1)Print Numbers from 1 to N

Write a C program that takes an integer N as input and prints all the numbers from 1 to N

Sample input1:

```
Enter a value for N: 5
```

Sample output1:

```
1 2 3 4 5
```

2)Print Even Numbers

Write a C program that takes an integer N as input and prints all the even numbers from 1 to N

Sample input1:

```
Enter a value for N: 10
```

Sample output1:

```
2 4 6 8 10
```

3)Print Multiplication Table

Write a C program that takes an integer N as input and prints the multiplication table for N up to 10

Sample input1:

```
Enter a value for N: 3
```

Sample output1:

```
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
```

3 x 9 = 27
3 x 10 = 30

4)Factorial of a Number

Write a C program that takes an integer N as input and calculates the factorial of N

Sample input1:

Enter a value for N: 4

Sample output1:

Factorial of 4 is 24

5) Craft a program that prompts a user to input an integer, find the smallest 5 prime number greater than a given input, and output the result.

Sample Input:

Enter an integer: 10

Sample Output:

The smallest 5 prime numbers greater than 10 are: 11 13 17 19 23

6) Craft a program that prompts the user to input a number. The program should then determine whether the entered number is a prime or composite number.

Sample Input 1:

Enter an integer: 17

Sample Output 1:

17 is a prime number.

Sample Input 2:

```
Enter an integer: 18
```

Sample Output 2:

```
18 is a composite number.
```

7) Craft a program that prompts the user to input the next term of a series and the number of terms in the series. The program then calculates the sum of the series and displays it along with the series itself.

Sample Input 1:

```
Enter the first term of the series: 2
Enter the common difference of the series: 3
Enter the number of terms in the series: 5
```

Sample Output 1:

```
The series is: 2 5 8 11 14
The sum of the series is: 40
```

8) Craft a program to prompt the user to input a number. A loop is used to find and display the positive divisors of the entered number. If a number i is a divisor, its value is added to the sum.After finding all the divisors, it prints the sum of the divisors. Finally, it checks if the sum is equal to the original number and prints a output
eg: X is an equal number
Y is not an equal number

Sample Input 1:

```
Enter an integer: 6
```

Sample Output 1:

```
The positive divisors of 6 are: 1 2 3 6
The sum of the divisors is: 12
6 is not an equal number.
```

9) Craft a program that prompts the user to enter the input. An abundant number is a positive integer for which the sum of its proper divisors (excluding itself) is greater than the number itself. Your program should be able to take an integer input from the user and check whether a given integer number is an abundant number or not.

Sample Input 1:

```
Enter an integer: 12
```

Sample Output 1:

```
The proper divisors of 12 are: 1 2 3 4 6
The sum of the proper divisors is: 16
12 is an abundant number.
```

Explanation:
Since 16 is greater than 12, 12 is an abundant number.

10) Craft a program that prompts the user to enter a year as input. Check whether the given input is leap year or not. Then count the number of leap years and non-leap years in the next decade from the given input. Finally print the output.

Sample Input 1:

```
Enter a year: 2020
```

Sample Output 1:

```
2020 is a leap year.
In the next decade from 2020:
Number of leap years: 2
Number of non-leap years: 8
```

11) Craft a program that takes an input from the user and N representing the number of terms and that calculates the sum of the first N terms of the given geometric series. Finally, the program should output the sum of the first N terms of the series with 2 digit precision. For negative input print 0.00 as output.

Sample Input 1:

```
Enter the initial term of the series: 2
Enter the common ratio of the series: 3
Enter the number of terms in the series: 4
```

Sample Output 1:

```
The sum of the first 4 terms of the series is: 80.00
```

12) Craft a program that prompts the user to input a positive integer N. Calculate the sum of the squares by squaring each natural number from 1 to N and adding them together.Ensure the program handles invalid inputs if N is less than or equal to 0 then Handle the input as "Invalid input"

Sample Input 1:

```
Enter a positive integer: 5
```

Sample Output 1:

```
The sum of the squares of the first 5 natural numbers is: 55
```

13) Craft a program that takes the positive integer 'n' from the user and then meticulously computes the sum of the harmonic series up to 'n' . If a non-positive number is given, print "Invalid Input". Print the results with two decimal points(precision 2).

Sample Input 1:

```
Enter a positive integer: 5
```

Sample Output 1:

```
The sum of the harmonic series up to 5 is: 2.28
```

14) Craft a program that prompts the user to enter an input. Develop a program that generates random pairs of people and counts the total number of handshakes among them. Ensure that no person shakes hands with themselves.Finally print the output.Total handshakes=n×(n−1)/2

Sample Input 1:

```
Enter the number of people: 4
```

Sample Output 1:

```
Total number of handshakes among 4 people is: 6
```

15) Craft a program that takes two user-input integers, m and n, representing the lower and upper bounds of a range respectively. The program should then identify and print all the perfect numbers within that range. A perfect number is a positive integer that is equal to the sum of its proper divisors, excluding itself.

Sample input:

```
Enter the lower bound (m): 1
Enter the upper bound (n): 500
```

Sample output:

```
Perfect numbers between 1 and 500 are:
6 28 496
```

16) Craft a program that prompts the user to input two integers, m and n, representing the lower and upper bounds of a range respectively. The program should then calculate and print the sum of all cubes within this range, inclusive of the endpoints m and n.

Sample Input 1:

```
Enter the lower bound (m): 1
Enter the upper bound (n): 3
```

Sample Output 1:

```
The sum of cubes from 1 to 3 is: 36
```

17) Craft a program that prompts the user to enter the height of the inverted left half pyramid. Then, it uses nested loops to print the pattern by printing spaces and characters (from 'A' to 'Z') in decreasing order on each row.

Sample Input:

```
Enter the height of the inverted left half pyramid: 4
```

Sample Output:

```
ABCD
 ABCD
  ABCD
   ABCD
```

18) Craft a program that takes an integer input from the user and calculates the factorial of each digit of the input number. Then, sum up the factorials of all digits and print the result.

Sample Input:

Enter an integer: 123

Sample Output:

Sum of factorials of digits: 9

19) Craft a program that takes an input number 'n' from the user, then iterates through numbers from 2 to 'n', checking each number for primality by dividing it by all numbers less than itself. If a number is prime, it adds it to the sum. Finally, it prints the sum of prime numbers.

Sample Input:

Enter a number: 10

Sample Output:

Sum of prime numbers from 2 to 10 is: 17

20) Craft a program that prompts the user to input a single character, then checks whether the input is vowel or consonant .It keeps track of the counts of vowels and consonants encountered. Finally, it decides whether the word is predominantly vowels, predominantly Consonants, or if the input is invalid.

Sample Input:

Enter a single character (or '0' to end): A
Enter a single character (or '0' to end): b
Enter a single character (or '0' to end): C

Enter a single character (or '0' to end): e
Enter a single character (or '0' to end): i
Enter a single character (or '0' to end): 0

Sample Output:

Total vowels: 3
Total consonants: 2
The input is predominantly vowels.
17)Write a C program that takes a positive integer n from the user and prints a multiplication table from 1 to n.

Sample Input:
Enter the number for the multiplication table: 5

Sample Output:
Multiplication table for 5:

```
      1   2   3   4   5
  1   1   2   3   4   5
  2   2   4   6   8   10
  3   3   6   9  12  15
  4   4   8  12  16  20
  5   5  10  15  20 25
```

21)Write a C program that generates a series of numbers where each number is twice the previous number.

Example:
Input:

Starting number: 3

Output:

3 6 12 24 48

22)Write a C program that takes an integer input count from the user, prints the first count odd numbers, and calculates their sum.

Input:

Count:5

Output:

1 3 5 7 9

Sum: 25

23)Write a C program that performs the following tasks:

1. Prompt the user to input a positive integer.
2. Use a loop to find and display all the positive divisors of the entered number.
3. Calculate the sum of these divisors.
4. After displaying all the divisors and their sum, check if the sum is equal to the original number.
5. Print whether the original number is an equal number or not based on the comparison.

An **equal number** is a number where the sum of its positive divisors (excluding itself) equals the number.

Sample input1:

Enter a number: 6

Sample Output1:

6 is an equal number

Explanation:

The divisors of 6 are: 1 2 3

The sum of the divisors is: 6

24)Write a C program that calculates the sum of the squares of the first n natural numbers, where n is provided by the user. The program should:

1. Prompt the user to enter a positive integer n.
2. Compute the sum of the squares of the numbers from 1 to n.
3. Display the result.

Sample Input1:

4

Sample Output1:

The sum of the squares of the first 4 natural numbers is: 30

Sample input2:

6

Sample output2:

The sum of the squares of the first 6 natural numbers is: 91

25)Write a program that gets two integers as input, calculates the sum of the given numbers, and checks if the sum is a palindrome or not.

Input Format

- A (First integer)
- B (Second integer)

Sample test case:

1. Input:
   123
   321
   Output:
   Sum of numbers: 444
   444 is a palindrome.

2. Input:
   123
   456
   Output:
   Sum of numbers: 579
   579 is not a palindrome.

23)Write a program that calculates the sum of the Fibonacci series up to a given number of terms and checks if the sum is a perfect number or not.

Input Format

- N (The number of terms in the Fibonacci series)

Sample test case:

1. Input:
   12
   Output:
   Sum of Fibonacci series up to 5 terms: 12
   12 is a perfect number.

2. Input:
   6
   Output:
   Sum of Fibonacci series up to 6 terms: 20
   20 is not a perfect number.

24)Write a program to print the following number triangle pattern

Input Format

- N (size of pattern)

Sample test case:

1. Input:
   5
   Output:

```
    A
   BB
  CCC
 DDDD
EEEEE
```

25)Craft a program that takes two user-input integers, m and n, representing the lower and upper bounds of a range respectively. The program should then identify and print all the perfect numbers within that range. A perfect number is a positive integer that is equal to the sum of its proper divisors, excluding itself.

Sample input:
Enter the lower bound (m): 1
Enter the upper bound (n): 1000

Sample output:

```
Perfect numbers between 1 and 1000 are:
6
28
496
```

26)Craft a program that prompts the user to input two integers, m and n. representing the lower and upper bounds of a range respectively. The program should then calculate and print the sum of all cubes within this range, inclusive of the endpoints m and n.

Sample input:

```
Enter the lower bound (m): 2

Enter the upper bound (n): 5
```

Sample output:

```
The sum of cubes between 2 and 5 is: 224
```

27)Write a program that checks whether a given number is a Betrothed number or not. Betrothed numbers are two numbers where the sum of the proper divisors of each is one more than the other number.

Two numbers A and B are betrothed if:

- The sum of the proper divisors of A is equal to B+1.
- The sum of the proper divisors of B is equal to A+1.

Input Format

Two integers, A and B (The numbers to check for being betrothed)

Sample Input 1:
48 (Value of A)
75(Value of B)
Sample Output 1:
48 and 75 are betrothed numbers.
Sample Input 2:
6 (Value of A)
28 (Value of B)
Sample Output 2:
6 and 28 are not betrothed numbers.

28)Write a program that calculates the count of digits in a given number.

Input Format

N (The number to count the digits of)

Sample Input 1:

12345

Sample Output 1:

5

Sample Input 2:

100

Sample Output 2:

3

29)Write a program that calculates the sum of the digits in a given number.

Input Format

N (The number to calculate the sum of its digits)

Sample Input 1:

12345

Sample Output 1:

15

Sample Input 2:

1001

Sample Output 2:

2

30)Write a program that calculates the sum of the even digits in a given number.

Input Format

N (The number to calculate the sum of its even digits)

Sample Input 1:

129456

Sample Output 1:

12

Sample Input 2:

1466

Sample Output 2:

16

31)Write a program that reverses the given number, considering the input as a whole number.

Input Format

- N (The number to reverse)

Sample Input 1:

8745

Sample Output 1:

5478

Sample Input 2:

123

Sample Output 2:

321

32)Write a program that rotates the given number once in the clockwise direction.

Input Format

N (The number to rotate)

Sample Input 1:

12345

Sample Output 1:

51234

Sample Input 2:

8754

Sample Output 2:

4875

33)Write a C program that finds and prints all twin prime pairs within a given range [a, b], where a and b are provided by the user. Twin primes are pairs of prime numbers (p, q) where q = p + 2 and both p and q are prime numbers.

Sample Input1:

Enter the range (a b): 10 20

Sample Output1:

Twin prime pairs in the range [10, 20] are:

(11, 13)

(17, 19)

Sample Input2:

Enter the range (a b): 30 50

Sample Output2:

Twin prime pairs in the range [30, 50] are:

(41, 43)

34)Write a program in C to display a pattern like a right angle triangle using an asterisk.

```
*
* *
* * *
* * * *
```

Sample Input 1:

3

Sample Output 1:

```
*
* *
* * *
```

Sample Input 2:

6

Sample Output 2:

```
*
* *
* * *
* * * *
* * * * *
* * * * * *
```

35)Write a program in C to display the n terms of a harmonic series and their sum.

1 + 1/2 + 1/3 + 1/4 + 1/5 ... 1/n terms

Sample Input 1:

Input the number of terms : 5

Sample Output 1:

1/1 + 1/2 + 1/3 + 1/4 + 1/5

Sum of Series upto 5 terms : 2.283334

36)Write a program in C to display the cube of the number up to an integer.

Sample Input 1:

Input the number of terms : 5

Sample Output 1:

1 8 27 64 125

Sample Input 2:

Input the number of terms : 10

Sample Output 2:

1 8 27 64 125 216 343 512 729 1000

37) Fibonacci Series

Write a C program to print the first n numbers of the Fibonacci series, where n is provided by the user.

Sample Input1:

5

Sample Output1:

0 1 1 2 3

Sample Input1:

7

Sample Output1:

0 1 1 2 3 5 8

37)Inverted Right-Angled Triangle Pattern

Write a C program to print an inverted right-angled triangle pattern of stars (*). The number of rows should be specified by the user.

Sample Input1:

Enter the number of rows: 4

Sample Output1:

```
****

***

**
```

```
*
```

38)Square Pattern

Write a C program to print a square pattern of stars (*). The side length of the square should be specified by the user.

Sample Input1:

Enter the side length of the square: 4

Sample Output1:

```
****
****
****
****
```

39)Pyramid Pattern

Write a C program to print a pyramid pattern of stars (*). The number of rows in the pyramid should be specified by the user.

Sample Input1:

Enter the number of rows: 5

Sample Output1:

```
    *
   ***
```

```
  *****
 *******
*********
```

38)Hollow Square

Print a square pattern of asterisks (*) with a hollow interior filled with spaces ().

TEST CASE 1:

Side Length: 3:

Output:

```
***
* *
***
```

TEST CASE 2:

Side Length: 5 (Even):

Output:

```
*****
*   *
*   *
*   *
*****
```

TEST CASE 3:

Side Length: 6 (Odd):

Output:

```
******
*    *
*    *
*    *
*    *
******
```

39. Largest Number after digit swaps by parity

You are given a positive integer num. You may swap any two digits of num that have the same parity (i.e. both odd digits or both even digits).

Return the largest possible value of num after any number of swaps.

Constraints:

1 <= num <= 109

TEST CASE 1:

Input: num = 1234

Output: 3412

Also note that we may not swap the digit 4 with the digit 1 since they are of different parities.

TEST CASE 2:

Input: num = 65875

Output: 87655

TEST CASE 3:

Input: num = 6587

Output: 8765

41. Floyd's Triangle

Write a program to print Floyd's Triangle.

Floyd's Triangle is a triangular pattern of numbers where the numbers are arranged in a sequential manner, starting from 1 and incrementing by 1 for each subsequent row. The pattern is formed by printing the numbers in a triangular shape.

TEST CASE 1:

Input : 5

Output:

```
1
01
```

```
101
0101
10101
```

TEST CASE 2:

Input : 7

Output:

```
1
01
101
0101
10101
010101
1010101
```

TEST CASE 3:

Input : -1

Output:

No output

43. Program to display the sum of the series [9 + 99 + 999 + 9999 ...].

Calculate the sum of a series where each term is formed by concatenating the digit 9 to the previous term.

Print Invalid input if the input is <=0.

TEST CASE 1:

Input:

Number or terms :5

Output :

```
9 99 999 9999 99999
The sum of the series = 111105
```

TEST CASE 2:

Input:

Number or terms :3
Output :
9 99 999
The sum of the series = 1107

TEST CASE 3:

Input:
Number or terms :0
Output :
Invalid Input

44)