Week-3: Activity Questions

Problem 1

Question

Write a code using a while loop that adds all odd numbers from 1 to 100 (inclusive).

Answer

```
n=1
total=0
while(n<=100):
    if n%2!=0:
        total+=n
    n+=1
print(total)
```

or

```
n=1
total=0
while(n<=100):
    total+=n
    n+=2
print(total)
```

Problem 2

Question

Write a program using loop to print first 10 even numbers in reverse order.

Answer

Using for loop:

```
for num in range(20, 0, -2):
      print(num)
```

Using while loop

```
num = 20
while num > 0:
      print(num)
      num = num - 2
```

Problem 3

Question

Write a program to accept the positive integer `n` from the user and print the average of all number's factorial from 1 to `n` .

Answer

```
n = int(input())
total = 0
for i in range(1,n+1):
    fact = 1
    j = 1
    while j <= i:
        fact *= j
        j += 1
    total += fact

avg = total / n
print(avg)
```

or

```
n = int(input())
i = 1
total = 0
while i <= n:
    fact = 1
    for j in range(1,i+1):
        fact *= j
    total += fact
    i += 1
avg = total / n
print(avg)
```

Problem 4

Question

Write a program to accept the positive integer `n` from the user and print counting of numbers which are not prime from 1 to `n`.

Answer

```
n = int(input())
count = 1
for num in range(2,n+1):
    flag = True
    for j in range (2,num):
        if num % j == 0:
            flag = False
            break
    if flag == False:
        count += 1
print(count)
```

Problem 5

Question

Write a program to accept the string `s` from the user and print all alphabets in one line separated by , before first occurrence of vowel .

Answer

```
s = input()
pos = 0
vowels = 'aeiouAEIOU'
while pos < len(s) and s[pos] not in vowels:
    if s[pos].isalpha() == True:
         print(s[pos] , end = ",")
    pos += 1
```

Problem 6

Question

Write a program to accept the two positive integers `start` and `stop` where `0< start<stop<=100` print all numbers from start to stop both inclusive with following constraints.

- If the number is divisible by 3 print `Divisible by 3` at the place of number.
- If the number is divisible by 5 print `Divisible by 5` at the place of number.
- If the number is divisible by 10 print `Divisible by 10` at the place of number.
- If the number is divisible by any of two from 3 ,5 and 10 , print nothing, just skip.
- If the number is divisible by all (3 , 5 and 10) , stop printing the number.
- If the number is not divisible by any of (3 , 5 and 10) , just print the number as it is.

Answer

```
start=int(input())
stop=int(input())

for i in range (start,stop+1):
    if i%3==0 and i%5==0 and i%10==0:
        break
    elif (i%3==0 and i%5==0) or (i%5==0 and i%10==0) or (i%3==0 and
i%10==0):
        pass
    elif i%3==0:
        print("Divisible by 3")
    elif i%5==0:
        print("Divisible by 5")
    elif i%10==0:
        print("Divisible by 10")
    else:
        print(i)
```

Problem 7

Question

Accept the positive integer `n` and display the cube of the number up to a given integer `n` in the following pattern using formatted string.

Input

```
6
```

Output

```
Current Number is : 1  and the cube is 1
Current Number is : 2  and the cube is 8
Current Number is : 3  and the cube is 27
Current Number is : 4  and the cube is 64
Current Number is : 5  and the cube is 125
Current Number is : 6  and the cube is 216
```

Answer

```
n=int(input())
for i in range(1,n+1):
    print("Current Number is : {}  and the cube is {}".format(i,i**3))
```

or

```
n=int(input())
for i in range(1,n+1):
    print(f"Current Number is : {i}  and the cube is {i**3}")
```

or

```
n=int(input())
for i in range(1,n+1):
    print("Current Number is : %d  and the cube is %d"%(i,i**3))
```

Problem 8

Question

Run the following code and observe the output.

Code

```
1 print("The sum of {} , {} and {} is {}".format(1,2,3,6) )
2 print("The sum of {0} , {1} and {2} is {3}".format(1,2,3,6) )
3 print("The sum of {0} , {1} and {c} is {d}".format(1,2,c=3,d=6) )
4 print("The sum of {a} , {b} and {c} is {d}".format(a=1,b=2,c=3,d=6) )
```

Answer

line 1, if we do not assign any position inside `{}` then the interpreter will assign a position from 0 to 3 in `{}` from left to right in the string.

line 2, we can manually assign position inside `{}` and order is not mandatory.

line3, we can assign position and keyword inside `{}` at the same time. but inside `format()` keyword argument should be defined after positional argument.

line 4, we can manually assign keywords inside `{}` order is not mandatory.

Problem 9

Question

Write a code to accept the positive integer `n` from user and print average of `1 + 11 +111 + 1111 + ..n terms`.

Input

```
6
```

so series will be `1 + 11 + 111 + 1111 + 11111 + 111111`

Output

```
20576.0
```

Answer

```python
n = int(input())
num = "1"
total = 0
for i in range(1 , n + 1):
    total += int(num * i)
avg = total / n
print(avg)
```

Problem 10

Question

Place a string and `{}` inside `""` in line number 3 of the code in the appropriate place and fill the argument inside `.format()'s` parenthesis so that the output prints in the following manner.

```
1 name = input()
2 age = input()
3 message="".format()
4 print(message)
```

Input

```
1 Rahul21
```

Output

```
1 Hii!, I am Rahul, I am 21 year's old.
```

Answer

```
1 name = input()
2 age = input()
3 str="Hii!, I am {0}, I am {1} year's old.".format(name,age)
4 print(str)
```

Problem 11

Question

Write a code to accept positive integer `n` from user(`n<10`) and print following pattern.

Input

```
5
```

Output

```
5 4 3 2 1 2 3 4 5
  4 3 2 1 2 3 4
    3 2 1 2 3
      2 1 2
        1
```

Answer

```python
n=int(input())
for i in range(0,n):
    print(" "*2*i,end="")
    for j in range(n-i,1,-1):
        print(j,end=" ")
    for k in range(1,n-i+1):
        if k == n-i:
            print(k,end="")
        else:
            print(k,end=" ")

    print(" "*2*i,end="")
    print()
```

Problem 12

Question

Write a code to accept the string of length 10 from the user and print `True` if string has any character occurring 5 times consecutively in it, otherwise print `False`.

Answer

```
s=input()
valid=False
for i in range(0,len(s)-4):
    count= s.count(s[i])
    if count > 4 :
        if s[i:i+5].count(s[i])==5:
            valid=True
    if valid==True:
        break
print(valid)
```

or

```
s=input()
valid=False
for i in range(0,len(s)-4):
    count= s.count(s[i])
    if count > 4 :
        valid = True
        for i in range(i , i+4):
            if s[i] != s[i+1]:
                valid = False
                break
    if valid==True:
        break
print(valid)
```

Problem 13

Question

Write a code to accept the positive integer `n` from the user and print the following pattern.

Input

```
5
```

Output

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Answer

```
n= int(input())
for i in range(1,n+1):
    for j in range(1,i+1):
        print(j,end=" ")
    print()
```

Problem 14

Question

Write a code to accept a string as input and reverse it using loop:

Answer

```
st = input()
reverse = ''
for char in st:
    reverse = char + reverse
print(reverse)
```

Problem 15

Question

Write a code to accept a string as input and determine if it is a palindrome or not.

Answer

```
st = input()
reverse = ''
for char in st:
    reverse = char + reverse
if st == reverse:
    print("Yes,string is palindrome")
else:
    print("No,string is not palindrome")
```

Problem 16

Question

Write a code to accept the name of a person as input and print the initials as output. Assume that the name will be of this form: `<first name> <last name>`. Also assume that the first name and last name will be a single word, and there will be exactly one space between the two names. For example, if the input is `Rohit Sharma`, the output should be `RS`.

Answer

```
name = input()
res = ''
first_char = True
for char in name:
    if first_char == True:
        res = res + char
        first_char = False
    if char == ' ':
        first_char = True
print(res)
```

Problem 17

Question

Write a code for the following sequence:

```
|0||0|||0|
```

This sequence should be understood as follows: `||` represents an empty box. `|0|` represents a box that has an egg in it. These boxes are placed side by side so that the right wall of a box overlaps with the left wall of the box immediately to its right. For any arbitrary sequence given as input, your task is to find:

- total number of boxes
- number of empty boxes
- number of boxes that have eggs in them

In the sequence given above, there are six boxes. Three are empty and the other three are non-empty

Answer

```python
box = input()
total_box = 0
egg = 0
for char in box:
    if char == '0':
        egg = egg + 1
    if char == "|":
        total_box = total_box + 1
total_box = total_box - 1
print("The total number of box =",total_box)
print("The total number of empty box =",total_box - egg)
print("Tumber of boxes that have eggs in them =",egg)
```

Problem 18

Question

You are given the results of a sequence of matches played by India in ODIs. A win is represented by `'W'` and a loss is represented by `'L'`. A winning streak is a string of consecutive wins. For example, if India has played five matches with the following results - `'WLWWWL'` - then it has a three-match streak. Accept the result-sequence as input and find the longest streak in it.

Answer

```
match_result = input()
max_streak = 0
count_win_streak = 0
for char in match_result:
    if char == 'W':
       count_win_streak = count_win_streak + 1
       if max_streak < count_win_streak:
           max_streak = count_win_streak
    if char == 'L':
        count_win_streak = 0
print("the longest win streak =",max_streak)
```

Problem 19

Question

Write a code to accept the sequence of positive integers in sorted order(either in non ascending order or non descending order) ending with `-1` and at least two integers must be before `-1` . In Output print the number of distinct elements in the sorted sequence before `-1`.

Sample Input - 1

```
1
2
-1
```

Sample Output - 1

```
2
```

Sample Input - 2

```
7
6
6
5
5
5
-1
```

Sample Output - 2

```
3
```

Answer

```
curr=int(input())
distinct_num = 1
while ( curr != -1 ):
    prev = curr
    curr=int(input())
    if ( prev != curr and curr != -1):
        distinct_num += 1
print(distinct_num)
```

Problem 20

Question

Write a code to accept the sequence of positive integers ending with `-1` and at least two distinct integers must be before `-1` .

In Output:

- Print `asc` if sequence before `-1` is sorted in ascending order or non descending order.
- Print `desc` if sequence before `-1` is sorted in descending order or non ascending order.
- Print `False` if sequence before `-1` is not sorted in any order.

Sample Input - 1

```
1
2
2
3
3
4
-1
```

Sample Output - 1

```
asc
```

Sample Input - 2

```
7
6
6
5
5
5
-1
```

Sample Output - 2

```
desc
```

Sample Input - 3

```
3
4
4
5
6
2
8
-1
```

Sample Output - 3

```
False
```

Answer

```
asc_sort=True
desc_sort=True
curr=int(input())
while ( curr != -1 ):
    prev = curr
    curr=int(input())
    if ( curr != -1 and prev > curr and asc_sort == True) :
        asc_sort=False
    if (curr != -1  and prev < curr and desc_sort == True) :
        desc_sort=False
if asc_sort==True:
    print('asc')
elif desc_sort==True:
    print('desc')
else:
    print('False')
```

Problem 21

Question

Write a program to print the product of the digits of a number entered by the user. Number can be either positive or a negative.

Public

Input	OUTPUT
12345	120
999	729
1000	0
-12345	120

Answer

```
num = int(input())
if num < 0:
    num = -num
product = 1
while(num):
   remainder = num % 10
   product = product * remainder
   num = num // 10
print(product)
```

Problem 22

Question

Write a program to find highest common factor (HCF) of two numbers.

Answer

```
num_1 = int(input())
num_2 = int(input())
while num_2 != 0:
  num_1, num_2 = num_2, num_1 % num_2
print(num_1)
```

Problem 23

Question

Write a Program to find the sum of an below series. Consider x and n as positive integers.

$1 + x^2 + x^3 + \ldots + x^n$

Answer

```python
num_of_terms = int(input())
x = int(input())
sum_series = 1
for power in range(2, num_of_terms + 1):
    sum_series = sum_series + (x ** power)
    #print(sum_series)
print("{}".format(sum_series))
```

Problem 24

Question

Write a program for the below scenario:

The sum of a harmonic series approaches infinity when as the number of terms increases.

$1 + 1/2 + 1/3 + 1/4 + \ldots$

How many terms are required to reach the value near to `10`. Consider the value for `tolerance = 0.05`.

Answer

11764

Solution

```
term = 1
count = 0
sum_series = 0
while term:
    sum_series = sum_series + (1 / term)
    count += 1
    term += 1
    if 10 - sum_series < tolerance:
        break
print("{:.2f}".format(sum_series))
print("{}".format(count))
```

Problem 25

Question

Write a program for harmonic mean of first `n` numbers. Accept `n` as an input from the user. Round off the answer to `2` decimal number.

For an example , the harmonic mean for first `5` numbers can be written as.

$$\text{harmonic-mean} = \frac{5}{1/1+1/2+1/3+1/4+1/5}$$

Answer

```python
num_of_terms = int(input())
term = 1
count = 0
sum_series = 0
for term in range(1, num_of_terms + 1):
    sum_series = sum_series + 1 / term
    count += 1
    term += 1
print("{:.2f}".format(count / sum_series))
```

Problem 26

Question

Write a program to number guessing game using loops. The program arbitrarily choses a number in the range 1 to 100 (inclusive) and asks user to enter a guessed number. The program repeatedly asks user to guess the number 5 times. For each guess, the program says `TOO HIGH` or `TOO LOW` or `GOOD` based on the below table.

Difference	output
`actual_number - guess_number = 0`	`EXCELLENT`
`(actual_number - guess_number <= 10) or (guess_number - actual_number <= 10)`	`GOOD`
`guess_number - actual_number > 10`	`TOO_HIGH`
`actual_number - guess_number > 10`	`TOO_LOW`

Answer

```python
import random
num_of_guesses = 5
actual_number  = random.randint(1, 101)
#print(actual_number)
for count in range(num_of_guesses):
    guess_number  = int(input())
    if actual_number == guess_number:
        print("EXCELLENT")
    elif abs(actual_number - guess_number <= 10):
        print("GOOD")
    elif actual_number - guess_number > 10:
        print("TOO_LOW")
    else:
        print("TOO_HIGH")
```

Problem 27

Question

Write a program to print the Fibonacci series of n terms where n is always greater than or equal to 2.

Answer

```
n = int(input("Enter any number: "))
a, b = 0, 1
print(a, b, end = ' ')
for i in range(3, n+1):
    fib = a + b
    print(fib, end = ' ')
    a = b
    b = fib
```

Problem 28

Question

Two numbers n1 and n2 are said to be same if they have equal number of digits in them. Write a program to check whether n1 and n2 are same. n1 and n2 are positive integers entered by the user without converting the number to string.

Answer

```
n1 = int(input("Enter 1st number: "))
n2 = int(input("Enter 2nd number: "))
count1 = 0
count2 = 0
while (n1 > 0):
    count1 = count1 + 1
    n1 = n1 // 10
while (n2 > 0):
    count2 = count2 + 1
    n2 = n2 // 10
if (count1 == count2):
    print(True)
else:
    print(False)
```

Problem 29

Question

Write a program to print the first and last digits of a number without converting it to string.

Answer

```
n = int(input("Enter any number: "))
x = n
last_digit = n % 10
first_digit = 0
while(n > 0):
    first_digit = n % 10
    n = n // 10
print("first digit is: ",first_digit)
print("last digit is: ",last_digit)
```

Below 4 Questions are MCQ/MSQ Type.

Problem 1

Question

You want to make sure that a set of statements are executed at least once before a condition is evaluated. Which loop will be preferred in this situation. This is a Multiple Choice Question (MCQ).

(a) for

(b) do-while

(c) while

(d) None of these

Answer

(d)

Problem 2

Question

How many time the loop will run. This is a Multiple Select Question (MSQ).

```
while x:
    Print("Python")
```

(a) Will run exactly `1` time for any value of `x`

(b) Will run `x` times where `x` is a positive integer

(c) Will run `0` rimes when `x = 0`

(d) Will run indefinitely when `x` is a positive or negative integer

(e) Will run indefinitely when `x = True`

Answer

(c), (d), (e)

Problem 3

Question

What will the output of the following code. This is a Multiple Choice Question (MCQ).

```
x = 2
y = 5
example = f'Two plus five is {x + y} and not {3 * (x + y)}'
print(example)
```

(a) `Two plus five is {x + y} and not {3 * (x + y)}`

(b) `Two plus five is 2 + 5 and not 3 * (2 + 7)`

(c) `Two plus five is {2 + 5} and not {3 * (7)}`

(d) `Two plus five is 7 and not 21`

Answer

(d)

Problem 4

Question

What will be the output of the following code.

```
a = "28"
b = "8"
c = "India"
print("There are {} states and {} union territories in {}.".format(a,b,c))

```

(a) `"There are {} states and {} union territories in {}."`

(b) `There are {28} states and {8} union territories in {India}.`

(c) `There are 28 states and 8 union territories in India.`

(d) `"There are 28 states and 8 union territories in India."`

Answer

(c)