

# Python -Control Flow

September 19, 2023

```
[ ]: #Write a Python program to check if a given number is positive or negative.  
num = float(input("Enter a number: "))  
if num > 0:  
    print("Positive number")  
else:  
    print("Negative number")
```

```
[4]: #Create a program that determines if a person is eligible to vote based on  
    ↳ their age.  
age = int(input("Enter age : "))  
if age >= 18: print("Eligible for Voting!")  
else: print("Not Eligible for Voting!")
```

Enter age : 67

Eligible for Voting!

```
[5]: #Develop a program to find the maximum of two numbers using if-else statements.  
    # Python program to return maximum of two numbers  
  
    # Getting input from user  
num1 = int(input("Enter the first number: "))  
num2 = int(input("Enter the second number: "))  
  
    # printing the maximum value  
if(num1 > num2):  
    print(num1, "is greater")  
elif(num1 < num2):  
    print(num2, "is greater")  
else:  
    print("Both are equal")
```

Enter the first number: 56

Enter the second number: 51

56 is greater

```
[7]: #Write a Python script to classify a given year as a leap year or not.
year=int(input("Enter year to be checked:"))
if(year%4==0):
    print("The year is a leap year!")
else:
    print("The year isn't a leap year!")
```

Enter year to be checked: 2001

The year isn't a leap year!

```
[9]: #Create a program that checks whether a character is a vowel or a consonant.
# Get an input character from the user
character = input("Enter a character: ")

# Creating a list of vowels
vowels = ['a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U']

# Check if the character is a vowel or not
if character in vowels:
    print(f"The character '{character}' is a vowel!")
else:
    print(f"The character '{character}' is a consonant!")
```

Enter a character: o

The character 'o' is a vowel!

```
[13]: #Implement a program to determine whether a given number is even or odd.
num = int (input ("Enter any number to test whether it is odd or even: "))

if (num % 2) == 0:

    print ("The number is even")

else:

    print ("The provided number is odd")
```

Enter any number to test whether it is odd or even: 56

The number is even

```
[14]: #Write a Python function to calculate the absolute value of a number without_
↳using the `abs()`` function.
def my_abs(value):
    """Returns absolute value without using abs function"""
    if value <= 0:
        return value * -1
```

```
    return value * 1

print(my_abs(-3.5))
```

3.5

[15]: *#Develop a program that determines the largest of three given numbers using if-else statements.*

```
a = 5
b = 4
c = 7

largest = 0

if a > b and a > c:
    largest = a
if b > a and b > c:
    largest = b
if c > a and c > b:
    largest = c

print(largest, "is the largest of three numbers.")
```

7 is the largest of three numbers.

[20]: *#Create a program that checks if a given string is a palindrome.*

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

# Driver code
s = "malayalam"
ans = isPalindrome(s)

if ans:
    print("Yes")
else:
    print("No")
```

Yes

[23]: *#Write a Python program to calculate the grade based on a student's score.*

```
total1 = 44
total2 = 67
total3 = 76
total4 = 99
total5 = 58
```

```

tot = total1 + total2 + total3 + total4 + total4
avg = tot / 5

if avg >= 91 and avg <= 100:
    print("Your Grade is A1")
elif avg >= 81 and avg < 91:
    print("Your Grade is A2")
elif avg >= 71 and avg < 81:
    print("Your Grade is B1")
elif avg >= 61 and avg < 71:
    print("Your Grade is B2")
elif avg >= 51 and avg < 61:
    print("Your Grade is C1")
elif avg >= 41 and avg < 51:
    print("Your Grade is C2")
elif avg >= 33 and avg < 41:
    print("Your Grade is D")
elif avg >= 21 and avg < 33:
    print("Your Grade is E1")
elif avg >= 0 and avg < 21:
    print("Your Grade is E2")
else:
    print("Invalid Input!")

```

Your Grade is B1

[26]: *#Write a program to find the largest among three numbers using nested if-else statements.*

```

# input three integer numbers
a=int(input("Enter A: "))
b=int(input("Enter B: "))
c=int(input("Enter C: "))

# conditions to find largest
if a>b:
    if a>c:
        g=a
    else:
        g=c
else:
    if b>c:
        g=b
    else:
        g=c

# print the largest number

```

```
print("Greatest = ",g)
```

Enter A: 56

Enter B: 50

Enter C: 1

Greatest = 56

[27]: *#Implement a program to determine if a triangle is equilateral, isosceles, or scalene.*

```
print("Input lengths of the triangle sides: ")
x = int(input("x: "))
y = int(input("y: "))
z = int(input("z: "))

if x == y == z:
    print("Equilateral triangle")
elif x==y or y==z or z==x:
    print("isosceles triangle")
else:
    print("Scalene triangle")
```

Input lengths of the triangle sides:

x: 10

y: 20

z: 30

Scalene triangle

[28]: *#Develop a program that checks if a year is a leap year and also if it is a century year.*

*# Python program to check if year is a leap year or not*

```
year = 2000
```

*# To get year (integer input) from the user*

```
# year = int(input("Enter a year: "))
```

*# divided by 100 means century year (ending with 00)*

*# century year divided by 400 is leap year*

```
if (year % 400 == 0) and (year % 100 == 0):
    print("{0} is a leap year".format(year))
```

*# not divided by 100 means not a century year*

*# year divided by 4 is a leap year*

```
elif (year % 4 == 0) and (year % 100 != 0):
    print("{0} is a leap year".format(year))
```

```

# if not divided by both 400 (century year) and 4 (not century year)
# year is not leap year
else:
    print("{0} is not a leap year".format(year))

```

2000 is a leap year

[29]: #Write a Python script to determine if a number is positive, negative, or zero.

```

num = float(input("Enter a number: "))
if num > 0:
    print("Positive number")
elif num == 0:
    print("Zero")
else:
    print("Negative number")

```

Enter a number: -45

Negative number

[31]: #Create a program to check if a person is a teenager (between 13 and 19 years old).

```

age = int(input("Enter your age"))
if age < 13: # condition - age < 13
    print("Hey! kid")
elif age > 13 and age <= 19: # condition - age > 13 and age < 20
    print("So, you are enjoying your teenage.")
else:
    print("You are grown up.")

```

Enter your age 19

So, you are enjoying your teenage.

[32]: #Develop a program that determines the type of angle based on its measure (acute, obtuse, or right).

```

def triangleType(a, b, c):
    sa = pow(a, 2)
    sb = pow(b, 2)
    sc = pow(c, 2)
    if (sa == sc + sb or sb == sa+sc or sc == sa+sb):
        print("Right-angled triangle")
    elif (sa > sc + sb or sb > sa+sc or sc > sa+sb):
        print("Obtuse-angled triangle")
    else:
        print("Acute-angled triangle")

a = int(input("Enter side a: "))

```

```

b = int(input("Enter side b: "))
c = int(input("Enter side c: "))
triangleType(a, b, c)

```

Enter side a: 3

Enter side b: 4

Enter side c: 5

Right-angled triangle

[33]: *#Write a Python program to calculate the roots of a quadratic equation.*

```

import cmath
a = float(input('Enter a: '))
b = float(input('Enter b: '))
c = float(input('Enter c: '))

# calculate the discriminant
d = (b**2) - (4*a*c)

# find two solutions
sol1 = (-b-cmath.sqrt(d))/(2*a)
sol2 = (-b+cmath.sqrt(d))/(2*a)
print('The solution are {0} and {1}'.format(sol1,sol2))

```

Enter a: 12

Enter b: 13

Enter c: 10

The solution are (-0.5416666666666666-0.7347996703561832j) and  
 (-0.5416666666666666+0.7347996703561832j)

[34]: *#18. Implement a program to determine the day of the week based on a  
 ↪user-provided number (1 for Monday, 2  
 #for Tuesday, etc.).*

```

import datetime
import calendar

def findDay(date):
    born = datetime.datetime.strptime(date, '%d %m %Y').weekday()
    return (calendar.day_name[born])

# Driver program
date = '03 02 2019'
print(findDay(date))

```

Sunday

[35]: #18. Implement a program to determine the day of the week based on a user-provided number (1 for Monday, 2 for Tuesday, etc.).

```
weekday = int(input("Enter weekday number (1-7) : "))

if weekday == 1 :
    print("\nMonday");

elif weekday == 2 :
    print("\nTuesday")

elif(weekday == 3) :
    print("\nWednesday")

elif(weekday == 4) :
    print("\nThursday")

elif(weekday == 5) :
    print("\nFriday")

elif(weekday == 6) :
    print("\nSaturday")

elif (weekday == 7) :
    print("\nSunday")

else :
    print("\nPlease enter any weekday number (1-7)")
```

Enter weekday number (1-7) : 2

Tuesday

[46]: #Create a program that determines if a year is a leap year and also if it is evenly divisible by 400.

```
year = int(input("Enter year to be checked: "))

if year % 4 == 0:

    if year % 400 == 0:
        print("The year is a leap year!")
    else:
        print("The year is not a leap year!")

else:
    print("The year is not a leap year!")
```



Enter year to be checked: 2000

The year is a leap year!

[47]: *#Develop a program that checks if a given number is prime or not using nested  
↪if-else statements.*

```
num = 11
# If given number is greater than 1
if num > 1:
    # Iterate from 2 to n / 2
    for i in range(2, int(num/2)+1):
        # If num is divisible by any number between
        # 2 and n / 2, it is not prime
        if (num % i) == 0:
            print(num, "is not a prime number")
            break
    else:
        print(num, "is a prime number")
else:
    print(num, "is not a prime number")
```

11 is a prime number

[59]: *#Write a Python program to assign grades based on different ranges of scores  
↪using elif statements.*

```
m=int(input("Enter your mark "))
if(m>=80):
    print("You got A grade")
elif(m>=60):
    print("You got B grade")
elif(m>=40):
    print("You got C grade")
else:
    print("Failed in this Exam ")
```

Enter your mark 81

You got A grade

[63]: *#Develop a program to categorize a given person's BMI into underweight, normal,  
↪overweight, or obese using*

*#elif statements.*

```
def BMI(height, weight):
    bmi = weight/(height**2)
    return bmi
```

*# Driver code*

```
height = 1.79832
```

```

weight = 70

# calling the BMI function
bmi = BMI(height, weight)
print("The BMI is", format(bmi), "so ", end='')

# Conditions to find out BMI category
if (bmi < 18.5):
    print("underweight")

elif ( bmi >= 18.5 and bmi < 24.9):
    print("Healthy")

elif ( bmi >= 24.9 and bmi < 30):
    print("overweight")

elif ( bmi >=30):
    print("Suffering from Obesity")

```

The BMI is 21.64532402096181 so Healthy

[64]: *#Create a program that determines whether a given number is positive, negative, or zero using elif statements.*

```

num = float(input("Enter a number: "))
if num > 0:
    print("Positive number")
elif num == 0:
    print("Zero")
else:
    print("Negative number")

```

Enter a number: 12

Positive number

[65]: *#Write a Python script to determine the type of a character (uppercase, lowercase, or special) using elif statements.*

```

ch = input("Please Enter Your Own Character : ")

if(ch.isupper()):
    print("The Given Character ", ch, "is an Uppercase Alphabet")
elif(ch.islower()):
    print("The Given Character ", ch, "is a Lowercase Alphabet")
else:
    print("The Given Character ", ch, "is Not a Lower or Uppercase Alphabet")

```

Please Enter Your Own Character : C

The Given Character C is an Uppercase Alphabet

```
[67]: #26. Implement a program to calculate the discounted price based on different
      ↪purchase amounts using elif
      #statements.
      p=int(input('Enter purchase amount '))

      if p<=5000:
          dis=0
      elif p>5000 and p<=10000:
          dis=(p-5000)*(10/100.0)
      elif p>10000 and p<=20000:
          dis=(p-10000)*(20/100.0)
      elif p>20000 and p<=30000:
          dis=(p-20000)*(30/100.0)
      else:
          dis=(p-30000)*(40/100.0)

      print('Discount Amount = {:.2f}'.format(dis))
      print('Net Bill Amount = {:.2f}'.format(p-dis))
```

Enter purchase amount 6000

Discount Amount = 100.00

Net Bill Amount = 5900.00

```
[68]: #Develop a program to calculate the electricity bill based on different
      ↪consumption slabs using elif
      #statements.
      units = int(input(" Please enter Number of Units you Consumed : "))

      if(units < 50):
          amount = units * 2.60
          surcharge = 25
      elif(units <= 100):
          amount = 130 + ((units - 50) * 3.25)
          surcharge = 35
      elif(units <= 200):
          amount = 130 + 162.50 + ((units - 100) * 5.26)
          surcharge = 45
      else:
          amount = 130 + 162.50 + 526 + ((units - 200) * 8.45)
          surcharge = 75

      total = amount + surcharge
      print("\nElectricity Bill = %.2f" %total)
```

Please enter Number of Units you Consumed : 3000

Electricity Bill = 24553.50

```
[75]: #Write a Python script to determine the season based on a user-provided month,
      ↪ using elif statements.
month = input("Input the month (e.g. January, February etc.): ")

if month in ('January', 'February', 'March'):
    season = 'winter'
elif month in ('April', 'May', 'June'):
    season = 'spring'
elif month in ('July', 'August', 'September'):
    season = 'summer'
else:
    season = 'autumn'

print("Season is",season)
```

Input the month (e.g. January, February etc.): January

Season is winter

```
[77]: #Implement a program to determine the type of a year (leap or common) and month,
      ↪ (30 or 31 days) using
      #elif statements.

year=int(input("Enter year to be checked:"))
if(year%4==0 and year%100!=0 or year%400==0):
    print("The year is a leap year!")
else:
    print("The is a common year")

month = 12
year=2023

if((month==2) and ((year%4==0) or ((year%100==0) and (year%400==0)))) :
    print("Number of days is 29");

elif(month==2) :
    print("Number of days is 28");

elif(month==1 or month==3 or month==5 or month==7 or month==8 or month==10 or
      ↪ month==12) :
    print("Number of days is 31");
```

```
else :  
    print("Number of days is 30");
```

Enter year to be checked: 2000

The year is a leap year!

Number of days is 31

[1]: *#Write a Python program that checks if a given number is positive, negative, or zero.*

```
def NumberCheck(a):  
    # Checking if the number is positive  
    if a > 0:  
        print("Number given by you is Positive")  
    # Checking if the number is negative  
    elif a < 0:  
        print("Number given by you is Negative")  
    # Else the number is zero  
    else:  
        print("Number given by you is zero")  
# Taking number from user  
a = float(input("Enter a number as input value: "))  
# Printing result  
NumberCheck(a)
```

Enter a number as input value: 56

Number given by you is Positive

[2]: *#Create a program to determine if a person is eligible to vote based on their age.*

```
age = int(input("Enter age : "))  
  
if age >= 18:  
    print("Eligible for Voting!")  
else:  
    print("Not Eligible for Voting!")
```

Enter age : 17

Not Eligible for Voting!

[3]: *#Write a program to find the maximum of two given numbers using conditional statements.*

```
# Getting input from user  
num1 = int(input("Enter the first number: "))  
num2 = int(input("Enter the second number: "))
```

```
# printing the maximum value
if(num1 > num2):
    print(num1, "is greater")
elif(num1 < num2):
    print(num2, "is greater")
else:
    print("Both are equal")
```

Enter the first number: 12  
Enter the second number: 11

12 is greater

[4]: #Develop a program that calculates the grade of a student based on their exam score.

```
print("Enter Marks Obtained in 5 Subjects: ")
total1 = 44
total2 = 67
total3 = 76
total4 = 99
total5 = 58

tot = total1 + total2 + total3 + total4 + total4
avg = tot / 5

if avg >= 91 and avg <= 100:
    print("Your Grade is A1")
elif avg >= 81 and avg < 91:
    print("Your Grade is A2")
elif avg >= 71 and avg < 81:
    print("Your Grade is B1")
elif avg >= 61 and avg < 71:
    print("Your Grade is B2")
elif avg >= 51 and avg < 61:
    print("Your Grade is C1")
elif avg >= 41 and avg < 51:
    print("Your Grade is C2")
elif avg >= 33 and avg < 41:
    print("Your Grade is D")
elif avg >= 21 and avg < 33:
    print("Your Grade is E1")
elif avg >= 0 and avg < 21:
    print("Your Grade is E2")
else:
    print("Invalid Input!")
```

Enter Marks Obtained in 5 Subjects:  
Your Grade is B1

```
[5]: #Create a program that checks if a year is a leap year or not.
year=int(input("Enter year to be checked:"))
if(year%4==0 and year%100!=0 or year%400==0):
    print("The year is a leap year!")
else:
    print("The year isn't a leap year!")
```

Enter year to be checked: 2001

The year isn't a leap year!

```
[6]: #Write a program to classify a triangle based on its sides' lengths.
# Function definition to check validity
def is_valid_triangle(a,b,c):
    if a+b>=c and b+c>=a and c+a>=b:
        return True
    else:
        return False

# Function definition for type
def type_of_triangle(a,b,c):
    if a==b and b==c:
        print('Triangle is Equilateral.')
    elif a==b or b==c or a==c:
        print('Triangle is Isosceles.')
    else:
        print('Triangle is Scalane')

# Reading Three Sides
side_a = float(input('Enter length of side a: '))
side_b = float(input('Enter length of side b: '))
side_c = float(input('Enter length of side c: '))

# Function call & making decision
if is_valid_triangle(side_a, side_b, side_c):
    type_of_triangle(side_a, side_b, side_c)
else:
    print('Tringle is not possible from given sides.')
```

Enter length of side a: 10

Enter length of side b: 11

Enter length of side c: 12

Triangle is Scalane

```
[7]: #Build a program that determines the largest of three given numbers.
# Python program to find the largest number among the three input numbers
# take three numbers from user
```

```

num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

if (num1 > num2) and (num1 > num3):
    largest = num1
elif (num2 > num1) and (num2 > num3):
    largest = num2
else:
    largest = num3

print("The largest number is",largest)

```

Enter first number: 1  
Enter second number: 2  
Enter third number: 3  
The largest number is 3.0

```

[8]: #Develop a program that checks whether a character is a vowel or a consonant.
# Get an input character from the user
character = input("Enter a character: ")

# Creating a list of vowels
vowels = ['a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U']

# Check if the character is a vowel or not
if character in vowels:
    print(f"The character '{character}' is a vowel!")
else:
    print(f"The character '{character}' is a consonant!")

```

Enter a character: r  
The character 'r' is a consonant!

```

[10]: #Create a program to calculate the total cost of a shopping cart based on
↪discounts.
def discountPercentage(S, M):

    # Calculating discount
    discount = M - S

    # Calculating discount percentage
    disPercent = (discount /M) * 100

    return disPercent

```



```
# Driver code
if __name__=='__main__':
    M = 120
    S = 100

    print(discountPercentage(S, M), "%")

    M = 1000
    S = 500

    print(discountPercentage(S, M), "%")
```

16.666666666666664 %  
50.0 %

```
[14]: #Write a program that checks if a given number is even or odd.
num = int (input ("Enter any number to test whether it is odd or even: "))

if (num % 2) == 0:

    print ("The number is even")

else:

    print ("The provided number is odd")
```

Enter any number to test whether it is odd or even: 91  
The provided number is odd

```
[16]: #Write a program that calculates the roots of a quadratic equation .
from math import sqrt

print("Quadratic function : (a * x^2) + b*x + c")
a = float(input("a: "))
b = float(input("b: "))
c = float(input("c: "))

r = b**2 - 4*a*c

if r > 0:
    num_roots = 2
    x1 = (((-b) + sqrt(r))/(2*a))
    x2 = (((-b) - sqrt(r))/(2*a))
    print("There are 2 roots: %f and %f" % (x1, x2))
elif r == 0:
```

```

num_roots = 1
x = (-b) / 2*a
print("There is one root: ", x)
else:
    num_roots = 0
    print("No roots, discriminant < 0.")
    exit()

```

Quadratic function :  $(a * x^2) + b*x + c$

a: 2

b: 3

c: 4

No roots, discriminant < 0.

```

[1]: #Create a program that determines the day of the week based on the day number
      ↪ (1-7).
      # Taken day number from user
      weekday = int(input("Enter weekday day number (1-7) : "))

      if weekday == 1 :
          print("\nMonday");

      elif weekday == 2 :
          print("\nTuesday")

      elif(weekday == 3) :
          print("\nWednesday")

      elif(weekday == 4) :
          print("\nThursday")

      elif(weekday == 5) :
          print("\nFriday")

      elif(weekday == 6) :
          print("\nSaturday")

      elif (weekday == 7) :
          print("\nSunday")

      else :
          print("\nPlease enter weekday number between 1-7.")

```

Enter weekday day number (1-7) : 3

Wednesday

[2]: *#Develop a program that calculates the factorial of a given number using ↵  
↵recursion.*

```
def recur_factorial(n):  
    if n == 1:  
        return n  
    else:  
        return n*recur_factorial(n-1)  
# take input from the user  
num = int(input("Enter a number: "))  
# check is the number is negative  
if num < 0:  
    print("Sorry, factorial does not exist for negative numbers")  
elif num == 0:  
    print("The factorial of 0 is 1")  
else:  
    print("The factorial of",num,"is",recur_factorial(num))
```

Enter a number: 4

The factorial of 4 is 24

[3]: *#Write a program to find the largest among three numbers without using the ↵  
↵`max()` function.*

```
num1 = float(input("Enter first number: "))  
num2 = float(input("Enter second number: "))  
num3 = float(input("Enter third number: "))  
  
if (num1 > num2) and (num1 > num3):  
    largest = num1  
elif (num2 > num1) and (num2 > num3):  
    largest = num2  
else:  
    largest = num3  
  
print("The largest number is",largest)
```

Enter first number: 14

Enter second number: 12

Enter third number: 11

The largest number is 14.0

[4]: *#Build a program that checks if a given string is a palindrome or not.*

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

*# Driver code*

```
s = "malayalam"
ans = isPalindrome(s)

if ans:
    print("Yes")
else:
    print("No")
```

Yes

[7]: *#Write a program that calculates the average of a list of numbers, excluding the smallest and largest values.*

```
def Average(lst):
    return sum(lst) / len(lst)

list=[1,4,5,6,]
print('my list: ',list)
print('minimum of list: ',min(list))
print('maximum of list: ',max(list))
list.remove(min(list))
print('after removal of minimum from list: ',list)
list.remove(max(list))
print('after removal of maximum from list: ',list)

average = Average(list)

# Printing average of the list
print("Average of the list =", average)
```

```
my list:  [1, 4, 5, 6]
minimum of list:  1
maximum of list:  6
after removal of minimum from list:  [4, 5, 6]
after removal of maximum from list:  [4, 5]
Average of the list = 4.5
```

[8]: *#Develop a program that converts a given temperature from Celsius to Fahrenheit.*

```
celsius = int(input("Enter the Temperature in Celsius :\n"))
fahrenheit = (1.8 * celsius) + 32
print("Temperature in Fahrenheit :", fahrenheit)
```

```
Enter the Temperature in Celsius :
12
```

```
Temperature in Fahrenheit : 53.6
```

[9]: *#Create a program that simulates a basic calculator for addition, subtraction, multiplication, and division.*

```
n1 = float(input("Enter the First Number: "))
n2 = float(input("Enter the Second Number: "))
```

*#addition*

```
print("{} + {} = ".format(n1, n2))
print(n1 + n2)
```

*#subtraction*

```
print("{} - {} = ".format(n1, n2))
print(n1 - n2)
```

*#multiplication*

```
print("{} * {} = ".format(n1, n2))
print(n1 * n2)
```

*#division*

```
print("{} / {} = ".format(n1, n2))
print(n1 / n2)
```

Enter the First Number: 13  
Enter the Second Number: 23

13.0 + 23.0 =  
36.0  
13.0 - 23.0 =  
-10.0  
13.0 \* 23.0 =  
299.0  
13.0 / 23.0 =  
0.5652173913043478

[10]: *#Create a program that calculates the income tax based on the user's income and tax brackets.*

```
def calculate(amount, percent):
    return (amount * percent) / 100

def calculate_income_tax(total_income:
                          float) -> float:

    if total_income <= 250000:
        return 0
```



```

# Getting character set for password
while(True):
    choice = int(input("Pick a number "))
    if(choice == 1):

        # Adding letters to possible characters
        characterList += string.ascii_letters
    elif(choice == 2):

        # Adding digits to possible characters
        characterList += string.digits
    elif(choice == 3):

        # Adding special characters to possible
        # characters
        characterList += string.punctuation
    elif(choice == 4):
        break
    else:
        print("Please pick a valid option!")

password = []

for i in range(length):

    # Picking a random character from our
    # character list
    randomchar = random.choice(characterList)

    # appending a random character to password
    password.append(randomchar)

# printing password as a string
print("The random password is " + "".join(password))

```

Enter password length: 10

Choose character set for password from these :

1. Digits
2. Letters
3. Special characters
4. Exit

Pick a number 3

Pick a number 4

The random password is ##`%)(-\$\_'

```

[2]: #Write a program that simulates a basic quiz game with multiple-choice
      ↪ questions and scoring.
print("Wellcome to quiz game !!")
print('NOTE: if your spelling is incorrect then it is considered as wrong
      ↪ answer')
score = 0
question_no = 0
playing = input('Do you want to play ? ').lower()
if playing == 'yes':
    question_no += 1
    ques = input(f'\n{question_no}. what does CPU stand for? ').lower()
    if ques == 'central processing unit':
        score +=1
        print('correct! you got 1 point')

    else:
        print('Incorrect!')
        print(f'current answer is --> central processing unit')

# -----1
    question_no += 1
    ques = input(f'\n{question_no}. what does GPU stand for? ').lower()

    if ques == 'graphics processing unit':
        score +=1
        print('correct! you got 1 point')

    else:
        print('Incorrect!')
        print(f'current answer is --> graphics processing unit')

# -----2
    question_no += 1
    ques = input(f'\n{question_no}. what does RAM stand for? ').lower()

    if ques == 'random access memory':
        score +=1
        print('correct! you got 1 point')

    else:
        print('Incorrect!')
        print(f'current answer is --> random access memory')

# -----3
    question_no += 1
    ques = input(f'\n{question_no}. what does PSU stand for? ').lower()

```



```

if ques == 'power supply unit':
    score +=1
    print('correct! you got 1 point')

else:
    print('Incorrect!')
    print(f'current answer is --> power supply unit')

# -----4
question_no += 1
ques = input(f'\n{question_no}. what does ROM stand for? ').lower()

if ques == 'read only memory':
    score +=1
    print('correct! you got 1 point')

else:
    print('Incorrect!')
    print(f'current answer is --> read only memory')

# -----5

else:
    print('thankyou you are out of a game.')
    quit()

print(f'\nnumber of question is {question_no}')
print(f'your score is {score}')
try:
    percentage = (score *100)/question_no
except ZeroDivisionError:
    print('0% quetions are correct')

print(f'{percentage}% questions are correct.')

```

Wellcome to quiz game !!

NOTE: if your spelling is incorrect then it is considered as wrong answer

Do you want to play ? yes

1. what does CPU stand for? central processing unit

correct! you got 1 point

2. what does GPU stand for?

Incorrect!

current answer is --> graphics processing unit

3. what does RAM stand for?

Incorrect!

current answer is --> random access memory

4. what does PSU stand for?

Incorrect!

current answer is --> power supply unit

5. what does ROM stand for?

Incorrect!

current answer is --> read only memory

number of question is 5

your score is 1

20.0% questions are correct.

[2]: *#Develop a program that determines whether a given year is a prime number or not.*

```
year = 1969
# If given number is greater than 1
if year > 1:
    # Iterate from 2 to n / 2
    for i in range(2, int(year/2)+1):
        # If num is divisible by any number between
        # 2 and n / 2, it is not prime
        if (year%i) == 0:
            print(year, "is not a prime number")
            break
    else:
        print(year, "is a prime number")
else:
    print(year, "is not a prime number")
```

1969 is not a prime number

[3]: *#Create a program that sorts three numbers in ascending order using conditional statements.*

```
a, b, c = 3, 1, 8

x = min(a, b, c) # Smallest of the three
z = max(a, b, c) # Largest of the three
```

```
y = (a + b + c) - (x + z) # Since you have two of the three, you can solve for
                           # the third
```

```
print(a, b, c)
print(x, y, z)
```

```
3 1 8
```

```
1 3 8
```

[4]: #Build a program that determines the roots of a quartic equation using  
 ↪ numerical methods.

```
import math

# Prints roots of quadratic equation
#  $ax^2 + bx + c$ 
def findRoots(a, b, c):

    # If a is 0, then equation is
    # not quadratic, but linear
    if a == 0:
        print("Invalid")
        return -1
    d = b * b - 4 * a * c
    sqrt_val = math.sqrt(abs(d))

    if d > 0:
        print("Roots are real and different ")
        print((-b + sqrt_val)/(2 * a))
        print((-b - sqrt_val)/(2 * a))
    elif d == 0:
        print("Roots are real and same")
        print(-b / (2*a))
    else: # d<0
        print("Roots are complex")
        print(- b / (2*a), " + i", sqrt_val / (2 * a))
        print(- b / (2*a), " - i", sqrt_val / (2 * a))

# Driver Program
if __name__ == '__main__':
    a = 1
    b = -7
    c = 12

    # Function call
    findRoots(a, b, c)
```

Roots are real and different  
4.0  
3.0

```
[5]: #Write a program that calculates the BMI (Body Mass Index) and provides health
      ↳ recommendations based
      on the user's input.
      # asking for input from the users
the_height = float(input("Enter the height in cm: "))
the_weight = float(input("Enter the weight in kg: "))
# defining a function for BMI
the_BMI = the_weight / (the_height/100)**2
# printing the BMI
print("Your Body Mass Index is", the_BMI)
# using the if-elif-else conditions
if the_BMI <= 18.5:
    print("Oops! You are underweight.")
elif the_BMI <= 24.9:
    print("Awesome! You are healthy.")
elif the_BMI <= 29.9:
    the_print("Eee! You are over weight.")
else:
    print("Seesh! You are obese.")
```

Enter the height in cm: 177  
Enter the weight in kg: 100  
  
Your Body Mass Index is 31.91930798940279  
Seesh! You are obese.

```
[6]: #Create a program that validates a password based on complexity rules (length,
      ↳ characters, etc.).
def password_check(passwd):

    SpecialSym = ['$%', '@', '#', '%']
    val = True

    if len(passwd) < 6:
        print('length should be at least 6')
        val = False

    if len(passwd) > 20:
        print('length should be not be greater than 8')
        val = False

    if not any(char.isdigit() for char in passwd):
        print('Password should have at least one numeral')
        val = False
```

```

if not any(char.isupper() for char in passwd):
    print('Password should have at least one uppercase letter')
    val = False

if not any(char.islower() for char in passwd):
    print('Password should have at least one lowercase letter')
    val = False

if not any(char in SpecialSym for char in passwd):
    print('Password should have at least one of the symbols $@#')
    val = False
if val:
    return val

# Main method
def main():
    passwd = 'Geek12@'

    if (password_check(passwd)):
        print("Password is valid")
    else:
        print("Invalid Password !!")

# Driver Code
if __name__ == '__main__':
    main()

```

Password is valid

```

[8]: #Develop a program that performs matrix addition and subtraction based on user
      ↪ input.
      # Program to add two matrices using nested loop

X = [[12,7,3],
      [4 ,5,6],
      [7 ,8,9]]

Y = [[5,8,1],
      [6,7,3],
      [4,5,9]]

result = [[0,0,0],
           [0,0,0],
           [0,0,0]]

# iterate through rows

```

```

for i in range(len(X)):
    for j in range(len(X[0])):
        result[i][j] = X[i][j] + Y[i][j]

for r in result:
    print(r)

```

```

[17, 15, 4]
[10, 12, 9]
[11, 13, 18]

```

[9]: *#Write a program that calculates the greatest common divisor (GCD) of two numbers using the Euclidean*

```

#algorithm.
def gcd(x, y):
    gcd = 1
    if x % y == 0:
        return y
    for k in range(int(y / 2), 0, -1):
        if x % k == 0 and y % k == 0:
            gcd = k
            break
    return gcd
print("GCD of 12 & 17 =",gcd(12, 17))
print("GCD of 4 & 6 =",gcd(4, 6))
print("GCD of 336 & 360 =",gcd(336, 360))

```

```

GCD of 12 & 17 = 1
GCD of 4 & 6 = 2
GCD of 336 & 360 = 24

```

[10]: *#Build a program that performs matrix multiplication using nested loops and conditional statements.*

```

X = [[1,2,3],
      [4,5,6],
      [7,8,9]]

Y = [[10,11,12],
      [13,14,15],
      [16,17,18]]

result = [[0,0,0],
           [0,0,0],
           [0,0,0]]

# iterate through rows of X
for i in range(len(X)):

```

```

    for j in range(len(Y[0])):
        for k in range(len(Y)):
            result[i][j] += X[i][k] * Y[k][j]
for r in result:
    print(r)

```

```

[84, 90, 96]
[201, 216, 231]
[318, 342, 366]

```

[13]: *#Write a program that generates Fibonacci numbers up to a specified term using iterative methods.*  
*# Program to display the Fibonacci sequence up to n-th term*

```

def fibonacci(n):
    a,b = 0,1
    for i in range(n):
        a,b = b,a+b
    return a

print(fibonacci(6))

```

8

[14]: *#Develop a program that calculates the nth term of the Fibonacci sequence using memoization.*

```

def fibonacci(n):
    """Return the nth Fibonacci number."""
    # r[i] will contain the ith Fibonacci number
    r = [-1]*(n + 1)
    return fibonacci_helper(n, r)

def fibonacci_helper(n, r):
    """Return the nth Fibonacci number and store the ith Fibonacci number in r[i] for 0 <= i <= n."""
    if r[n] >= 0:
        return r[n]

    if (n == 0 or n == 1):
        q = n
    else:
        q = fibonacci_helper(n - 1, r) + fibonacci_helper(n - 2, r)
    r[n] = q

    return q

```

```
n = int(input('Enter n: '))

ans = fibonacci(n)
print('The nth Fibonacci number:', ans)
```

Enter n: 23

The nth Fibonacci number: 28657

[17]: #Create a program that generates a calendar for a given month and year using  
↳ conditional statements.

```
import calendar

year = int(input ("Please enter the Year: ")) # Here, it will take the year
month = int(input ("Please enter the month: ")) # Here, it will take the
↳ month

# Now, we will display the calendar
print ("The Calendar of: ", calendar.month(year, month))
```

Please enter the Year: 2023

Please enter the month: 9

The Calendar of: September 2023

```
Mo Tu We Th Fr Sa Su
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```

[18]: #Build a program that simulates a basic text-based blackjack game against the  
↳ computer.

```
import random

card_categories = ['Hearts', 'Diamonds', 'Clubs', 'Spades']
cards_list = ['Ace', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'Jack',
↳ 'Queen', 'King']
deck = [(card, category) for category in card_categories for card in cards_list]

def card_value(card):
    if card[0] in ['Jack', 'Queen', 'King']:
        return 10
    elif card[0] == 'Ace':
        return 11
    else:
```



```

        return int(card[0])

random.shuffle(deck)
player_card = [deck.pop(), deck.pop()]
dealer_card = [deck.pop(), deck.pop()]

while True:
    player_score = sum(card_value(card) for card in player_card)
    dealer_score = sum(card_value(card) for card in dealer_card)
    print("Cards Player Has:", player_card)
    print("Score Of The Player:", player_score)
    print("\n")
    choice = input('What do you want? ["play" to request another card, "stop" to stop]: ').lower()
    if choice == "play":
        new_card = deck.pop()
        player_card.append(new_card)
    elif choice == "stop":
        break
    else:
        print("Invalid choice. Please try again.")
        continue

    if player_score > 21:
        print("Cards Dealer Has:", dealer_card)
        print("Score Of The Dealer:", dealer_score)
        print("Cards Player Has:", player_card)
        print("Score Of The Player:", player_score)
        print("Dealer wins (Player Loss Because Player Score is exceeding 21)")
        break

while dealer_score < 17:
    new_card = deck.pop()
    dealer_card.append(new_card)
    dealer_score += card_value(new_card)

print("Cards Dealer Has:", dealer_card)
print("Score Of The Dealer:", dealer_score)
print("\n")

if dealer_score > 21:
    print("Cards Dealer Has:", dealer_card)
    print("Score Of The Dealer:", dealer_score)
    print("Cards Player Has:", player_card)
    print("Score Of The Player:", player_score)
    print("Player wins (Dealer Loss Because Dealer Score is exceeding 21)")
elif player_score > dealer_score:

```

```

print("Cards Dealer Has:", dealer_card)
print("Score Of The Dealer:", dealer_score)
print("Cards Player Has:", player_card)
print("Score Of The Player:", player_score)
print("Player wins (Player Has High Score than Dealer)")
elif dealer_score > player_score:
    print("Cards Dealer Has:", dealer_card)
    print("Score Of The Dealer:", dealer_score)
    print("Cards Player Has:", player_card)
    print("Score Of The Player:", player_score)
    print("Dealer wins (Dealer Has High Score than Player)")
else:
    print("Cards Dealer Has:", dealer_card)
    print("Score Of The Dealer:", dealer_score)
    print("Cards Player Has:", player_card)
    print("Score Of The Player:", player_score)
    print("It's a tie.")

```

Cards Player Has: [('Queen', 'Spades'), ('3', 'Clubs')]  
 Score Of The Player: 13

What do you want? ["play" to request another card, "stop" to stop]: stop  
 Cards Dealer Has: [('2', 'Hearts'), ('10', 'Spades'), ('9', 'Hearts')]  
 Score Of The Dealer: 21

Cards Dealer Has: [('2', 'Hearts'), ('10', 'Spades'), ('9', 'Hearts')]  
 Score Of The Dealer: 21  
 Cards Player Has: [('Queen', 'Spades'), ('3', 'Clubs')]  
 Score Of The Player: 13  
 Dealer wins (Dealer Has High Score than Player)

[19]: *#Write a program that generates the prime factors of a given number using trial\_*  
*↪division.*

```

def TrialDivision(N):

    # Initializing with the value 2
    # from where the number is checked
    i = 2

    # Computing the square root of
    # the number N
    k = int(N ** 0.5)

    # While loop till the

```

```

# square root of N
while(i<= k):

    # If any of the numbers between
    # [2, sqrt(N)] is a factor of N
    # Then the number is composite
    if(N % i == 0):
        return 0
    i += 1

# If none of the numbers is a factor,
# then it is a prime number
return 1

# Driver code
if __name__ == "__main__":
    N = 49
    p = TrialDivision(N)

# To check if a number is a prime or not
if(p):
    print("Prime")
else:
    print("Composite")

```

Composite

[ ]: