1. Predict the output

```
#include <bits/stdc++.h>
using namespace std;
int main() {
while ('1' < '2')
cout << "In while loop" << endl;
}
```

Output:
In while loop
In while loop
In while loop
…..Infinite times

2.Predict the output

```
#include <bits/stdc++.h>
using namespace std;
int main( ) {
int t = 10;
while (t /= 2) {
cout << "Hello" << endl;
}
}
```

Output:
Hello
Hello
Hello

3.Predict the output

```
#include <bits/stdc++.h>
using namespace std;
int main( ) {
for (int x = 1; x * x <= 10; x++)
cout << "In for loop" << endl;
}
```

Output:
In for loop
In for loop
In for loop

4.Predict the output

```
#include <bits/stdc++.h>
using namespace std;
int main( ) {
int x = 10, y = 0 ;
while ( x >= y ) {
x-- ;
y++ ;
cout << x << " " << y << endl ;
}
}
```

Output:

9 1
8 2
7 3
6 4
5 5
4 6

5.WAP to print the sum of all the even digits of a given number.

main.cpp

```
#include <iostream>
using namespace std;

int main() {
    int number, sum_even = 0;
    cout << "Enter a number: ";
    cin >> number;

    while (number > 0) {
        int digit = number % 10;
        if (digit % 2 == 0) {
            sum_even += digit;
        }
        number /= 10;
    }

    cout << "The sum of all even digits is: " << sum_even << endl;
    return 0;
}
```

Run

Output

```
/tmp/wcrFnp6J4Z.o
Enter a number: 4556
The sum of all even digits is: 10


=== Code Execution Successful ===
```

6.WAP to print the sum of a given number and its reverse.

main.cpp Run Output

```cpp
#include <iostream>
using namespace std;

int main() {
    int number, reversedNumber = 0, originalNumber;
    cout << "Enter a number: ";
    cin >> number;

    originalNumber = number;

    while (number > 0) {
        int digit = number % 10;
        reversedNumber = reversedNumber * 10 + digit;
        number /= 10;
    }

    int sum = originalNumber + reversedNumber;

    cout << "The sum of " << originalNumber << " and its reverse " <<
        reversedNumber << " is: " << sum << endl;

    return 0;
}
```

```
/tmp/Y2ZgPTNJxQ.o
Enter a number: 12
The sum of 12 and its reverse 21 is: 33


=== Code Execution Successful ===
```

7.Print the factorials of first ‘n’ numbers

main.cpp Run Output

```cpp
#include <iostream>
using namespace std;

int main() {
    int n;
    cout << "Enter the value of n: ";
    cin >> n;

    for (int i = 1; i <= n; ++i) {
        int fact = 1;
        for (int j = 1; j <= i; ++j) {
            fact *= j;
        }
        cout << fact << endl;
    }

    return 0;
}
```

```
/tmp/45IGqXXOiw.o
Enter the value of n: 10
1
2
6
24
120
720
5040
40320
362880
3628800


=== Code Execution Successful ===
```

8.Print first ‘n’ fibonacci numbers.

main.cpp Run Output

```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
    int n;
    cin >> n;

    int firstFib = 1, secondFib = 1;
    cout << firstFib << " " << secondFib << " ";

    for (int i = 3; i <= n; i++) {
        int nextFib = firstFib + secondFib;
        cout << nextFib << " ";
        firstFib = secondFib;
        secondFib = nextFib;
    }
    cout << endl;

    return 0;
}
```

```
/tmp/LT41W638I3.o
10
1 1 2 3 5 8 13 21 34 55


=== Code Execution Successful ===
```

9.Write a program to print out all Armstrong numbers between 1 and 500. If the sum of cubes of each digit of the number is equal to the number itself, then the number is called an Armstrong number. For example, 153 = (1 * 1 * 1) + (5 * 5 * 5) + (3 * 3 * 3)

main.cpp

Run

Output

```cpp
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    cout << "Armstrong numbers between 1 and 500 are: " << endl;

    for (int number = 1; number <= 500; number++) {
        int sum = 0;
        int temp = number;
        while (temp > 0) {
            int digit = temp % 10;
            sum += pow(digit, 3);
            temp /= 10;
        }
        if (sum == number) {
            cout << number << " ";
        }
    }

    cout << endl;
    return 0;
}

```

```
/tmp/yN1kY85qMj.o
Armstrong numbers between 1 and 500 are:
1 153 370 371 407


=== Code Execution Successful ===
```