1. **[Palindrome Program In Python](https://www.edureka.co/blog/python-programs/" \l "1)**

string=input(("Enter a string:"))

if(string==string[::-1]):

print("The string is a palindrome")

else:

print("Not a palindrome")

1. [**Factorial Program In Python**](https://www.edureka.co/blog/python-programs/#2)

num = int(input("Enter a number: "))

factorial = 1

if num < 0:

print("Sorry, factorial does not exist for negative numbers")

elif num == 0:

print("The factorial of 0 is 1")

else:

for i in range(1,num + 1):

factorial = factorial\*i

print("The factorial of",num,"is",factorial)

1. [**Fibonacci Series Program**](https://www.edureka.co/blog/python-programs/#3)

n\_terms = int(input ("Enter the number of terms to be printed: "))

n\_1 = 0

n\_2 = 1

count = 0

if n\_terms <= 0:

print ("Please enter a positive integer, the given number is not valid")

elif n\_terms == 1:

print ("The Fibonacci sequence of the numbers up to", n\_terms, ": ")

print(n\_1)

else:

print ("The fibonacci sequence of the numbers is:")

while count < n\_terms:

print(n\_1, end=", ")

nth = n\_1 + n\_2

n\_1 = n\_2

n\_2 = nth

count += 1

1. [**Armstrong Number Program In Python**](https://www.edureka.co/blog/python-programs/#4)

n = int(input("Enter a number: "))

sum = 0

temp=n

while n > 0:

dig= n%10

sum = sum + (dig \*\* 3)

n=n//10

if temp == sum:

print(temp,"is an Armstrong number")

else:

print(temp,"is not an Armstrong number")

1. [**Calculator Program**](https://www.edureka.co/blog/python-programs/#5)

print ("Select the choice\n1. Add\n2. Sub\n3. Multiply\n4. Divide")

choice = int(input("Enter the choice:"))

A = int(input("Enter first number: "))

B = int(input("Enter second number: "))

if choice == 1:

ans=A+B

elif choice == 2:

ans = A-B

elif choice == 3:

ans = A\*B

elif choice == 4:

ans = A/B

else:

print("Invalid input")

print("the answer is", ans)

1. [**Patterns Program In Python**](https://www.edureka.co/blog/python-programs/#6)

for i in range(5):

for j in range(5):

print("\*", end="")

print()

1. **Leap Year Program**

year = int(input('enter year: '))

if year % 400 == 0:

print('it is a leap year')

elif year % 4 == 0:

print('it is a leap year')

elif year % 100 == 0:

print('not a leap year')

else:

print('not a leap year')

1. [**Prime Number Program In Python**](https://www.edureka.co/blog/python-programs/#8)

num = int(input('enter number: '))

for i in range(2, num):

if num% i == 0:

print('not a prime number')

break

else:

print('prime number')

1. [**Program To Find Area In Python**](https://www.edureka.co/blog/python-programs/#9)

print ("1. cude\n2. sphere\n3. cylinder")

a=int(input("Enter the choice: "))

if a == 1:

s=float(input("Enter side of cude: "))

print(6\*(s\*\*3))

elif a == 2:

r= float(input("Enter radius of sphere: "))

print(2\*3.14\*(r\*\*2))

elif a == 3:

r1=float(input("Enter the radius of cylinder: "))

h=float(input("Enter height of cylinder: "))

print(2\*3.14\*(r1+h))

else:

print("Enter the correct choice")

1. [**Program To Reverse A List**](https://www.edureka.co/blog/python-programs/#10)

a = [1,2,3,4,5,6,7,8,9,10]

print(a[: : -1])

1. **Program to find sum of all elements in list**

numbers = [3, 8, 3, 8, 4, 5, 30, 6, 16]

sum = 0

for i in numbers:

sum = sum + i

print("The sum is", sum)

1. **Find average, max, min of list elements**

list = [3, 12, 5, 33, 13, 6, 27]

max\_value = max(list)

min\_value = min(list)

avg\_value = 0 if len(list) == 0 else sum(list) / len(list)

print("average: ",avg\_value)

print("max value: ",max\_value)

print("min value: ",min\_value)

1. **Write a Python program to create a dictionary grouping a sequence of key-value pairs into a dictionary of lists.**

def grouping\_dictionary(l):

result = {}

for k, v in l:

result.setdefault(k, []).append(v)

return result

colors = [('yellow', 1), ('blue', 2), ('yellow', 3), ('blue', 4), ('red', 1)]

print("Original list:")

print(colors)

print("\nGrouping a sequence of key-value pairs into a dictionary of lists:")

print(grouping\_dictionary(colors))

1. **Write a Python program to convert more than one list to nested dictionary.**

def nested\_dictionary(l1, l2, l3):

result = [{x: {y: z}} for (x, y, z) in zip(l1, l2, l3)]

return result

student\_id = ["S001", "S002", "S003", "S004"]

student\_name = ["Adina Park", "Leyton Marsh", "Duncan Boyle", "Saim Richards"]

student\_grade = [85, 98, 89, 92]

print("Original strings:")

print(student\_id)

print(student\_name)

print(student\_grade)

print("\nNested dictionary:")

ch = 'a'

print(nested\_dictionary(student\_id, student\_name, student\_grade))

1. **Python program to check if a set is a subset of another set.**

print("Check if a set is a subset of another set, using comparison operators and issubset():\n")

setx = set(["apple", "mango"])

sety = set(["mango", "orange"])

setz = set(["mango"])

print("x: ",setx)

print("y: ",sety)

print("z: ",setz,"\n")

print("If x is subset of y")

print(setx <= sety)

print(setx.issubset(sety))

print("If y is subset of x")

print(sety <= setx)

print(sety.issubset(setx))

print("\nIf y is subset of z")

print(sety <= setz)

print(sety.issubset(setz))

print("If z is subset of y")

print(setz <= sety)

print(setz.issubset(sety))

1. **Write a Python program to create a symmetric difference and set difference**

setc1 = set(["green", "blue"])

setc2 = set(["blue", "yellow"])

print("Original sets:")

print(setc1)

print(setc2)

r1 = setc1.symmetric\_difference(setc2)

print("\nSymmetric difference of setc1 - setc2:")

print(r1)

r2 = setc2.symmetric\_difference(setc1)

print("\nSymmetric difference of setc2 - setc1:")

print(r2)

setn1 = set([1, 1, 2, 3, 4, 5])

setn2 = set([1, 5, 6, 7, 8, 9])

print("\nOriginal sets:")

print(setn1)

print(setn2)

r1 = setn1.symmetric\_difference(setn2)

print("\nSymmetric difference of setn1 - setn2:")

print(r1)

r2 = setn2.symmetric\_difference(setn1)

print("\nSymmetric difference of setn2 - setn1:")

print(r2)

1. **Write a Python program to remove an empty tuple(s) from a list of tuples.**

L = [(), (), ('',), ('a', 'b'), ('a', 'b', 'c'), ('d')]

L = [t for t in L if t]

print(L)

1. **Write a Python program to calculate the average value of the numbers in a given tuple of tuples.**

def average\_tuple(nums):

result = [sum(x) / len(x) for x in zip(\*nums)]

return result

nums = ((10, 10, 10, 12), (30, 45, 56, 45), (81, 80, 39, 32), (1, 2, 3, 4))

print ("Original Tuple: ")

print(nums)

print("\nAverage value of the numbers of the said tuple of tuples:\n",average\_tuple(nums))

nums = ((1, 1, -5), (30, -15, 56), (81, -60, -39), (-10, 2, 3))

print ("\nOriginal Tuple: ")

print(nums)

print("\nAverage value of the numbers of the said tuple of tuples:\n",average\_tuple(nums))

1. **Write a Python program to check the validity of a password (input from users).**

"""

Validation :

At least 1 letter between [a-z] and 1 letter between [A-Z].

At least 1 number between [0-9].

At least 1 character from [$#@].

Minimum length 6 characters.

Maximum length 16 characters.

"""

import re

p= input("Input your password")

x = True

while x:

if (len(p)<6 or len(p)>12):

break

elif not re.search("[a-z]",p):

break

elif not re.search("[0-9]",p):

break

elif not re.search("[A-Z]",p):

break

elif not re.search("[$#@]",p):

break

elif re.search("\s",p):

break

else:

print("Valid Password")

x=False

break

if x:

print("Not a Valid Password")