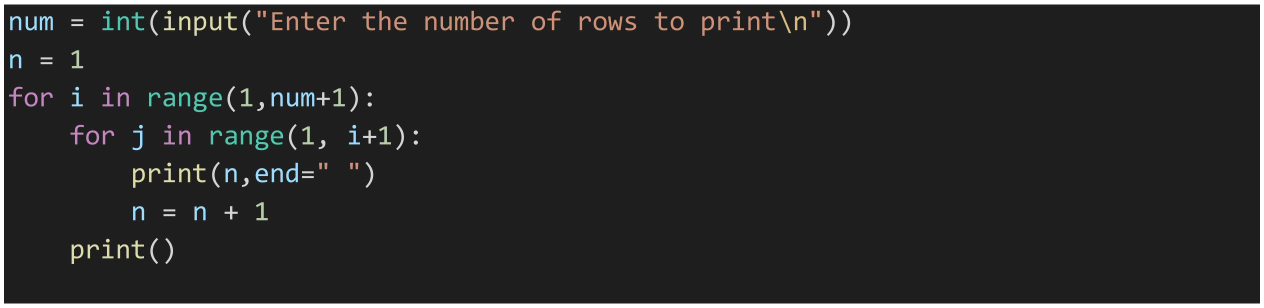
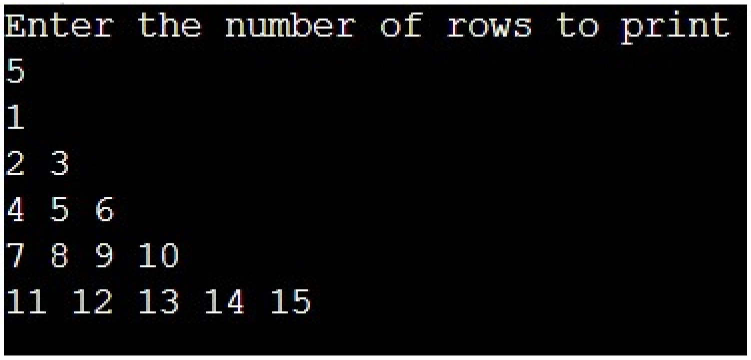
Question Bank For Python Programming Language

1) **Write a program to print Floyd's Triangle.**
   Code:



```python
num = int(input("Enter the number of rows to print\n"))
n = 1
for i in range(1,num+1):
    for j in range(1, i+1):
        print(n,end=" ")
        n = n + 1
    print()
```

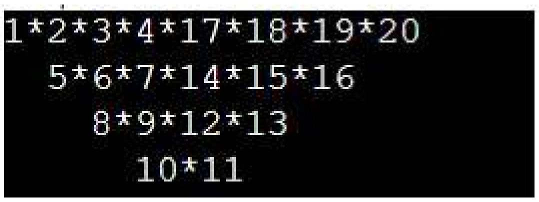
Output:



```
Enter the number of rows to print
5
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

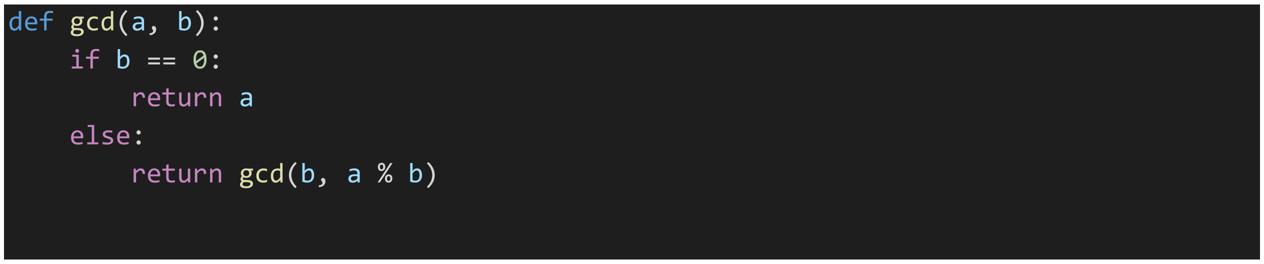
2) **Write a program to print Trapezium Pattern.**
   Code:

Output:



```
1*2*3*4*17*18*19*20
  5*6*7*14*15*16
    8*9*12*13
      10*11
```

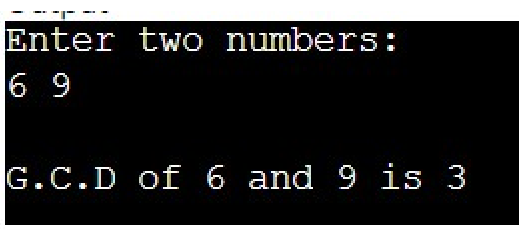
3) **Write a python program to find the GCD of two numbers.**
   Code:



```python
def gcd(a, b):
    if b == 0:
        return a
    else:
        return gcd(b, a % b)
```

```python
a,b = map(int,input("Enter two numbers:\n").split())

print(f"The gcd of {a} and {b} is  ", end="")
print(gcd(a,b))
```



```
Enter two numbers:
6 9

G.C.D of 6 and 9 is 3
```

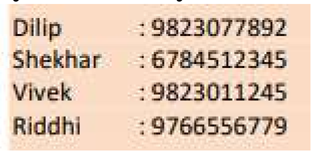
4) **Write a program to print the Fibonacci series.**
Code:

```python
terms = int(input("Enter the number of terms:\n"))


def fib(num):
    if num == 1 or num == 0:
        return num
    elif num < 0:
        print("Invalid input")
    else:
        end_answer = fib(num - 1) + fib(num - 2)
        return end_answer


print(f"Fibonacci series for {terms} terms is: {fib(num=terms)}")
```

5) **Write a program to maintain names and cell numbers of 4 persons and then print them systematically in a tabular form.**



Dilip	: 9823077892
Shekhar	: 6784512345
Vivek	: 9823011245
Riddhi	: 9766556779

Code:

```python
list_ = []
list__ = []
for i in range(4):
    name = input(f"Enter the {i+1} name:\n")
    number = input(f"Enter the number of {name}:\n")
    list_.append(name)
    list__.append(number)

contacts = {list_[i]:list__[i] for i in range(len(list_))}
for name, cellno in contacts.items():
    print(f'{name:10}:{int(cellno):10d}')
```

6) **Write a program to print out all Armstrong numbers between 1 and 500. If sum of cubes of each digit of the number is equal to the number itself, then the number is called an Armstrong number. For example, 153 = (1 * 1 * 1) + (5 * 5 * 5) + (3 * 3 * 3).**

   Code:

```python
# unsorted
for num in range(1, 501):
    order = len(str(num))
    sum = 0
    temp = num
    while temp > 0:
        digit = temp % 10
        sum += digit ** order
        temp //= 10

    if num == sum:
        print(num)
```

7) **Write a Python program to convert two lists into a dictionary in a way that item from list1 is the key and item from list2 is the values.**

Code:

```python
list_x = [1,2,3,4]
list_x2 = [5,6,7,8]
if len(list_x) == len(list_x2):
    Dict = {list_x[i]: list_x2[i] for i in range(len(list_x))}
    print(Dict)
else:
    print("Lengths of list are not equal")
```

8) Write a program to find all occurrences of 'T' in the string 'The Terrible Tiger Tore The Towel'. Replace all occurrences of 'T' with 't'. (let us python – pg no 51)

Code:

```
string1 = 'The Terrible Tiger Tore The Towel'

print(string1.count("T"))
print(string1.replace("T","t"))
```

9) Write a Python program to create a dictionary storing shopping items in cart with details such as - (item name: cost of item) and perform the following operations

i. Add a new item
ii. Display item value greater than 100
iii. Remove the details of the specific item
iv. Removes the last item details from cart.

Code:

```
item_list = ["Chips","kurkure","Pasta","Maggie","Soup"]
item_price = [60,20,100,80,120]
shopping = {item_list[i]:item_price[i] for i in range(len(item_price))}

wanna_try_again = "Y"

while wanna_try_again:
    print(f"Initial Dictionary: {shopping}")
    new_item = input("Enter the name of the item you want to add:\n")
    new_item_price = int(input(f"Enter the price of {new_item}:\n"))
    shopping[new_item] = new_item_price
    print(f"Dictionary after adding {new_item} and {new_item_price}: {shopping}")
    value = {i for i in shopping if shopping[i] > 100}
    print("Products with price more than 100:\n", value)
    which_one = input("Enter the name of the item you want to remove from the
cart:\n")
    shopping.pop(which_one)
    print(f"Dictionary after removing {which_one}: {shopping}")
    shopping.popitem()
    print(f"Dictionary after removing last item: {shopping}")
    wanna_try_again = input("Enter 'Y to perform again:\n").upper()
```

10) Perform the following operations on a list of numbers. –

i. Create a list of 5 odd numbers
ii. Create a list of 5 even numbers
iii. Combine the two lists
iv. Add prime numbers 11, 17, 29 at the beginning of the combined list
v. Report how many elements are present in the list
vi. Replace last 3 numbers in the list with 100, 200, 300
vii. Reverse the list.
viii. Delete the list

Code:

```
odd5 = [1,3,5,7,9]
even5 = [2,4,6,8,10]

print(f"list of 5 odd number = {odd5}.\n")
print(f"list of 5 even number = {even5}.\n")
combine = odd5 + even5
print(f"After combining {odd5} and {even5}: {combine}\n")
combine.insert(0,11)
combine.insert(1,17)
combine.insert(2,29)
print(f"After inserting 11, 17 and 29 at beginning: {combine}\n")
print(f"Number of elements in {combine} are {len(combine)}\n")
combine[len(combine)-1] = 300
combine[len(combine)-2] = 200
combine[len(combine)-3] = 100
print(f"After replacing last 3 elements in combine with 100,200 and 300:
{combine}\n")

combine.reverse()
print(f"Reversing the list:{combine}\n")

combine.clear()
print(f"After deleting the list : {combine}\n")
```

11) Write a Python function to create user module to receive five integers from keyboard and get their sum and product calculated. (let us Python pp164

Code:

```python
def func():
    a,b,c,d,e = map(int, input("Enter 5 integers:\n").split())
    calculate_sum = a+b+c+d+e
    calculate_product = a*b*c*d*e
    print(f"Sum of 5 integers in {calculate_sum}.")
    print(f"Product of 5 integers in {calculate_product}.")


func()
```

12) A list contains tuples containing roll number, names and age of student. Write a Python program to gather all the names from this list into another list. (let us python – pg117)

Code:

```python
lst = [('A101', 'Aaryan', 27), ('A104', 'Aditya', 47), ('A111', 'Atharv', 19)]
nlst = [ ]
for ele in lst:
    nlst = nlst + [ele[1]]
print(nlst)
```

13) Write a necessary program to build a database connectivity for simple student and Mini Project management System using Python.

i. Accept [Mini project name, Team members, Technologies used, Mini Project completion deadline]
ii. Display - displays the details of every Project
iii. Search - Look for a specific student and the project on which he or she is working
iv. Delete - delete a particular record (based on either project/employee
v. Update - Update the team members for a particular mini project"

Code:

```python
import mysql.connector
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Sourav@45",
    database='student'
    )
cur = mydb.cursor()

def accept():
    b = "INSERT INTO student (Srno , project_name ,team_members ,tech_used
,deadline ) VALUES(%s,%s,%s,%s,%s)"
    Srno = int(input("Enter Team Number:"))
    project_name = input("Enter project name:")
    team_members = input("Enter team members(seperate them with space):")
    tech_used = input("Enter the technology used:")
    deadline = input("Enter the deadline:")
    c = [Srno, project_name, team_members, tech_used, deadline]
    d = tuple(c)
    cur.execute(b,d)
    mydb.commit()
    print("The data you just entered is now stored in database.")

def display():
    e = "select * from student"
    cur.execute(e)
    result = cur.fetchall()
    for rec in result:
        print(rec)


def search():
    m = input("Enter the name of the student: ")
    n = "select * from student WHERE team_members = %s"
    o = [m]
    cur.execute(n,o)
    record = cur.fetchmany()
    print(record)

def delete():
    f = input("Enter the name of project you want to delete record for:")
    g = "DELETE FROM student WHERE project_name = %s"
    h = [f]
    cur.execute(g,h)
    mydb.commit()
    print(f"The records of {f} project now have been deleted from the database.")
```

```python
def update():
    i = input("Enter the team members for a particular project you want to update
data for:")
    j = input("Enter the new team members for a particular project:")
    k = "UPDATE student SET team_members = %s WHERE team_members = %s"
    l = [j, i]
    cur.execute(k, l)
    mydb.commit()
    print(f"Name of team members have been updated to {j} from {i}.")


def task1():
    task = int(input('''Choose the task from the following:
1. Accept the detail.
2. Display the details.
3. Search the detail.
4. Delete the detail.
5. Update the detail.\n'''))
    if task == 1:
        accept()
    elif task == 2:
        display()
    elif task == 3:
        search()
    elif task == 4:
        delete()
    elif task == 5:
        update()
    else:
        print("Input error! Try again.")
        task1()


task1()
```

14) Write a program to build a simple Employee and Project management System using Python functions which can perform the following operations:

i. Accept [Project name, Team members, Technologies used, Project completion deadline]
ii. Display - displays the details of every Project
iii. Search - Look for a specific employee and the project on which he or she is working
iv. Delete - delete a particular record (based on either project/employee

v. Update - Update the team members for a particular project"

Code:

```python
import mysql.connector
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Sourav@45",
    database='employee'
    )
cur = mydb.cursor()

def accept():
    b = "INSERT INTO EMPLOYEE (Srno , project_name ,team_members ,tech_used
,deadline ) VALUES(%s,%s,%s,%s,%s)"
    Srno = int(input("Enter Team Number:"))
    project_name = input("Enter project name:")
    team_members = input("Enter team members(seperate them with space):")
    tech_used = input("Enter the technology used:")
    deadline = input("Enter the deadline:")
    c = [Srno, project_name, team_members, tech_used, deadline]
    d = tuple(c)
    cur.execute(b,d)
    mydb.commit()
    print("The data you just entered is now stored in database.")

def display():
    e = "select * from employee"
    cur.execute(e)
    result = cur.fetchall()
    for rec in result:
        print(rec)

def search():
    m = input("Enter the name of the employee: ")
    n = "select * from employee WHERE team_members = %s"
    o = [m]
    cur.execute(n,o)
    record = cur.fetchmany()
    print(record)

def delete():
    f = input("Enter the name of project you want to delete record for:")
    g = "DELETE FROM employee WHERE project_name = %s"
    h = [f]
    cur.execute(g,h)
    mydb.commit()
    print(f"The records of {f} project now have been deleted from the database.")

def update():
    i = input("Enter the team members for a particular project you want to update
data for:")
    j = input("Enter the new team members for a particular project:")
    k = "UPDATE EMPLOYEE SET team_members = %s WHERE team_members = %s"
    l = [j, i]
    cur.execute(k, l)
    mydb.commit()
    print(f"Name of team members have been updated to {j} from {i}.")


def task1():
    task = int(input('''Choose the task from the following:
1. Accept the detail.
2. Display the details.
3. Search the detail.
4. Delete the detail.
5. Update the detail.\n'''))
    if task == 1:
        accept()
    elif task == 2:
        display()
    elif task == 3:
        search()
    elif task == 4:
        delete()
    elif task == 5:
        update()
    else:
        print("Input error! Try again.")
        task1()


task1()
```

15) Write a python program to demonstrate different functions of OS module in python.
(let us python – pg no 336)

Code:

```
import os
```

16) **Write a program to read the contents of file 'message' one character at a time. Print each character that is read.**(let us python – pg no 337)

Code:

```
file = open('file.txt', 'r')

while 1:
    char = file.read(1)
    if not char:
        break

    print(char)

file.close()
```

17) **Write a Python program that accepts a list of integers and checks the length and the third element. Return true if the length of the list is more than 10 and the third element occurs twice in the said list. And also Sort the list in ascending order.**

Code:

```
num_list = []
num_of_ele = int(input("Enter the number of elements:\n"))
for i in range(num_of_ele):
    ele = int(input("Enter the integer:"))
    num_list.append(ele)

print(num_list)
print(f"The length of {num_list} is: {len(num_list)}.")
print(f"The third element in {num_list} is {num_list[2]}.")

if len(num_list) > 10 and num_list.count(num_list[2]) == 2:
    print("True")
else:
    print("False")
```

```python
num_list.sort()
print(f"After sorting in ascending order : {num_list}.")
```

18) Write a generic program to handle exception generated in following scenarios:

i. Division by Zero
ii. Accessing a file which does not exist.
iii. Addition of two incompatible types
iv. Trying to access a nonexistent index of a sequence"

Code:

```python
def divide(x, y):
    try:
        result = x / y
        print("\n")
        print("Yeah ! Your answer is :", result)
    except ZeroDivisionError:
        print("\n")
        print("Sorry ! You are dividing by zero ")


print("Output A:\n")
num1 = int(input("Enter first number :"))
num2 = int(input("Enter second number :"))
divide(num1, num2)

print("\nOutput B:")

file1 = input("Enter file name :")
try:
    file = open('file1')
except Exception as e:
    print('File not found. Check the name of file.')

print("\nOutput C:")

num3 = input("Enter first number :")
num4 = int(input("Enter second number :"))
try:
    print(num3 + num4)
```

```python
except TypeError:
    print("\nInvalid input,both the inputs should be of same datatype")

print("\nOutput D:")


list_ = []
num = int(input("Enter the number of elements in the list:"))
for ele in range(num):
    num_ = input("Enter the elements:")
    list_.append(num_)

search = int(input("Enter the index:"))

try:
    print(list_[search])
except:
    print("Index error")
```

19) **Write a program that infinitely receives positive integer as input and prints its square. If a negative number is entered then raise an exception, display a relevant error message and make a graceful exit. .**(let us python – pg no 309)

Code:

```python
try:
    while True:
        num = int(input('Enter a positive number: '))
        if num >= 0:
            print(num * num)
        else:
            raise ValueError('Negative number')
except ValueError as ve:
    print(ve.args)
```

20) **Write a Python program that reports the time of creation, time of last access and time of last modification for a given file. (let us python – pg no 330)**

Code:

```python
import os
import time

file = 'sampledata'
print(file)

created = os.path.getctime(file)
modified = os.path.getmtime(file)
accessed = os.path.getatime(file)

print('Date created: ' + time.ctime(created))
print('Date modified: ' + time.ctime(modified))
print('Date accessed: ' + time.ctime(accessed))
```

21) **Write a program that receives an integer as input. If a string is entered instead of an integer, then report an error and give another chance to user to enter an integer. Continue this process till correct input is supplied. (let us python – pg no 314)**

Code:

```python
while True:
    try:
        num = int(input('Enter a number: '))
        break
    except ValueError:
        print('Incorrect Input')
print('You entered: ', num)
```