Write a program to find out the prime numbers

A prime number is a number that is only divisible by 1 and itself.

```
[15]: test=int(input("Enter upper limit: "))
      for a in range(2,test+1):
          k=0
          for i in range(2,a//2+1):
              if(a%i==0):
                  k=k+1
          if(k<=0):
              print(a)
```

```
Enter upper limit: 10
2
3
5
7
```

1. User must enter the upper limit of the range and store it in a different variable test.

2. The first for loop ranges till the upper limit entered by the user.

3. The count variable is first initialized to 0.

4. The for loop ranges from 2 to the half of the number

5. The if statement then checks for the divisors of the number if the remainder is equal to 0.

6. The count variable counts the number of divisors and if the count is lesser or equal to 0, the number is a prime number.

7. If the count is greater than 0, the number is not prime

8. The final result is printed.

Q2: write a program to create the equation (a+b+c) * (a-b-c) * ab + a^2 + b ^2 + (abc)^3

```
In [40]: import numpy as np
         def Guvipython(a,b,c):
             add = a+b+c
             sub = a-b-c
             mul =a*b
             power1= a**2
             power2= b**2
             abc=  a*b*c
             cube= abc**3
             total = add*sub*mul+power1+power2+cube
             return total
         calculated = Guvipython(1,2,3)
         print (calculated)

         173
```

Sample manual computation with excel

	A	B	C	D	E	F	G	H	I	J
1										
2	Values	Arg								
3	1	a	a+b+c	6		173	(a+b+c) * (a-b-c) * ab + a^2 + b ^2 + (abc)^3			
4	2	b	a-b-c	-4						
5	3	c	ab	2						
6			Power a	1						
7			Power b	4						
8			abc	6						
9			Cube power abc	216						
10										
11										

1. Import the numpy package.
2. Define function guvipython with arguments a,b,c
3. The formula is splitted into various parts for understanding purpose as add,sub,mul,power,cube
4. Total is the final formula computation
5. Return the function as total
6. Pass the arguments for function as 1,2,3
7. Print calculated – which is the Output

Q3) urlist = ['wood','knife','axe'] , mylist = ['tree', 'apple', 'mango', 'melon'] – combine two lists

```
In [20]: urlist = ['wood','knife','axe']
         mylist = ['tree', 'apple', 'mango', 'melon']
         finallist = urlist+mylist
         print (finallist)

         ['wood', 'knife', 'axe', 'tree', 'apple', 'mango', 'melon']

In [21]: def extend_lists():
             urlist = ['wood','knife','axe']
             mylist = ['tree', 'apple', 'mango', 'melon']
             urlist.extend(mylist)
             print(urlist)
         extend_lists()

         ['wood', 'knife', 'axe', 'tree', 'apple', 'mango', 'melon']

In [ ]:
```

1) List 1 and list 2 are defined
2) Two ways we can merge the list. Either by first method urlist +Mylist
3) Second option is we can extend the list by extend function
4) Ouput is both the list are merged together

Q4) write a program for natural number based on user input

```
0]: def checkNatNum(n):
        if  int(n) > 0:
            return True
        else:
            return False
    checkNatNum(1)

0]: True
```

```
1]: def checkNatNum(n):
        if  int(n) > 0:
            return True
        else:
            return False
    checkNatNum(0)

1]: False
```

```
]:
```

1) Natural numbers include all the whole numbers excluding the number 0
2) Function checkNatNum is created and if check is done to verify whether the input number is greater than 0
3) If is greater than 0 then true else false is displayed
4) In **checkNatNum** is have passed 1 and 0 in the screenshot to show the expected results

Q5) write class and function for the equation sqrt(x1-x2) ^ 2 + sqrt(y1 – y2) ^2 using try except handling

Try :

```
In [25]: import math
class Calulation:
    def __init__(self, X1, X2,Y1,Y2):
        self.X1=X1
        self.X2=X2
        self.Y1=Y1
        self.Y2=Y2
        try:
            c1=math.sqrt(X1 - X2)**2
            c2=math.sqrt(Y1 - Y2)**2
            c3 =c1+c2
            print(c3)
        except:
             print("Check ur input- X1 should be greater than X2 and also Y1 Should be greater than Y2")
s=Calulation(4,2,6,5)
```

```
3.0000000000000004
```

Except part:

```
In [28]: import math
class Calulation:
    def __init__(self, X1, X2,Y1,Y2):
        self.X1=X1
        self.X2=X2
        self.Y1=Y1
        self.Y2=Y2
        try:
            c1=math.sqrt(X1 - X2)**2
            c2=math.sqrt(Y1 - Y2)**2
            c3 =c1+c2
            print(c3)
        except:
             print("Check ur input- X1 should be greater than X2 and also Y1 Should be greater than Y2")
s=Calulation(1,2,4,5)
```

```
Check ur input- X1 should be greater than X2 and also Y1 Should be greater than Y2
```

1) I have created a class named calculation
2) Constructors are used to initializing the object's state. The task of constructors is to initialize(assign values) to the data members of the class when an object of the class is created.
3) X,X2,Y1,Y2 are created and initialized
4) Inside try, I am calculating the equation **sqrt(x1-x2) ^ 2 + sqrt(y1 – y2) ^2**

5) When invalid entries are passed as input for the method, I am using except to throw the exception

Q6) Name = "Guvi python" - write a program to get "python" word from the string

```
string = "Guvi python"
words = string.split(' ')[1]
print(words)
```

```
python
```

```
|
```

```
: string = "Guvi python"
  words = string.split(' ')[0]
  print(words)
```

```
Guvi
```

1) guvi python is set to variable
2) using split function I am using space as delimiter
3) Index [1] is set to find the second value - PYTHON
4) Index [0] is set to find the first value - GUVI
5) Printing the words

Q7) Using class and function - Write a program for palindrome Ex. Madam

```
def isPalindrome(s):
    return s == s[::-1]
s = "madam"
ans = isPalindrome(s)

if ans:
    print("Yes, its a palindrome")
else:
    print("Not a Palindrome")
```

```
Yes, its a palindrome
```

```
def isPalindrome(s):
    return s == s[::-1]
s = "test"
ans = isPalindrome(s)

if ans:
    print("Yes, its a palindrome")
else:
    print("Not a Palindrome")
```

```
Not a Palindrome
```

1) This program is to find the input string is palindrome or not.
2) Find reverse of string
3) s == s[::-1] means reversing a string, list, or any iterable with an ordering.
4) Store op in ans
5) Perfrom a if condition to check it's a palindrome or not.

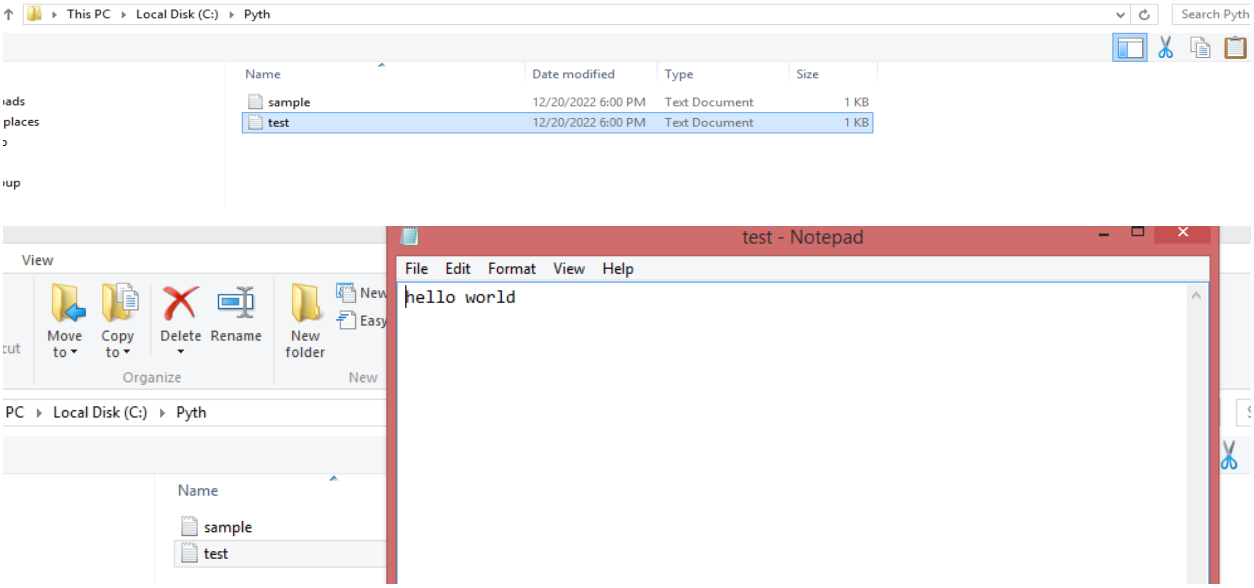
Q8) using file handling – write a text file in ur system with “hello world”

```
31]: from pathlib import Path

dir_path = Path('C:/Pyth')
file_name = "test.txt"
file_path = dir_path.joinpath(file_name)

# check that directory exists
if dir_path.is_dir():
    with open(file_path, "w") as f:
        f.write("hello world")
        print('File is created.')
```

```
File is created.
```



1) The pathlib module is object-oriented way to handle file system paths.
1) On Windows, paths are written using backslashes (\) as the separator between folder names. On Unix based operating system such as macOS, Linux, and BSDs, the forward slash (/) is used as the path separator. Which this pathlib path we can handle these scenarios .
2) Dirpath is the destination path where the file has to be created
3) file_name is the name of file to be created
4) file_path – we use Joinpath – for combining the paths
5) if the path exists then write the with open and write the text (file_path, "w") as f:
6) f.write("hello world") is added to text file and final notification of file creation is notified as print
7) if directory not available print ("Directory doesn\'t exist.")

Q9) create option button using tkinter GUI in python

```
In [*]: import tkinter as tk

root = tk.Tk()

v = tk.IntVar()

tk.Label(root,
        text="""Choose a
programming language:""",
        justify = tk.LEFT,
        padx = 20).pack()

tk.Radiobutton(root,
               text="Python",
               padx = 20,
               variable=v,
               value=1).pack(anchor=tk.W)

tk.Radiobutton(root,
               text="Perl",
               padx = 20,
               variable=v,
               value=2).pack(anchor=tk.W)
tk.Radiobutton(root,
               text="Java",
               padx = 20,
               variable=v,
               value=2).pack(anchor=tk.W)
root.mainloop()
```

1) The code starts by creating a Tkinter window with the root object.
2) The window is then given an ID of "root".
3) Next, two variables are created: v and tk.Label(root, text="Choose a programming language:").
4) The first variable is v and it will be used to store the value chosen by the user when they click on one of the radiobuttons in the program.
5) The second variable is called tk.Label(root, text="Python", justify = tk.LEFT) and it will be used to display what programming language was selected when Python or Perl were clicked on as options for this program's choice of programming language.

6) Next, two radiobuttons are created: one labeled "Python" which stores 1 as its value and another labeled "Perl" which stores 2 as its value (the values stored in these variables can change depending on how many times each button has been clicked).
7) These buttons have their own labels that say what they do ("Choose a Programming Language") so that users know what each button does before clicking them (this label also helps make sure that users don't accidentally choose something else).
8) Then these buttons are added to the main window using pack()
9) The code creates a Tkinter window with two radiobuttons.
10) The first radiobutton is labeled "Python" and the second is labeled "Perl".
11) When the user clicks on either of these buttons, they are toggled between 1 and 2 respectively.

Q10) Keep only numbers from the following string x = " 89e9jcd^o38829@3%3,/mkl$w1"

```
]: inp_str = "89e9jcd^o38829@3%3,/mkl$w1"
   num = ""
   for c in inp_str:
       if c.isdigit():
           num = num + c
   print("Extracted numbers from the list : " + num)

   Extracted numbers from the list : 89938829331

]:
```

1) Using for loop to extract the numbers from string .
2) Inp string is defined is value as per question.
3) Num is defined with empty string
4) Using loop for the desired input string , c value will start iterating over each and every values in the inp_str and check whether it is digit using is digit function.
5) Once the check is done and after end of all iteration the values are stored in num and printed the extracted numbers .