In [1]:

```
# program to check whether the number is positive or not
num=int(input("Enter the number: "))
if(num<0):
    print("the number is negative")
else:
    print("The number is positive")
```

```
The number is positive
```

In [2]:

```
num=int(input("Enter the number: "))
if(num<0):
    print("the number is negative")
else:
    print("The number is positive")
```

```
The number is positive
```

In [3]:

```
# checking the number vowels in the words

ar=input("Enter the : ").lower()
vowels=("aeiou")
c=0
for i in ar:
    if i in vowels:
        c=c+1
print(c)
```

```
0
```

In [4]:

```
# print number from 1 to 10
for i in range (1,11):
     print(i)
```

```
1
2
3
4
5
6
7
8
9
10
```

In [5]:

```
# calculate the sum of number in a list
numbers=[10,20,30,40,50,60]
total=0
for num in numbers:
    total+=num
print("sum :",total)
```

```
sum : 210
```

In [6]:

```
# print Evan number from list
num=[105,208,17,40,5,60,70,889,907]
for i in num:
    if i % 2==0:
        print(f"{i} is evan")
    else:
        print("f{i} is not evan")
```

```
f{i} is not evan
208 is evan
f{i} is not evan
40 is evan
f{i} is not evan
60 is evan
70 is evan
f{i} is not evan
f{i} is not evan
```

In [14]:

```
# program to check whether the program is evan or odd
num=int(input("enter the number: "))
if num % 2==0:
    print("the number is evan")
else:
    print("the number is odd")
```

```
the number is evan
```

program to check number is positive negative or zero

In [16]:

```
number=int(input("please enter the number: "))
if (number>0):
    print("the entered number is positive")
elif(number<0):
    print("the entered number is negative")
else:
    print("the number is zero")
```

```
the entered number is positive
```

Removing duplicates from list

In [18]:

```
a=[1,2,3,1,4,2,6,8,2,3,5,]
unique=list(set(a))
print("list after removing duplicates : ",unique)
```

```
list after removing duplicates :  [1, 2, 3, 4, 5, 6, 8]
```

program to print pyramid pattern

In [3]:

```
rows = int(input("Enter the number of rows: "))
for i in range(1, rows + 1):
    print(" " * (rows - i) + "* " * i)
```

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

In [22]:

```
c=int(input("enter the no. of rows: "))
for i in range(1,c+1):
    print(" "* (c-i)+"* "*i)
```

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

In [24]:

```
d=int(input("enter the number : "))
for i in range(1,d+1):
    print(" "*(d-i)+"* "*i)
```

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

In [26]:

```
input=int(input("enter the number of columns: "))
for  i in range(1,input+1):
    print(" "*(input-1)+"* "*i)
```

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

program tocheck whether the string is palindrome or not

In [9]:

```
string=input("enter the string please: ")
input=string[::-1]
if (string==input):
    print("the entered string is palindrome")
else:
    print("the enterd string is not palindrome")
```

```
the entered string is palindrome
```

print a pyramid pattern

In [7]:

```
row=int(input("enter the no.of rows: "))
for i in range(1,row+1):
    print(" " *(row - i) + "* "* i)
```

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

In [13]:

```
s=13
for i in range(1,s+1):
    print(" "*(s-i)+"* "*i)
```

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

In [17]:

```
sarde=11
for i in range(1,sarde+1):
    print(" " *(sarde-i)+"* ",i
```

```
  Cell In[17], line 3
    print(" " *(sarde-i)+"* ",i
                               ^
SyntaxError: incomplete input
```

In [7]:

```
row=10
for i in range(1,row+1):
    print(" "*(row-i)+"* "*i)
```

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

In [11]:

```
s=int(input("enter the no of rows: "))
for i in range(1,s+1):
    print(" "*(s-i)+"* "*i)
```

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

In [1]:

```
row=int(input("enter the no.of rows: "))
for i in range(1,row+1):
    print(" "*(row-i)+"* "*i)
```

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

In [3]:

```
t=10
for i in range(1,t+1):
    print(" "*(t-i)+"* "*i)
```

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

In [5]:

```python
for a in range(2,21):
    for i in range(1,11):
        print(a*i)
    print()
```

```
2
4
6
8
10
12
14
16
18
20

3
6
9
12
15
18
21
24
27
30

4
8
12
16
20
24
28
32
36
40

5
10
15
20
25
30
35
40
45
50

6
12
18
24
30
36
42
48
54
60

7
14
21
28
35
42
49
56
63
70

8
```

```
16
24
32
40
48
56
64
72
80

9
18
27
36
45
54
63
72
81
90

10
20
30
40
50
60
70
80
90
100

11
22
33
44
55
66
77
88
99
110

12
24
36
48
60
72
84
96
108
120

13
26
39
52
65
78
91
104
117
130

14
28
42
56
70
84
98
112
126
140

15
30
45
60
75
90
105
```

```
120
135
150

16
32
48
64
80
96
112
128
144
160

17
34
51
68
85
102
119
136
153
170

18
36
54
72
90
108
126
144
162
180

19
38
57
76
95
114
133
152
171
190

20
40
60
80
100
120
140
160
180
200
```

In [7]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [9]:

```python
sns.get_dataset_names()
```

```
Out[9]: ['anagrams',
         'anscombe',
         'attention',
         'brain_networks',
         'car_crashes',
         'diamonds',
         'dots',
         'dowjones',
         'exercise',
         'flights',
         'fmri',
         'geyser',
         'glue',
         'healthexp',
         'iris',
         'mpg',
         'penguins',
         'planets',
         'seaice',
         'taxis',
         'tips',
         'titanic']
```

```
In [15]: data=sns.load_dataframe("diamond")
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[15], line 1
----> 1 data=sns.load_dataframe("diamond")

AttributeError: module 'seaborn' has no attribute 'load_dataframe'
```

```
In [5]: c=int(input("enter the no of stars you want: "))
        for i in range(1,c+1):
            print(" "*(c-i)+"* "*i)
```

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

```
In [9]: for a in range(2,21):
            for i in range(1,11):
                print(i*a)
            print()
```

```
2
4
6
8
10
12
14
16
18
20

3
6
9
12
15
18
21
24
27
30

4
8
12
16
20
24
28
32
36
40

5
10
15
20
25
```

```
30
35
40
45
50

6
12
18
24
30
36
42
48
54
60

7
14
21
28
35
42
49
56
63
70

8
16
24
32
40
48
56
64
72
80

9
18
27
36
45
54
63
72
81
90

10
20
30
40
50
60
70
80
90
100

11
22
33
44
55
66
77
88
99
110

12
24
36
48
60
72
84
96
108
120
```

```
13
26
39
52
65
78
91
104
117
130

14
28
42
56
70
84
98
112
126
140

15
30
45
60
75
90
105
120
135
150

16
32
48
64
80
96
112
128
144
160

17
34
51
68
85
102
119
136
153
170

18
36
54
72
90
108
126
144
162
180

19
38
57
76
95
114
133
152
171
190

20
40
60
80
100
120
```

```
140
160
180
200
```

In [11]:

```
import math as o
```

In [15]:

```
res=o.sqrt(110)
res
```

Out[15]:

```
10.488088481701515
```

In [17]:

```
import random as r
```

In [25]:

```
c=r.randint(7,56)
c
```

Out[25]:

```
42
```

In [49]:

```
import datetime
```

In [51]:

```
v=datetime.datetime.now()
v
```

Out[51]:

```
datetime.datetime(2024, 12, 25, 22, 31, 28, 33722)
```

In [76]:

```
k=v.strftime("%Y-%m-%d    %H:%M:%S")
k
```

Out[76]:

```
'2024-12-25    22:31:28'
```

area of circle

In [83]:

```
r=int(input("Enter the radius of circle: "))
area=3.14*r*r
area
```

Out[83]:

```
78.5
```

In [3]:

```
import turtle
```

In [24]:

```
import matplotlib.pyplot as plt

# Define circle parameters
radius = 1
circle = plt.circle((0,0), radius, fill=False)

# Set up the plot
fig, ax = plt.subplots()
ax.add_artist(circle)
ax.set_xlim(-1.5, 1.5)
ax.set_ylim(-1.5, 1.5)
ax.set_aspect('equal', adjustable='datalim')
plt.show()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[24], line 5
      3 # Define circle parameters
      4 radius = 1
----> 5 circle = plt.circle((0,0), radius, fill=False)
      7 # Set up the plot
      8 fig, ax = plt.subplots()

AttributeError: module 'matplotlib.pyplot' has no attribute 'circle'
```

program to check wheather number is prime or not

In [7]:

```
num=int(input("enter the number:"))
c=0
for i in range(2,num):
    if(num%i==0):
      c=1
if(c==0):
    print("the number is prime")
else:
    print("the number is not prime")
```

```
the number is prime
```

progrm to check whether the number is positive or not

In [21]:

```
num=int(input("enter the number:"))
print(f"{num} is positive" if num>=0 else f"{num} is negative")
```

```
45 is positive
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js