

Programming Questions

272. Write a program to print one of the words *negative*, *zero*, or *positive*, according to whether variable x is less than zero, zero, or greater than zero, respectively.

[Textbook Q. 1, Chapter 1 (Type C)]

Ans.

```
n = int(input("Enter a number: "))
if n == 0:
    print("Zero")
elif n > 0:
    print("Positive")
else:
    print("Negative")
```

273. Write a program that returns True if the input number is an even number, False otherwise.

[Textbook Q. 2, Chapter 1 (Type C)]

Ans.

```
n = int(input("Enter a number:"))
def checkEven(n):
    return ((n % 2) == 0)
print(checkEven(n))
```

274. Write a Python program that calculates and prints the number of seconds in a year.

[Textbook Q. 3, Chapter 1 (Type C)]

Ans.

```
days = 365
hours = 24
minutes = 60
seconds = 60

total_sec = days * hours * minutes * seconds

print("Total seconds in a year are", total_sec)
```

275. Write a Python program that accepts two integers from the user and prints a message saying if first number is divisible by second number or if it is not.

[Textbook Q. 4, Chapter 1 (Type C)]

Ans.

```
x = int(input("Enter first number: "))
y = int(input("Enter second number: "))

if y == 0:
    print("Wrong input. Second number cannot be zero")
    exit(0)
if x % y == 0:
    print(x, "is divisible by", y)
else:
    print(x, "is not divisible by", y)
```

276. Write a program that creates a list of all the integers less than 100 that are multiples of 3 or 5.

[Textbook Q. 6, Chapter 2 (Type C)]

Ans.

```
for i in range(1, 101):
    if i % 3 == 0 or i % 5 == 0:
        print(i)
```

277. Define two variables first and second so that first = "Jimmy" and second = "Johny". Write a short Python code segment that swaps the values assigned to these two variables and prints the results.

[Textbook Q. 7, Chapter 2 (Type C)]

Ans.

```
first = "Jimmy"
second = "Johny"
```

```
# swap
first, second = second, first
# print
print(first, second)
```

278. Write a program that prints a table on two columns – table that helps converting miles into kilometres. [Textbook Q. 9, Chapter 1 (Type C)]

Ans.

```
for i in range(1, 10):
    print(i, i * 1.6)
```

279. Write another program printing a table with two columns that helps convert pounds in kilograms. [Textbook Q. 10, Chapter 1 (Type C)]

Ans.

```
for i in range(1, 10):
    print(i, i * 0.454)
```

280. Write a function that takes amount-in-dollars and dollar-to-rupee conversion price; it then returns the amount converted to rupees. Create the function in both void and non-void forms. [Textbook Q. 1, Chapter 3 (Type C)]

Ans.

```
def DollarToRupee(dollars, exchangeRate):
    return dollars * exchangeRate
```

281. Write a function to calculate volume of a box with appropriate default values for its parameters. Your function should have the following input parameters :

(a) length of box ; (b) width of box ; (c) height of box.

Test it by writing complete program to invoke it. [Textbook Q. 2, Chapter 3 (Type C)]

Ans.

```
def volume(length = 1, breadth = 1, height = 1):
    return length * breadth * height
```

```
l = 2
```

```
b = 3
```

```
h = 0.5
```

```
print(volume(l, b, h))
```

282. Write a function namely nthRoot() that receives two parameters x and n and returns n th root of x i.e., $x^{\frac{1}{n}}$.

The default value of n is 2.

[Textbook Q. 6, Chapter 3 (Type C)]

Ans.

```
def nthRoot(x, n = 2):
    return x ** (1 / n)
```


TYPE C

LONG ANSWER QUESTIONS

[3, 4 Marks]

283. Write a program that asks the user the day number in a year in the range 2 to 365 and asks the first day of the year – Sunday or Monday or Tuesday etc. Then the program should display the day on the day-number that has been input.

[Textbook Q. 5, Chapter 1 (Type C)]

Ans.

```

days = {'Monday': 0, 'Tuesday': 1, 'Wednesday': 2, 'Thursday': 3, 'Friday': 4,
        'Saturday': 5, 'Sunday': 6}
n = int(input("Enter the day of the year:"))
added = (n - 1) % 7
first = input("First day of the year: ")
while first not in days:
    print("Please enter a correct day with first letter capital, e.g. Monday")
    first = input("First day of the year: ")
current_day = days[first] + added
for day in days:
    if days[day] == current_day:
        print(day)
        break

```

284. One foot equals 12 inches. Write a function that accepts a length written in feet as an argument and returns this length written in inches. Write a second function that asks the user for a number of feet and returns this value. Write a third function that accepts a number of inches and displays this to the screen. Use these three functions to write a program that asks the user for a number of feet and tells them the corresponding number of inches.

[Textbook Q. 6, Chapter 1 (Type C)]

Ans.

```

def feetToInches(n):
    return n * 12

def inputFeet():
    return int(input("Enter height in feet: "))

def displayInches(inches):
    print("The height in inches is", inches)
displayInches(feetToInches(inputFeet()))

```

285. Write a program that reads an integer N from the keyboard computes and displays the sum of the numbers from N to (2 * N) if N is nonnegative. If N is a negative number, then it's the sum of the numbers from (2 * N) to N. The starting and ending points are included in the sum.

Ans.

```

N = int(input("Enter N: "))
step = N // abs(N)

```

[Textbook Q. 7, Chapter 1 (Type C)]

```

sum = 0
for i in range(N, 2*N + step, step):
    sum += i
print(sum)

```

286. Write a program that reads a date as an integer in the format MMDDYYYY. The program will call a function that prints print out the date in the format <Month Name> <day>, <year>.

Sample run :

Enter date : 12252019

December 25, 2019

[Textbook Q. 8, Chapter 1 (Type C)]

Ans.

```

date = input("Enter date in MMDDYYYY format: ")
def prettyPrint(date):
    months = {1: 'January', 2: 'February', 3: 'March', 4: 'April', 5: 'May', 6:
              'June', 7: 'July', 8: 'August', 9: 'September', 10:
              'October', 11: 'November', 12: 'December'}
    month = months[int(date[:2])]
    day = date[2:4]
    year = date[4:]
    prettyDate = month + " " + day + ", " + year
    print(prettyDate)
prettyPrint(date)

```

287. Write a program that reads two times in military format (0900, 1730) and prints the number of hours and minutes between the two times.

A sample run is being given below :

Please enter the first time : 0900

Please enter the second time : 1730

8 hours 30 minutes

[Textbook Q. 11, Chapter 1 (Type C)]

Ans.

```

time1 = input("Time 1: ")
time2 = input("Time 2: ")
small = min(int(time1), int(time2))
big = max(int(time1), int(time2))
hours = 0
minutes = 0
while small < big:
    small += 1
    mints = small % 100
    if mints == 60:
        small += 100
        small //= 100

```



```

    small *= 100
    minutes += 1
    hours = minutes // 60
    minutes = minutes - (hours * 60)
    print(hours, "hours", minutes, "minutes")

```

288. Write a program that prompts for a phone number of 10 digits and two dashes, with dashes after the area code and the next three numbers. *For example*, 017-555-1212 is a legal input.

Display if the phone number entered is valid format or not and display if the phone number is valid or not (*i.e.*, contains just the digits and dash at specific places).

[Textbook Q. 1, Chapter 2 (Type C)]

Ans.

```

phoneNumber = input("Enter phone number: ")

def checkNumber(num):
    if len(num) != 12:
        return False
    if num[3] != '-':
        return False
    num = num[:3] + num[4:]
    if num[6] != '-':
        return False
    num = num[:6] + num[7:]
    return num.isdigit()

if checkNumber(phoneNumber):
    print("Valid")
else:
    print("Invalid")

```

289. Write a program that should prompt the user to type some sentence(s) followed by "enter". It should then print the original sentence(s) and the following statistics relating to the sentence(s) :

- Number of words
- Number of characters (including white-space and punctuation)
- Percentage of characters that are alpha numeric

Hints

- Assume any consecutive sequence of non-blank characters in a word.

[Textbook Q. 2, Chapter 2 (Type C)]

Ans.

```

para = input("Enter sentence(s): ")

words = 0
characters = len(para)
alphaNum = 0

```

```
splittedPara = para.split(' ')
for i in splittedPara:
    if i != '':
        words += 1
for i in para:
    if i.isalnum():
        alphaNum += 1
print("Words:", words)
print("Characters:", characters)
print("Percentage alpha numeric:", alphaNum * 100 / characters)
```

290. Write a program that takes any two lists L and M of the same size and adds their elements together to form a new list N whose elements are sums of the corresponding elements in L and M . For instance, if $L = [3, 1, 4]$ and $M = [1, 5, 9]$, then N should equal $[4, 6, 13]$. [Textbook Q. 3, Chapter 2 (Type C)]

Ans.

```
L = [3, 1, 4]
M = [1, 5, 9]
ln = len(L)
N = []
for a in range(ln):
    N.append(L[a] + M[a])
print(N)
```

291. Write a program that rotates the elements of a list so that the element at the first index moves to the second index, the element in the second index moves to the third index, etc., and the element in the last index moves to the first index. [Textbook Q. 4, Chapter 2 (Type C)]

Ans.

```
lis = eval(input("Enter list:"))
last = lis[-1]
for i in range(len(lis) - 1, 0, -1):
    lis[i] = lis[i - 1]
lis[0] = last
print(lis)
```

292. Write a short Python code segment that prints the longest word in a list of words. [Textbook Q. 5, Chapter 2 (Type C)]

Ans.

```
lis = input("Enter a space separated list of words: ").split()
longest = ""
maxlen = 0
for i in lis:
    if len(i) > maxlen:
        maxlen = len(i)
        longest = i
print(longest)
```

293. Write a Python program that creates a tuple storing first 9 terms of Fibonacci series.

[Textbook Q. 8, Chapter 2 (Type C)]

Ans.

```
prev1 = 1
prev2 = 1
fib = (1, 1,)
for i in range(3, 10):
    next = prev1 + prev2
    fib += (next,)
    prev1 = prev2
    prev2 = next
print(fib)
```

294. Create a dictionary whose keys are *month names* and whose values are the *number of days* in the corresponding months.

- Ask the user to enter a month name and use the dictionary to tell them how many days are in the month.
- Print out all of the keys in alphabetical order.
- Print out all of the months with 31 days.
- Print out the (key-value) pairs sorted by the number of days in each month.

[Textbook Q. 9, Chapter 2 (Type C)]

Ans.

```
daysInMonths = {'January': 31, 'February': 28, 'March': 31, 'April': 30,
                 'May': 31, 'June': 30, 'July': 31, 'August': 31,
                 'September': 30, 'October': 31, 'November': 30, 'December': 31}

# PART A
month = input("Enter month:")
while month not in daysInMonths:
    print("Please enter valid month with first letter capital")
    month = input("Enter month: ")
print("The number of days in", month, "are", daysInMonths[month])

# PART B
for i in sorted(daysInMonths.keys()):
    print(i)

# PART C
for i in daysInMonths:
    if daysInMonths[i] == 31:
        print(i)

# PART D
for i in sorted(daysInMonths, key = daysInMonths.get):
    print(i, daysInMonths[i])
```


295. Write a function called `addDict(dict1, dict2)` which computes the union of two dictionaries. It should return a new dictionary, with all the items in both its arguments (assumed to be dictionaries). If the same key appears in both arguments, feel free to pick a value from either. [Textbook Q. 10, Chapter 2 (Type C)]

Ans.

```
def addDict(dict1, dict2):
    newDict = {}
    for i in dict1:
        newDict[i] = dict1[i]
    for i in dict2:
        newDict[i] = dict2[i]
    return newDict
```

296. Write a program to sort a dictionary's keys using Bubble sort and produce the sorted keys as a list. [Textbook Q. 11, Chapter 2 (Type C)]

Ans.

```
def bubbleSort(arr):
    for i in range(len(arr) - 1):
        for j in range(len(arr) - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
    return arr

def sortDictKeys(dictionary):
    return bubbleSort(dictionary.keys())

def sortDictValues(dictionary):
    return bubbleSort(list(dictionary.values()))
```

297. Write a program to have following functions :

(i) a function that takes a number as argument and calculates cube for it. The function does not return a value. If there is no value passed to the function in function call, the function should calculate cube of 2.

(ii) a function that takes two char arguments and returns True if both the arguments are equal otherwise False.

Test both these functions by giving appropriate function call statements.

[Textbook Q. 3, Chapter 3 (Type C)]

Ans.

```
def cube(x = 2):
    print(x ** 3)
def comp(a, b):
    return a == b
cube(5)
cube()
print(comp('a', 'a'))
print(comp('z', 'x'))
```

298. Write a function that receives two numbers and generates a random number from that range. Using this function, the main program should be able to print three numbers randomly. [Textbook Q. 4, Chapter 3 (Type C)]

Ans.

```
import random

def randomInRange(a, b):
    return random.randrange(a, b)

for i in range(3):
    print(randomInRange(1, 10))
```

299. Write a function that takes a number n and then returns a randomly generated number having exactly n digits (not starting with zero) e.g., if n is 2 then function can randomly return a number 10-99 but 07, 02 etc., are not valid two digit numbers. [Textbook Q. 7, Chapter 3 (Type C)]

Ans.

```
import random

def randomNDigitNumber(n):
    num = ""
    firstIteration = True
    for i in range(n):
        randomDigit = ""
        if firstIteration:
            randomDigit = str(random.randint(1, 9))
            firstIteration = False
        else:
            randomDigit = str(random.randint(0, 9))
        num = randomDigit + num
    return num

print(randomNDigitNumber(7))
```

300. Write a function that takes two numbers and returns the number that has minimum one's digit.

[For example, if numbers passed are 491 and 278, then the function will return 491 because it has got minimum one's digit out of two given numbers (491's 1 is < 278's 8)].

[Textbook Q. 8, Chapter 3 (Type C)]

Ans.

```
def smallUnitPlace(x, y):
    unitX = x % 10
    unitY = y % 10
    if (unitX < unitY):
        return x
    else:
        return y
```


301. Write a program that generates a series using a function which takes first and last values of the series and then generates four terms that are equidistant e.g., if two numbers passed are 1 and 7 then function returns 1 3 5 7.

[Textbook Q. 9, Chapter 3 (Type C)]

Ans.

```
def completeSeries(a, l):
    d = (l - a) / 3
    return (a, a + d, a + 2*d, a + 3*d,)
print(completeSeries(1, 7))
```

302. Create a module lengthconversion.py that stores functions for various lengths conversion e.g.,

[Textbook Q. 1, Chapter 4 (Type C)]

- miletokm() to convert miles to kilometer
- kmtomile() to convert kilometers to miles
- feettoinches()
- inchestofeet()

It should also store constant values such as value of (mile in kilometers and vice versa).

[1 mile = 1.609344 kilometer ; 1 feet = 12 inches]

Help() function should display proper information.

Ans.

```
#lengthconversion.py
'''This module stores various conversion functions
to convert distances into different units'''

def miletokm( d ):
    # conversion function 1
    '''returns miles converted to kilometers'''
    return d * ONE_MILE

def kmtomile( d ):
    # conversion function 2
    '''returns kilometers converted to miles'''
    return d / ONE_MILE

def feettoinches( ln ):
    # conversion function 3
    '''returns feet converted to inches'''
    return ln * ONE_FEET

def inchestofeet( ln ):
    # conversion function 4
    '''returns inches converted to feet'''
    return ln / ONE_FEET

#constants
ONE_MILE = 1.609344      # = 1.609Km
ONE_FEET = 12           # = 12 inches
```


303. Create a module `MassConversion.py` that stores function for mass conversion e.g.,

- `kgtotonne()` to convert kg to tonnes
- `tonnetokg()` to convert tonne to kg
- `kgtopound()` to convert kg to pound
- `poundtokg()` to convert pound to kg

(Also store constants $1 \text{ kg} = 0.001 \text{ tonne}$, $1 \text{ kg} = 2.20462 \text{ pound}$)

`Help()` function should give proper information about the module.

[Textbook Q. 2, Chapter 4 (Type C)]

Ans.

```
#MassConversion.py
'''This module stores various conversion functions
to convert masses into different units'''

def kgtotonne( m ):
    # conversion function 1
    '''returns kilograms converted to tonnes'''
    return m * ONE_KG_TONNE

def tonnetokg( m ):
    # conversion function 2
    '''returns tonnes converted to kilograms'''
    return m / ONE_KG_TONNE

def kgtopound( m ):
    # conversion function 3
    '''returns kilograms converted to pounds'''
    return m * ONE_KG_POUND

def poundtokg( m ):
    # conversion function 4
    '''returns pounds converted to kilograms'''
    return m / ONE_KG_POUND

#constants
ONE_KG_TONNE = 0.001          # = 0.001 tonne
ONE_KG_POUND = 2.20462       # = 2.20462 pound
```