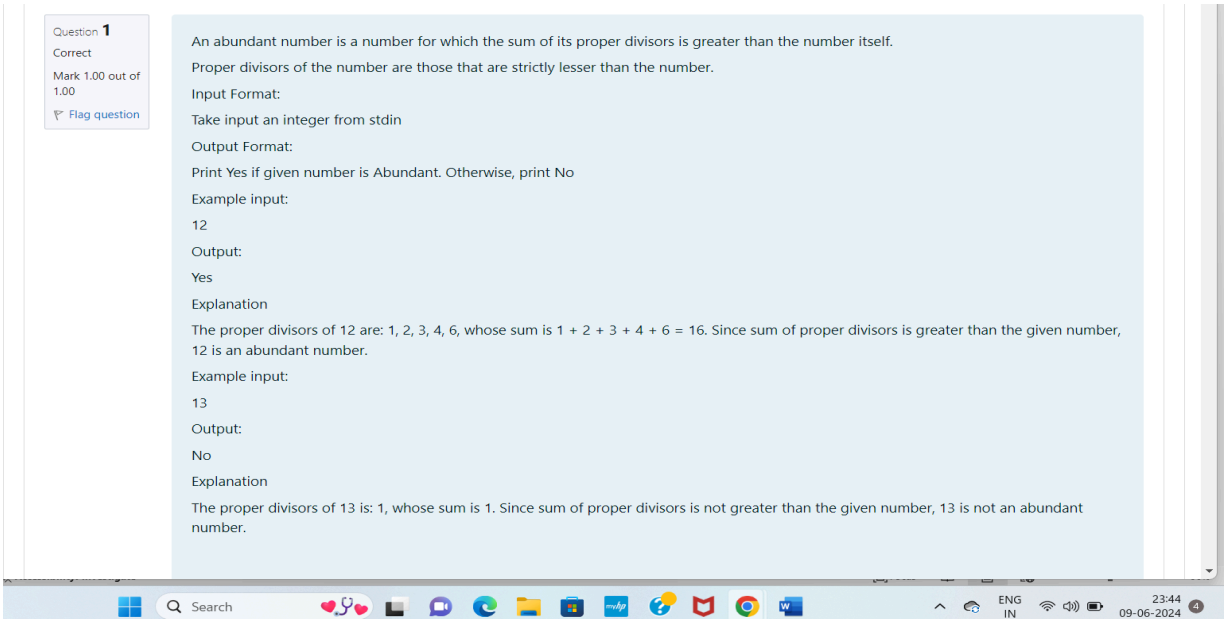
# WEEK 4

## Algorithmic Approach:Iteration and control structure



Question 1
Correct
Mark 1.00 out of 1.00
Flag question

An abundant number is a number for which the sum of its proper divisors is greater than the number itself.

Proper divisors of the number are those that are strictly lesser than the number.

Input Format:

Take input an integer from stdin

Output Format:

Print Yes if given number is Abundant. Otherwise, print No

Example input:

12

Output:

Yes

Explanation

The proper divisors of 12 are: 1, 2, 3, 4, 6, whose sum is 1 + 2 + 3 + 4 + 6 = 16. Since sum of proper divisors is greater than the given number, 12 is an abundant number.

Example input:

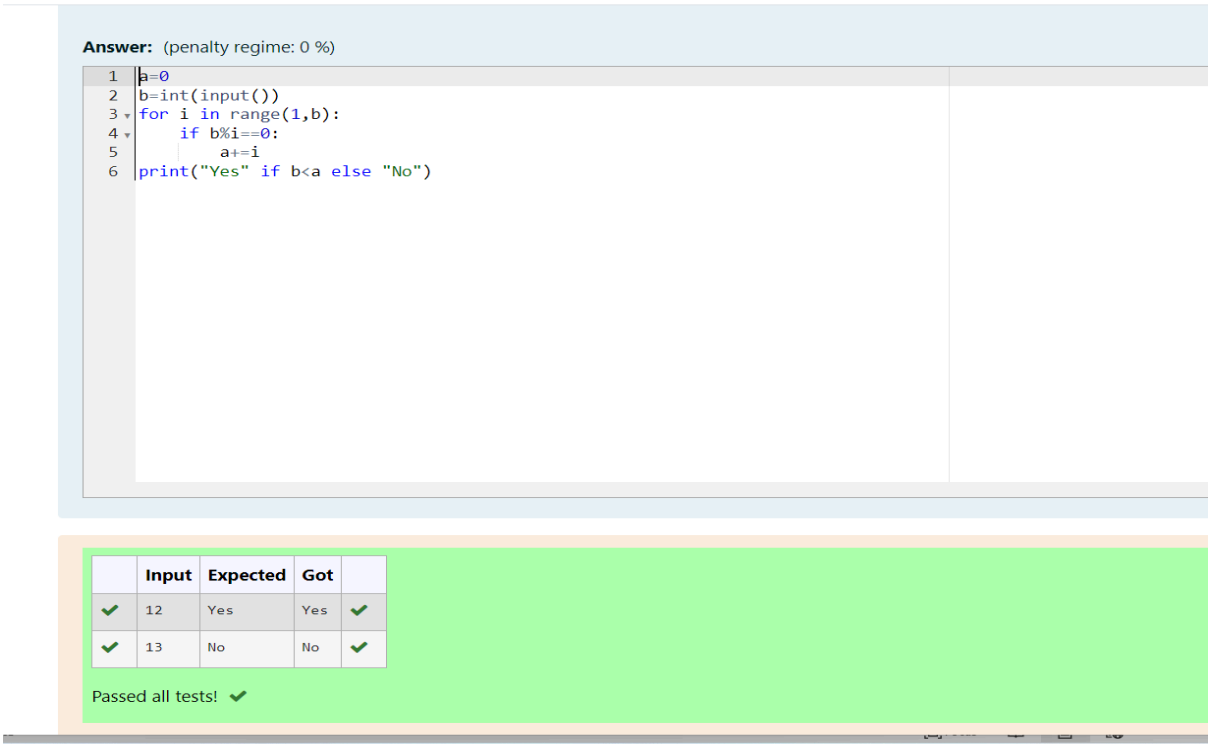
13

Output:

No

Explanation

The proper divisors of 13 is: 1, whose sum is 1. Since sum of proper divisors is not greater than the given number, 13 is not an abundant number.



**Answer:** (penalty regime: 0 %)

```
a=0
b=int(input())
for i in range(1,b):
    if b%i==0:
        a+=i
print("Yes" if b<a else "No")
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 12 | Yes | Yes | ✔ |
| ✔ | 13 | No | No | ✔ |

Passed all tests! ✔

Question **2**

Correct

Mark 1.00 out of 1.00

Flag question

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 print(len(set(list(input()))))
2
```

iz/review.php?attempt=858748&cmid=1064

| | 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 **3**

Correct

Mark 1.00 out of 1.00

Flag question

Write a program to find the count of the number of prime numbers in a specified range.

The starting and ending number of the range will be provided as input to the program.

Assumption: 2 <=starting number of the range<= ending number of the range<=7919

Example1: If the starting and ending number or the range is given as 2 and 20, the program must return 8, because there are 8 prime numbers in the specified range from 2 to 20. namely (2. 3. 5, 7, 11, 13, 17, 19)

Example2: If the starting and ending number of the range is given as 700 and 725, the program must return 3, because there are 3 prime numbers in the specified range from 700 to 725, namely (701, 709, 719)

**For example:**

| Input | Result |
|---|---|
| 2<br>20 | 8 |

| | |
|---|---|
| 700<br>725 | 3 |

**Answer:** (penalty regime: 0 %)

```
a=int(input())
b=int(input())
c=0
for num in range(a,b+1):
    if(num>1):
        for i in range(2,num):
            if(num%i==0):
                break
        else:
                c=c+1
print(c)
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 2<br>20 | 8 | 8 | ✔ |
| ✔ | 700<br>725 | 3 | 3 | ✔ |

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **4**

Correct

Mark 1.00 out of 1.00

Flag question

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 %)

```
from math import sqrt
a=int(sqrt(int(input())))
```

**Answer:** (penalty regime: 0 %)

```
from math import sqrt
a=int(sqrt(int(input())))
print((a+1)**2)
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 10 | 16 | 16 | ✔ |

Passed all tests! ✔

Correct

Question **5**

Correct

Mark 1.00 out of 1.00

Flag question

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 x 4 x 3 x 2 x 1 = 120

4! = 4 x 3 x 2 x 1 = 24

9! = 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 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 |

```
n=int(input())
fact=1
while(n>0):
    fact=fact*n
    n=n-1
print(fact)
```

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

Passed all tests! ✔

Correct

Mark 1.00 out of 1.00

Flag question

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 %)

```
a={}
for i in input():
    if i in a:a[i]+=1
    else:a[i]=1
print(sum([1 for i in a if a[i]==1]))
```

| | 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 **7**

Correct

Mark 1.00 out of 1.00

Flag question

Rakesh loves playing with numbers. He took the Fibonacci series and wants to find the sum of squares of the series until a given value. Write a code that implements his task.

Input Format:

Single Integer N

Output Format:

Display the sum of squares of the Fibonacci series until the Nth term.

Example Input:

9

Output:

1870

9

Output:

1870

Explanation:

The numbers are: 1 1 2 3 5 8 13 21 34

Sum of their squares is: 1 + 1 + 4 + 9 + 25 + 64 + 169 + 441 + 1156 = 1870

**For example:**

| Input | Result |
|---|---|
| 9 | 1870 |

**Answer:** (penalty regime: 0 %)

```
n=int(input())
f1=0
f2=1
sum=pow(f1,2)+pow(f2,2)
for i in range(1,n,1):
    fib=f1+f2
    sum=sum+pow(fib,2)
    f1=f2
    f2=fib
print(sum)
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 9 | 1870 | 1870 | ✔ |

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **8**

Correct

Mark 1.00 out of 1.00

Flag question

Write a program to find the count of the number of prime numbers in a specified range.

The starting and ending number of the range will be provided as input to the program.

Assumption: 2 <=starting number of the range<= ending number of the range<=7919

Example1: If the starting and ending number or the range is given as 2 and 20, the program must return 8, because there are 8 prime numbers in the specified range from 2 to 20. namely (2. 3. 5, 7, 11, 13, 17, 19)

Example2: If the starting and ending number of the range is given as 700 and 725, the program must return 3, because there are 3 prime numbers in the specified range from 700 to 725, namely (701, 709, 719)

**For example:**

| Input | Result |
|---|---|
| 2<br>20 | 8 |
| 700<br>725 | 3 |

```
a=int(input())
b=int(input())
c=0
for num in range(a,b+1):
    if(num>1):
      for i in range(2,num):
          if(num%i==0):
              break
      else:
          c=c+1
print(c)
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 2<br>20 | 8 | 8 | ✔ |
| ✔ | 700<br>725 | 3 | 3 | ✔ |

Passed all tests! ✔

Question 9

Correct

Mark 1.00 out of 1.00

Flag question

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 <= N <=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 %)

```
a=int(input())
flag=False
if(a==1):
    print("It's not a prime number")
elif(a>1):
    for i in range(2,a):
        if(a%i)==0:
            flag=True
            break
    if flag:
        print("1")
    else:
        print("2")
```

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **10**

Correct

Mark 1.00 out of 1.00

Flag question

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 %)

```
a=0
b=int(input())
for i in range(1,10):
    for j in range(1,b):
        if b%i==0:
            a+=1
print("Yes" if b<a else "No")
```

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

Passed all tests! ✔