

##Write a Python program to find the largest of three numbers

```
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 45
enter second number 44
enter third number 43

The largest number is : 45.0
```

##Create a program that checks if a number is even or odd.

```
def check_even_odd(number):
    if number % 2 ==0:
        return"even"
    else:
        return "odd"
try:
    num= int(input("Enter a number:"))
    result = check_even_odd(num)
    print(f"The number{num} is {result}.")
except ValueError:
    print("Please enter a valid integer.")

Enter a number: 65

The number65 is odd.
```

Write a program that prints numbers from 1 to 100 but skips numbers divisible by 7.

```
for num in range(1,101):
    if num % 7 == 0:
        continue
    print(num)

1
2
3
4
```

5
6
8
9
10
11
12
13
15
16
17
18
19
20
22
23
24
25
26
27
29
30
31
32
33
34
36
37
38
39
40
41
43
44
45
46
47
48
50
51
52
53
54
55
57
58
59
60
61

62
64
65
66
67
68
69
71
72
73
74
75
76
78
79
80
81
82
83
85
86
87
88
89
90
92
93
94
95
96
97
99
100

Write a program to print the multiplication table of a given number.

```
def print_multiplication_table(number, limit=10):  
    print(f"Multiplication table for {number}:")  
    for i in range(1, limit+1):  
        print(f"{number}*{i}={number*i}")  
try:  
    num = int(input("Enter a number:"))  
    print_multiplication_table(num)  
except ValueError:  
    print ("Please enter a valid intger.")  
Enter a number: 89
```

Multiplication table for 89:

```
89*1=89
89*2=178
89*3=267
89*4=356
89*5=445
89*6=534
89*7=623
89*8=712
89*9=801
89*10=890
```

Write a program that uses a nested loop to print a right-angled triangle pattern of stars based on user input.

```
try:
    rows = int(input("Enter the number of rows for the triangle"))
    if rows <=0:
        print("Please enter a positive intger.")
    else:
        for i in range(1,rows+1):
            for j in range (i):
                print("*", end = " ")
            print()
except ValueError:
    print("Please enter a valid positive intger .")
```

Enter the number of rows for the triangle 5

```
*
* *
* * *
* * * *
* * * * *
```

Create a Python program that prints the following pattern using nested loops: 1 12 123 1234

```
rows = int(input("Enter the number of rows: "))
for i in range(1, rows + 1):
    print(''.join(str(j) for j in range(1, i + 1)))
```

Enter the number of rows: 5

```
1
12
123
1234
12345
```

Create a Python program to check if a character entered by the user is a vowel or consonant

```
def check_char_type(char):
    vowels = "aeiouAEIOU"

    if len(char) != 1 or not char.isalpha():
        return "invalid input. Please enter a single alphabet character. "

    if char in vowels:
        return f" {char} is a vowel."
    else :
        return f"'{char}' is a consonant."

user_input = input("Enter a single character: ")
result = check_char_type(user_input)
print(result)
```

Enter a single character: o

o is a vowel.

Write a Python program that generates a diamond pattern of stars For n = 5, the output should look like: ***** ***** ***** **

```
def print_diamond(n):
    for i in range(n):
        spaces = ' ' * (n - i - 1)
        stars = '*' * (2 * i + 1)
        print(spaces + stars)
    for i in range (n-2, -1, -1):
        spaces = ' ' * (n - i - 1)
        stars = '*' * (2 * i + 1)
        print(spaces + stars)

print_diamond(5)
```

```

    *
   ***
  *****
 *****
*****
 *****
  *****
   ***
    *
```

Write a Python program to reverse the digits of a given number

```
def reverse_number(num):
    reversed_num = int(str(num)[::-1])
```

```
        return reversed_num
number = int(input("Enter a number : "))
reversed_result = reverse_number (number)
print (f"Reversed number : { reversed_result}")
```

Enter a number : 65

Reversed number : 56

Create a Python program that generates the following pyramid pattern: 1 121 12321 1234321
123454321

```
def number_pyramid(n):
    for i in range(1, n + 1):
        ascending = ''.join(str(j) for j in range (1 , i + 1))
        descending = ''.join(str(j) for j in range(i-1, 0, -1))
        print(ascending + descending)
```

```
number_pyramid(5)
```

```
1
121
12321
1234321
123454321
```