

[Dashboard](#) / [My courses](#) / [PSP/PUP](#) / [Algorithmic Approach: Iteration control structures.](#) / [Week4 Coding](#)

| | |
|---------------------|-------------------------------|
| Started on | Sunday, 9 June 2024, 11:00 PM |
| State | Finished |
| Completed on | Sunday, 9 June 2024, 11:10 PM |
| Time taken | 9 mins 30 secs |
| Marks | 10.00/10.00 |
| Grade | 100.00 out of 100.00 |

Question 1

Correct

Mark 1.00 out of 1.00

Write a program that finds whether the given number N is Prime or not.

If the number is prime, the program should return 2 else it must return 1.

Assumption: $2 \leq N \leq 5000$, where N is the given number.

Example1: if the given number N is 7, the method must return 2

Example2: if the given number N is 10, the method must return 1

For example:

| Input | Result |
|-------|--------|
| 7 | 2 |
| 10 | 1 |

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 if n>1:
3     for i in range(2,n):
4         if (n%i)==0:
5             print("1")
6             break
7 else:
8     print("2")

```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 7 | 2 | 2 | ✓ |
| ✓ | 10 | 1 | 1 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2

Correct

Mark 1.00 out of 1.00

In mathematics, the factorial of a non-negative integer n , denoted by $n!$, is the product of all positive integers less than or equal to n . For example,

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

$$4! = 4 \times 3 \times 2 \times 1 = 24$$

$$9! = 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 362880$$

Write a program to find the factorial of a given number.

The given number will be passed to the program as an input of type int.

The program is expected to calculate the factorial of the given number and return it as an int type.

Assumptions for this program:

The given input number will always be greater than or equal to 1.

Due to the range supported by int. the input numbers will range from 1 to 12.

For example:

| Input | Result |
|-------|--------|
| 5 | 120 |
| 4 | 24 |
| 9 | 362880 |

Answer: (penalty regime: 0 %)

```

1 def factorial(n):
2
3     if n==0:
4
5         return 1
6
7     return n*factorial(n-1)
8
9 num=int(input())
10
11 print(factorial(num))
12

```

| | Input | Expected | Got | |
|---|-------|----------|--------|---|
| ✓ | 5 | 120 | 120 | ✓ |
| ✓ | 4 | 24 | 24 | ✓ |
| ✓ | 9 | 362880 | 362880 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3

Correct

Mark 1.00 out of 1.00

Write a program to return the nth number in the fibonacci series.

The value of N will be passed to the program as input.

NOTE: Fibonacci series looks like –

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, . . . and so on.

i.e. Fibonacci series starts with 0 and 1, and continues generating the next number as the sum of the previous two numbers.

- first Fibonacci number is 0,
- second Fibonacci number is 1,
- third Fibonacci number is 1,
- fourth Fibonacci number is 2,
- fifth Fibonacci number is 3,
- sixth Fibonacci number is 5,
- seventh Fibonacci number is 8, and so on.

For example:

| Input | Result |
|-------|--------|
| 1 | 0 |
| 4 | 2 |
| 7 | 8 |

Answer: (penalty regime: 0 %)

```

1 x=int(input())
2 a=0
3 b=1
4 p=[]
5 for i in range (0,x):
6     p.append(a)
7     c=a+b
8     a=b
9     b=c
10 print (p[x-1])
11

```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 1 | 0 | 0 | ✓ |
| ✓ | 4 | 2 | 2 | ✓ |
| ✓ | 7 | 8 | 8 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4

Correct

Mark 1.00 out of 1.00

Write a program to find the count of unique digits in a given number N. The number will be passed to the program as an input of type int.

Assumption: The input number will be a positive integer number ≥ 1 and ≤ 25000 .

For e.g.

If the given number is 292, the program should return 2 because there are only 2 unique digits '2' and '9' in this number

If the given number is 1015, the program should return 3 because there are 3 unique digits in this number, '1', '0', and '5'.

For example:

| Input | Result |
|-------|--------|
| 292 | 2 |
| 1015 | 3 |

Answer: (penalty regime: 0 %)

```

1 num=input()
2 #Count unique digits
3 unique_digits=set(num)
4 #Print the count of unique digits
5 print(len(unique_digits))

```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 292 | 2 | 2 | ✓ |
| ✓ | 1015 | 3 | 3 | ✓ |
| ✓ | 123 | 3 | 3 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5

Correct

Mark 1.00 out of 1.00

A Number is said to be Disarium number when the sum of its digit raised to the power of their respective positions becomes equal to the number itself. Write a program to print number is Disarium or not.

Input Format:

Single Integer Input from stdin.

Output Format:

Yes or No.

Example Input:

175

Output:

Yes

Explanation

$$1^1 + 7^2 + 5^3 = 175$$

Example Input:

123

Output:

No

For example:

| Input | Result |
|-------|--------|
| 175 | Yes |
| 123 | No |

Answer: (penalty regime: 0 %)

```

1  # Input number
2  num=int(input())
3  #Calculate the power of digits
4  power_sum=0
5  count=len(str(num))
6  temp=num
7  while temp > 0:
8      digit=temp%10
9      power_sum+=digit**count
10     temp//=10
11     count-=1
12 #Check the number is Disarium
13 if num==power_sum:
14     print("Yes")
15 else:
16     print("No")

```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 175 | Yes | Yes | ✓ |
| ✓ | 123 | No | No | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6

Correct

Mark 1.00 out of 1.00

Write a program to find the sum of the series $1 + 11 + 111 + 1111 + \dots + n$ terms (n will be given as input from the user and sum will be the output)

Sample Test Cases

Test Case 1

Input

4

Output

1234

Test Case 2

Input

6

Output

123456

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2 sum_series=0
3 for i in range(1,n+1):
4     term=int('1'*i)
5     sum_series+=term
6 print(sum_series)
```

| | Input | Expected | Got | |
|---|-------|----------|--------|---|
| ✓ | 4 | 1234 | 1234 | ✓ |
| ✓ | 6 | 123456 | 123456 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7

Correct

Mark 1.00 out of 1.00

Write a program to find the count of non-repeated digits in a given number N. The number will be passed to the program as an input of type int.

Assumption: The input number will be a positive integer number ≥ 1 and ≤ 25000 .

Some examples are as below.

If the given number is 292, the program should return 1 because there is only 1 non-repeated digit '9' in this number

If the given number is 1015, the program should return 2 because there are 2 non-repeated digits in this number, '0', and '5'.

If the given number is 108, the program should return 3 because there are 3 non-repeated digits in this number, '1', '0', and '8'.

If the given number is 22, the function should return 0 because there are NO non-repeated digits in this number.

For example:

| Input | Result |
|-------|--------|
| 292 | 1 |
| 1015 | 2 |
| 108 | 3 |
| 22 | 0 |

Answer: (penalty regime: 0 %)

```

1 x=input()
2 a=[]
3 for i in range (0,10):
4     b=0
5     for digit in x:
6         if int(digit)==i:
7             b=b+1
8     if b==1:
9         a.append(str(i))
10 print(len(a))

```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 292 | 1 | 1 | ✓ |
| ✓ | 1015 | 2 | 2 | ✓ |
| ✓ | 108 | 3 | 3 | ✓ |
| ✓ | 22 | 0 | 0 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **8**

Correct

Mark 1.00 out of 1.00

Given an integer N, check whether N the given number can be made a perfect square after adding to it.

Input Format:

Single integer input.

Output Format:

Yes or No.

Example Input:

24

Output:

Yes

Example Input:

26

Output:

No

For example:

| Input | Result |
|-------|--------|
| 24 | Yes |

Answer: (penalty regime: 0 %)

```

1 | x=int(input())
2 | a=x+1
3 | b=a**0.5
4 | if b.is_integer():
5 |     print("Yes")
6 | else:
7 |     print("No")

```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 24 | Yes | Yes | ✓ |
| ✓ | 26 | No | No | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **9**

Correct

Mark 1.00 out of 1.00

Given a number N, find the next perfect square greater than N.

Input Format:

Integer input from stdin.

Output Format:

Perfect square greater than N.

Example Input:

10

Output:

16

Answer: (penalty regime: 0 %)

```
1 x=int(input())
2 a=int(x**0.5)+1
3 b=a*a
4 print(b)
```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 10 | 16 | 16 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **10**

Correct

Mark 1.00 out of 1.00

Given a positive integer N, check whether it can be represented as a product of single digit numbers.

Input Format:

Single Integer input.

Output Format:

Output displays Yes if condition satisfies else prints No.

Example Input:

14

Output:

Yes

Example Input:

13

Output:

No

Answer: (penalty regime: 0 %)

```

1 N=int(input())
2 can_be_represented=False
3 for i in range (1,10):
4     if N%i==0 and N//i<10:
5         can_be_represented=True
6 if can_be_represented:
7     print("Yes")
8 else:
9     print("No")

```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 14 | Yes | Yes | ✓ |
| ✓ | 13 | No | No | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

◀ Week4_mcq

Jump to...

Strings ►