```

Prob 17 :



```
#include <stdio.h>

int sumofdigit(int num);

int main() {
    int num, sum;
    printf("Enter your number:\n");
    scanf("%d", &num);
    sum = sumofdigit(num);
    printf("Sum of digits of %d = %d\n", num, sum);
    return 0;
}

int sumofdigit(int num) {
    if (num == 0)
        return 0;
    return (num % 10) + sumofdigit(num / 10);
}
```

Output window output:

```
Enter your number:
1234
Sum of digits of 1234 = 10
Process returned 0 (0x0)   execution time : 25.621 s
Press any key to continue.
```

Prob 18 :

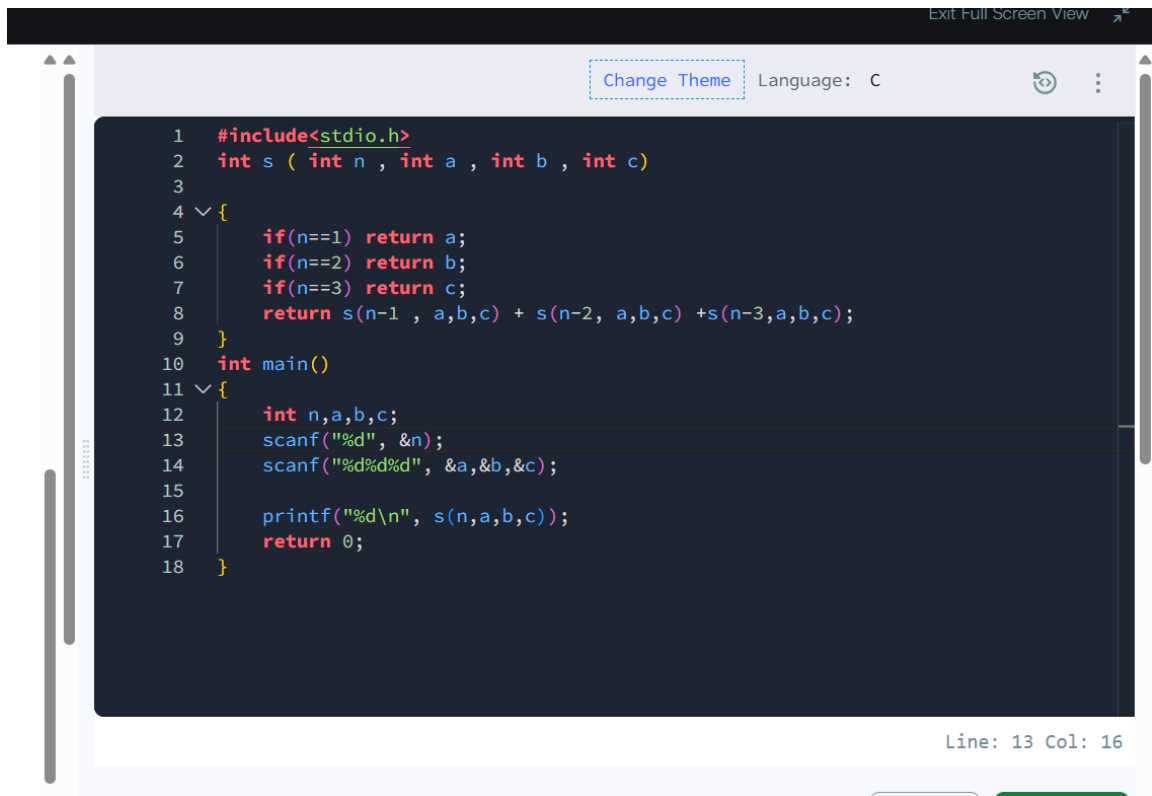
The screenshot shows the Code::Blocks IDE with a C program for calculating the factorial of a number. The program is named 'assaigement.c' and is located in the 'C:\Users\AdminOS\OneDrive\Documents\ctest\assaigement.exe' directory. The code is as follows:

```
1 #include <stdio.h>
2
3 unsigned long long fact(int num);
4
5
6 unsigned long long fact(int num) {
7     if (num == 0) {
8         return 1;
9     } else {
10         return num * fact(num - 1);
11     }
12 }
13
14 int main() {
15     int num;
16     printf("Input any number: ");
17     scanf("%d", &num);
18     printf("Factorial of %d = %llu\n", num, fact(num));
19     return 0;
20 }
21
22
```

The execution output is shown in the terminal window:

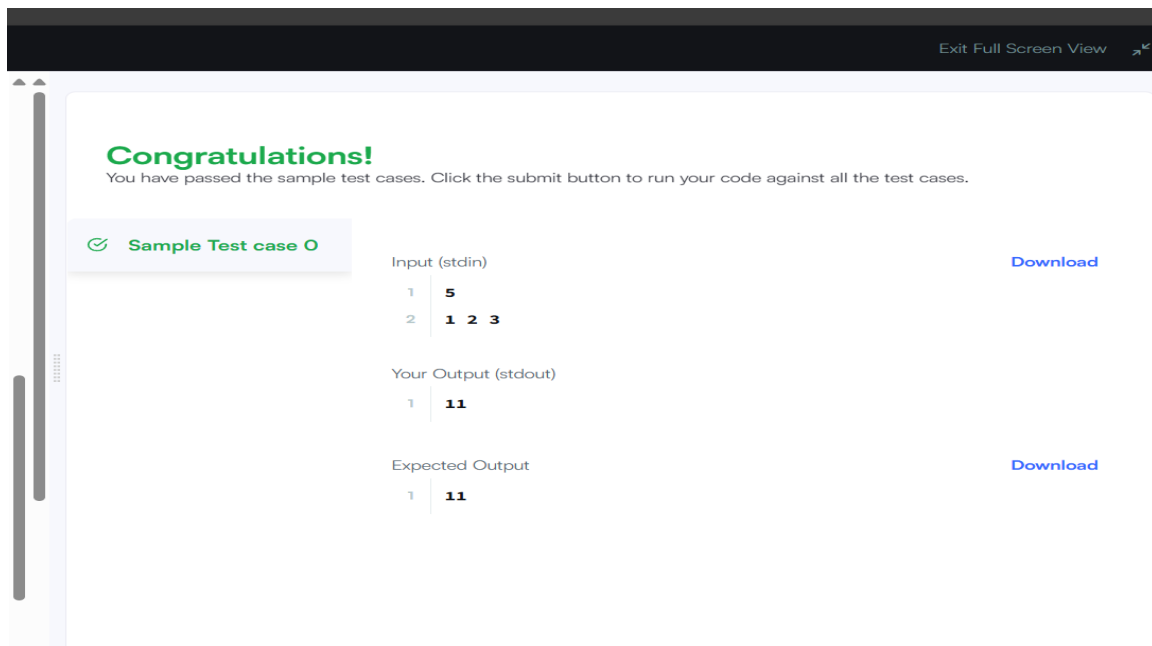
```
Input any number: 5
Factorial of 5 = 120
Process returned 0 (0x0)   execution time : 14.172 s
Press any key to continue.
```

Hacker rank:



```
1  #include<stdio.h>
2  int s ( int n , int a , int b , int c)
3
4  {
5      if(n==1) return a;
6      if(n==2) return b;
7      if(n==3) return c;
8      return s(n-1 , a,b,c) + s(n-2, a,b,c) +s(n-3,a,b,c);
9  }
10 int main()
11 {
12     int n,a,b,c;
13     scanf("%d", &n);
14     scanf("%d%d%d", &a,&b,&c);
15
16     printf("%d\n", s(n,a,b,c));
17     return 0;
18 }
```

Line: 13 Col: 16



Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ **Sample Test case 0**

Input (stdin) [Download](#)

1	5
2	1 2 3

Your Output (stdout)

1	11
---	----

Expected Output [Download](#)

1	11
---	----

Problem 2 :

The screenshot shows the Code::Blocks IDE with a C program named 'assaigement.c' open. The program calculates the sum of marks for a given number of students and gender. The code is as follows:

```
9  for (int i = start; i < number_of_students; i += 2) {
10      sum += marks[i];
11  }
12
13  return sum;
14
15  }
16
17  int main() {
18      int number_of_students;
19      char gender;
20      int sum;
21
22      scanf("%d", &number_of_students);
23      int *marks = (int *) malloc(number_of_students * sizeof(int));
24
25      for (int student = 0; student < number_of_students; student++)
26          scanf("%d", &marks[student]);
27
28      scanf("%c", &gender);
29      sum = marks_summation(marks, number_of_students, gender);
30      printf("%d\n", sum);
31
32      free(marks);
33
34      return 0;
35  }
```

The program is compiled and executed. The output window shows the following message:

```
Process returned 0 (0x0)   execution time : 49.688 s
Press any key to continue.
```

The 'Logs & others' panel at the bottom shows the following messages:

```
==== Build file: "no target" in "no project" (compiler: unknown) ====
In function 'main':
C:\Users\Ad... 22  warning: implicit declaration of function 'malloc' [-Wimplicit-function-dec...
C:\Users\Ad... 22  warning: incompatible implicit declaration of built-in function 'malloc'
```