

## ✓ Better programmer tips

1. Proper naming should be given to variables
2. Proper spacing

### Question 1

Write a Python Program to find equivalent binary number of a given decimal number

**Sample Input:**

20

**Sample Output:**

Binary Equivalent of 20 is 10100

```
1 def binary_conversion(num):
2     binary = ""
3     while(num != 0):
4         remainder = num % 2
5         #binary = binary + remainder
6         binary += str(remainder)
7         num = num // 2
8     return int( binary[::-1] )
9 #driver code
10 num = int(input())
11 print( binary_conversion(num) )
```

20

10100

**Q2. Problem Statement:**

You are given a 0-indexed array of strings `words` and a character `x`.

Return an array of indices representing the words that contain the character `x`.

Note that the returned array may be in any order.

**Example 1:**

**Input:** `words = ["leet","code"], x = "e"`

**Output:** `[0, 1]`

**Explanation:** "e" occurs in both words: "leet", and "code". Hence, we return indices 0 and 1.

**Example 2:**

**Input:** `words = ["abc","bcd","aaaa","cbc"], x = "a"`

**Output:** `[0, 2]`

**Explanation:** "a" occurs in "abc", and "aaaa". Hence, we return indices 0 and 2.

```

1 def matching_indices(words):
2     indices = []
3     for i in range(0, len(words)):
4         if x_value in words[i]:
5             indices.append(i)
6     return indices
7 #driver code
8 words = list(input().split(" "))
9 x_value = input()
10 print( matching_indices(words) )

```

```

➞ abc bcd aaaa cbc
a
[0, 2]

```

**Q3. Problem Statement:**

There is a programming language with only four operations and one variable `X`:

`++X` and `X++` increments the value of the variable `X` by 1.

`--X` and `X--` decrements the value of the variable `X` by 1.

Initially, the value of `X` is 0. Given an array of strings `operations` containing a list of operations, return the final value of `X` after performing all the operations.

**Input:** `operations = ["--X","X++","X++"]`

**Output:** 1

**Explanation:** The operations are performed as follows:

Initially, `X = 0`.

`--X`: `X` is decremented by 1, `X = 0 - 1 = -1`.

`X++`: `X` is incremented by 1, `X = -1 + 1 = 0`.

`X++`: `X` is incremented by 1, `X = 0 + 1 = 1`

```
1 def x_operations(operations):
2     X_value = 0
3     for operation in operations:
4         if(operation == "X++" or operation == "++X"):
5             X_value += 1
6         elif(operation == "X--" or operation == "--X"):
7             X_value -= 1
8         else:
9             continue
10    return X_value
11 #driver code
12 operations = list(input().split(" "))
13 print( x_operations(operations) )

--X X++ ++X
1
```

#### Question 4

Write a Python Program for Fibonacci series (0,1,1,2,3,5,8,13,21.....)

**Sample Input:**

5

**Sample Output:**

The fibonacci series is  
0 1 1 2 3

```
1 def fibonacci_series(number):
2     num1, num2 = 0, 1
3     result = []
4     for i in range(1, number + 1):
5         result.append(num1)
6         num3 = num1 + num2
7         num1 = num2
8         num2 = num3
9     return result
10 #driver code
11 number = int(input("Enter the number: "))
12 list_values = fibonacci_series(number)
13 for value in list_values:
14     print(value, end = " ")
15
```

```
Enter the number: 10
0 1 1 2 3 5 8 13 21 34
```

### Question 5

Write a Python Program to print Floyd's Triangle

**Sample Input:**

5

**Sample Output:**

```
1
2  3
4  5  6
7  8  9  10
11 12 13 14 15
```

```

1 def floyds_triangle(rows):
2     value = 1
3     for i in range(1, rows + 1):
4         for j in range(1, i + 1):
5             print(value, end = " ")
6             value = value + 1
7         print()
8 #driver code
9 rows = int(input("Enter the rows: "))
10 floyds_triangle(rows)
11

```

Enter the rows: 5

```

1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

```

## Question 6

Write a Python Program to print the following pattern.

**Sample Input:**

3

**Sample Output:**

```

*
***
*****
***
*

```

```

1 #Upper part (triangle pattern)
2 rows = int(input())
3 for i in range(1, rows + 1, +1):
4     print((rows - i) * " " + (2 * i - 1) * "*")

```

```

5
*
***

```

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

```
1 #Lower part (triangle pattern)
2 rows = int(input())
3 for i in range(rows, 0, -1):
4     print((rows - i) * " " + (2 * i - 1) * "*")
```

```
5
*****
*****
*****
***
*
```

```
1 #Diamond pattern
2 rows = int(input())
3 for i in range(1, rows + 1, +1):
4     print((rows - i) * " " + (2 * i - 1) * "*")
5 for i in range(rows - 1, 0, -1):
6     print((rows - i) * " " + (2 * i - 1) * "*")
```

```
3
*
***
*****
***
*
```

## Question 7

Write a Python Program to print the following pattern.

**Sample Input:**

3

**Sample Output:**

```
1
222
33333
222
1
```

```
1 #Diamond number pattern
2 rows = int(input())
3 for i in range(1, rows + 1, +1):
4     print((rows - i) * " " + (2 * i - 1) * str(i))
5 for i in range(rows - 1, 0, -1):
6     print((rows - i) * " " + (2 * i - 1) * str(i))
```

```
5
  1
 222
33333
4444444
555555555
4444444
33333
 222
  1
```

**Question 8**

Write a Python Program to print the following series up to N terms  
20 19 17 14 10 5.....

**Sample Input:**

9

**Sample Output:**

20 19 17 14 10 5 -1 -8 -16

```
1 def number_series(num):  
2     value = 20  
3     for i in range(1, num + 1):  
4         print(value, end = " ")  
5         value = value - i  
6 #driver code  
7 num = int(input("Enter the number: "))  
8 number_series(num)
```

```
Enter the number: 5  
20 19 17 14 10
```



**Question 9**

Write a Python Program to print the following series up to N terms  
0 2 8 14 24 34.....

**Sample Input:**

10

**Sample Output:**

0 2 8 14 24 34 48 62 80 98

```
1 def square_series(number):  
2     for value in range(1, number + 1):  
3         if value % 2 == 0:  
4             print(value * value - 2, end = " ")  
5         else:  
6             print(value * value - 1, end = " ")  
7 #driver code  
8 number = int(input("Enter the number: "))  
9 square_series(number)
```

Enter the number: 10

0 2 8 14 24 34 48 62 80 98

## Question 10

Write a program to find the sum of elements in a 2D list.

```
1 def matrix_sum():
2     rows = int(input("Number of rows: "))
3     cols = int(input("Number of columns: "))
4     matrix = []
5     for i in range(0, rows):
6         row_matrix = list(map(int, input().split(' ')))[:cols]
7         matrix.append(row_matrix)
8
9     #sum logic
10    total_sum = 0
11    for i in range(0, rows):
12        for j in range(0, cols):
13            total_sum += matrix[i][j]
```