Week-3 Practice Assignment (Programming)

Problem 1

Question

Write a program to print the numbers which are divisible by 12 and 13 in range of [1000 - 2000]

Answer

```
for i in range(1000, 2001, 1):
    if (i % 12 == 0 and i % 13 == 0):
        print(i)
```

Testcases

Public

Output

```
1092
1248
1404
1560
1716
1872
```

Private

Output

```
1092
1248
1404
1560
1716
1872
```

Solution

The range from 1000 to 2000 can be expressed as `range(1000, 2001, 1)`. For the inclusion of 2000, the end value of the range function is given as 2001. For each `i` the if-statement checks for the divisibility of 12 and 13. If that is satisfied, the number will be printed.

Problem 2

Question

Write a program to find the sum of the digits of the number got from the user.

Input	Output
`123456`	`21`
`67127`	`23`
`182638`	`28`

Answer 1

```python
n = int(input())
total = 0
while(n > 0):
    total = total + n % 10
    n = n // 10
print(total)
```

Answer 2

```python
n = input()
total = 0
for i in n:
    total = total + int(i)
print(total)
```

Testcases

Public

Input	Output
`123456`	`21`
`67127`	`23`
`182638`	`28`

Private

Input	Output
`000000`	`0`
`111111`	`6`
`1996`	`25`

Solution

The sum of all digits can be obtained by adding the digits from the last to the variable `total` using the modulo division by 10 and replacing the `n` by the quotient on division by 10. This will occur repeatedly when the `n` is greater than 0.

```
n = int(input())
total = 0
while(n > 0):
    total = total + n % 10
    n = n // 10
print(total)
```

For example:
if n = 1234

Iteration	`n > 0`	`n`	`n % 10`	`n // 10`	`total`
1	True	1234	4	123	4
2	True	123	3	12	7
3	True	12	2	1	9
4	True	1	1	0	10
5	False				

Another straight forward approach would be using a for-loop to iterate over the string input for each character and convert each character to integer and add that to `total`.

```
n = input()
total = 0
for i in n:
    total = total + int(i)
print(total)
```

Problem 3

Question

Write a program to find sum of all prime numbers between 1 to `n`, where `n` is a positive integer from user.

Input	Output
`10`	`17`
`100`	`1060`

Answer

```python
total = 0
n = int(input())
if (n == 0 or n == 1):
    total = 0
if (n == 2):
    total = 2
for i in range(2, n):
    for j in range(2, i):
        if (i % j == 0):
            break
    else:
        total = total + i
print(total)
```

Testcases

Public

Input	Output
`10`	`17`
`100`	`1060`

Private

Input	Output
`124`	`1593`
`0`	`0`
`1`	`0`

Solution

Any number which is divisible by 1 and itself alone are called prime number. Here, we have to find all prime numbers between 2 and `n`, `n` is the positive integer got from the user. If `n` is `0` or `1` then we are printing directly the value `0` to the screen using the if-statement.

```
if (n == 0 or n == 1):
    total = 0
#...
#...
print(total)
```

Otherwise, a for-loop with variable `i` is used to iterate between 2 and `n`, where the divisibility of `i` by any number less than `n` is checked by the if-statement. If `i` is divisible by any previous number of `i` from 2, then that `i` is exempted from adding to the variable `total` which will hold the sum of prime of numbers up to `n` at the end of the execution using the `flag` variable.

```
for i in range(2, n):
    flag = True
    for j in range(2, i):
        if (i % j == 0):
            flag = False
            break
    if flag:
        total = total + i
print(total)
```

Problem 4

Question

Write a python program to print the following pattern. The number of `*` in the first line taken from the user input (odd number). The example is given for the user input `15`.

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

Answer

```
n = int(input())
i = 0
while i < n:
    j = 0
    while j < n:
        if i==j or i+j==n-1 or j == n//2 or i == n//2 or i == 0 or i == n-1
or j == 0 or j == n-1:
            print('*', sep='', end='')
        else:
            print(' ', sep='', end='')
        j = j + 1
    i = i + 1
    print()
```

Testcases

Public

Private

Input 1

```
13
```

Output 1

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

Input 2

```
5
```

Output 2

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

Input 3

```
3
```

Output 3

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

Input 4

```
1
```

Output 4

```
*
```

Input 5

```
9
```

Output 5

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

Solution

The above pattern printed using a nested loop where the outer-loop for line and inner-loop for the character printed on the line.

For each line `*` has to be printed in some pattern which is determined by the if-statement. If any condition returns True `*` is printed, otherwise " " (space) is printed.

Condition	Satisfies
`i==j`	Line from top left to bottom right
`i+j==n-1`	Line from bottom left to top right
`i == 0`	Top horizontal line
`j == n//2`	Middle horizontal line
`i == n-1`	Bottom horizontal line
`j == 0`	Left vertical line
`i == n//2`	Middle vertical line
`j == n-1`	Right vertical line

Tags

Problem 5

Question

Write a python program to print all the combination that satisfies $x^4 + y^3 = z^2$.
Where,

1. x, y and z should be distinct, positive and less than `n` obtained from the user.
2. $x < y < z$
3. The output should be printed in the ascending order of x

Hint: Use nested loop

Answer

```
n = int(input())
for i in range(1, n):
    for j in range(i+1, n):
        for k in range(j+1, n):
            if i**4 + j**3 == k**2:
                print(i, j, k)
```

Testcases

Public

Input 1

```
100
```

Output 1

```
1 2 3
5 6 29
6 9 45
7 15 76
```

Input 2

```
10
```

Output 2

```
1 2 3
```

Private

Input 1

```
73
```

Output 1

```
1 2 3
5 6 29
6 9 45
```

Input 2

```
200
```

Output 2

```
1 2 3
5 6 29
6 9 45
7 15 76
8 32 192
9 27 162
```

Solution

Here a triple nested loop is required for the three variables `i`, `j` and `k`. In the outermost for-loop, `i` takes the value from the range of 1 to `n`. In the intermediate for-loop, the variable `j` takes the value from `i+1` to `n`. In the innermost for-loop, `k` takes the value from `j+1` to `n`. These are to ensure the conditions that x, y, z should be distinct, and $x < y < z < n$.
Within the innermost loop, the expression `i**4 + j**3 == k**2` is checked and printed if that is True.

```
for i in range(1, n):
    for j in range(i+1, n):
        for k in range(j+1, n):
            if i**4 + j**3 == k**2:
                print(i, j, k)
```

Tags

Problem 6

Question

Accept two strings from the user and remove all characters from the second string which are present in the first string.

Sample Input - 1

```
aeiou
this is python program
```

Sample Output - 1

```
ths s pythn prgrm
```

Sample Input - 2

```
lo
hello python
```

Sample Output - 2

```
he pythn
```

Sample Input - 3

```
hello python
```

Sample Output - 3

```
hello python
```

Answer

```
string1 = input()
string2 = input()
temp = ''
for i in range(0,len(string1)):
    for j in range(0,len(string2)):
        if (string1[i] == string2[j]):
            continue
        else:
            temp = temp + string2[j]
    string2 , temp = temp , ""
print(string2)
```

Testcases

Public

Input 1

```
aeiou
this is python program
```

Output 1

```
ths s pythn prgrm
```

Input 2

```
lo
hello python
```

Output 2

```
he pythn
```

Input 3

```
hello python
```

Output 3

```
hello python
```

Private

Input 1

```
abcdefghijkl
abcdmnop
```

Output 1

```
mnop
```

Input 2

```
@.
abc@onlinedegree.iitm.ac.in
```

Output 2

```
abconlinedegreeiitmacin
```

Input 3

```
abcdef
ijklmn
```

Output 3

```
ijklmn
```

Solution

- Accept two string from the user and assign to `string1` and `string2` variables.
- Initialize the `temp` variable with an empty string.
- In the first cycle of the outer loop (`i=0`) using a nested loop check for each index of `string2`, if the character of `string2` is not matched with character of `string1`, then appends this character of `string2` to `temp` string.
- In line 10, assign `temp` value to `string2` and assign an empty string to `temp` for the next cycle of the outer loop.
- So, in each cycle of outer loop, 1 character of `string1` will be removed from `string2`, then finally print the `string2` after the completion of loop.

Tags