Name NUZHAT QURESHI
Roll No BIT-24S-029
Github link [NuzhatQureshi90/Pythons-lab-manual-IT-A-: Python lab manuals]

LAB MANUAL # 01

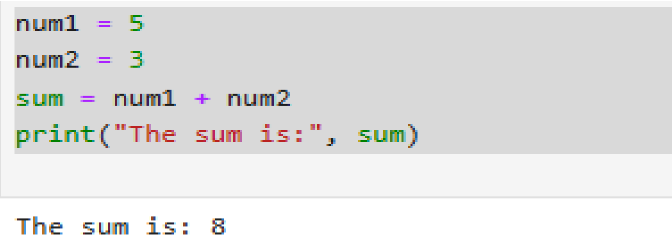
TASK 01

Q-1 Make 2-2 programs of each datatype.

1. NUMERIC TYPES:

- Integer:

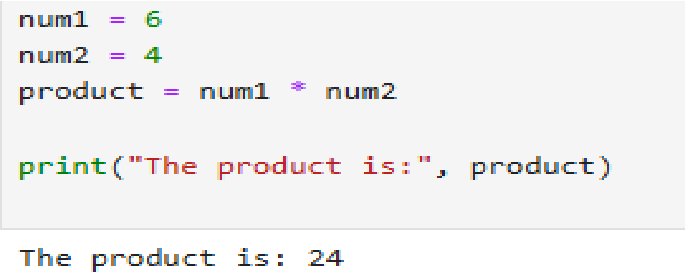
Program:



```
num1 = 5
num2 = 3
sum = num1 + num2
print("The sum is:", sum)
```

```
The sum is: 8
```

Program:

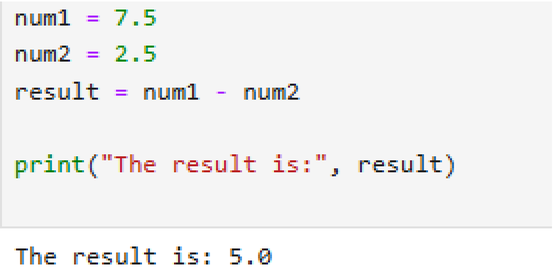


```
num1 = 6
num2 = 4
product = num1 * num2

print("The product is:", product)
```

```
The product is: 24
```

- Float:

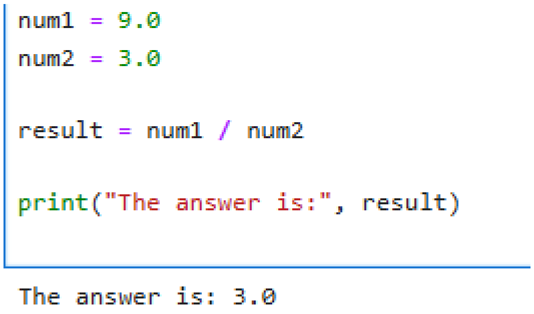
Program:



```
num1 = 7.5
num2 = 2.5
result = num1 - num2

print("The result is:", result)
```

```
The result is: 5.0
```

Program:

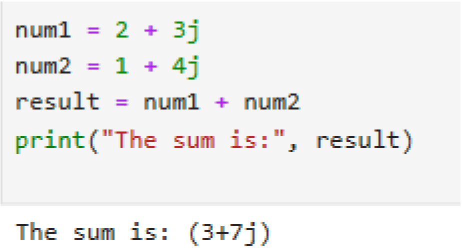


```
num1 = 9.0
num2 = 3.0

result = num1 / num2

print("The answer is:", result)
```

```
The answer is: 3.0
```

- Complex:

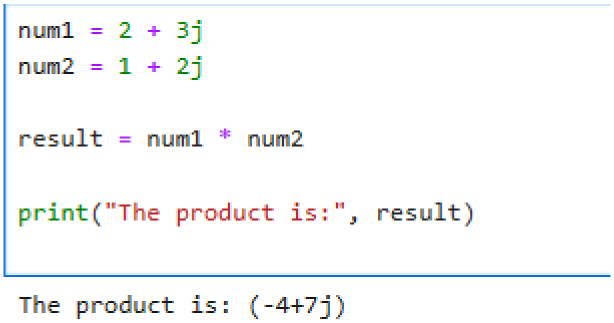
Program:



```
num1 = 2 + 3j
num2 = 1 + 4j
result = num1 + num2
print("The sum is:", result)
```

```
The sum is: (3+7j)
```

Program:



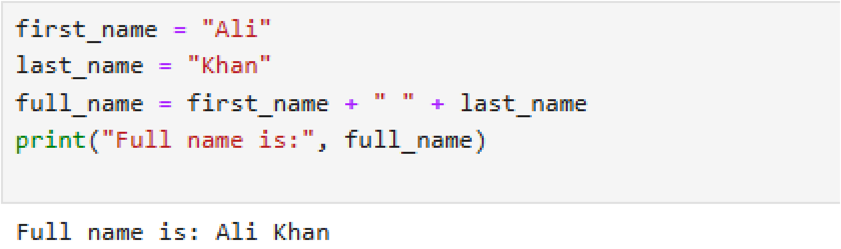
```
num1 = 2 + 3j
num2 = 1 + 2j

result = num1 * num2

print("The product is:", result)
```

```
The product is: (-4+7j)
```

2- <u>SEQUENCE TYPES:</u>

- String

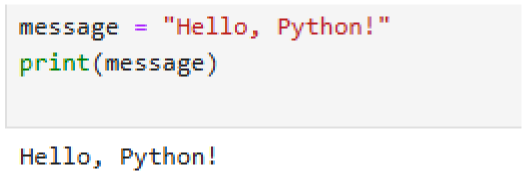
Program:



```
first_name = "Ali"
last_name = "Khan"
full_name = first_name + " " + last_name
print("Full name is:", full_name)
```

```
Full name is: Ali Khan
```

Program:

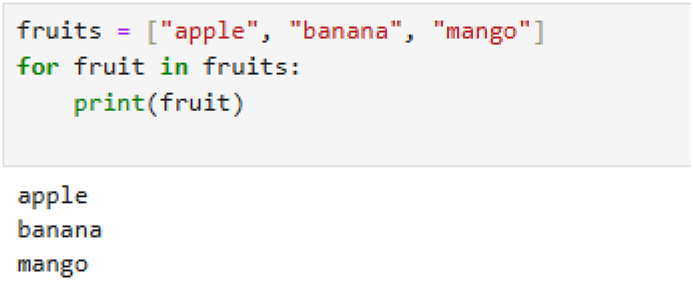


```
message = "Hello, Python!"
print(message)
```

```
Hello, Python!
```

- List:

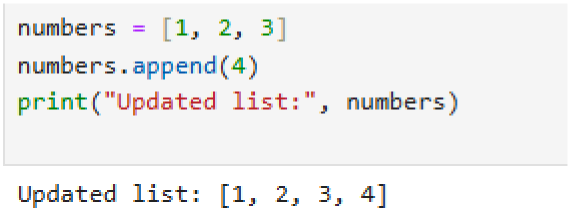
Program:



```
fruits = ["apple", "banana", "mango"]
for fruit in fruits:
    print(fruit)
```

```
apple
banana
mango
```

Program:

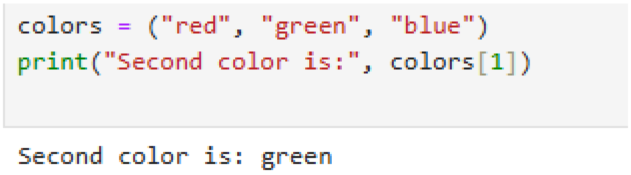


```
numbers = [1, 2, 3]
numbers.append(4)
print("Updated list:", numbers)
```

```
Updated list: [1, 2, 3, 4]
```

- Tuple:

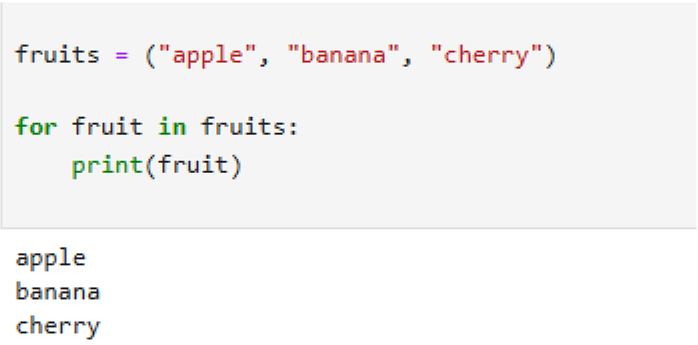
Program:



```
colors = ("red", "green", "blue")
print("Second color is:", colors[1])
```

```
Second color is: green
```

Program:

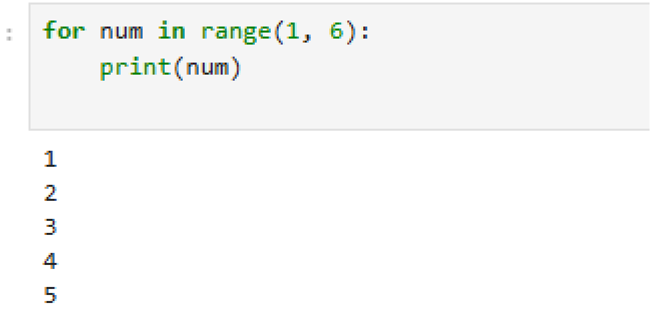


```
fruits = ("apple", "banana", "cherry")

for fruit in fruits:
    print(fruit)
```

```
apple
banana
cherry
```

- Range:

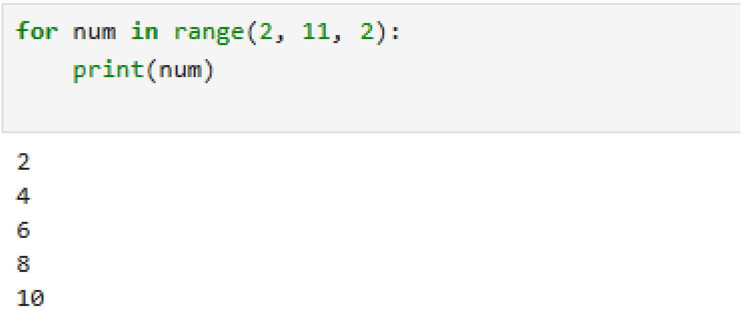
Program:



```
for num in range(1, 6):
    print(num)
```

```
1
2
3
4
5
```

Program:

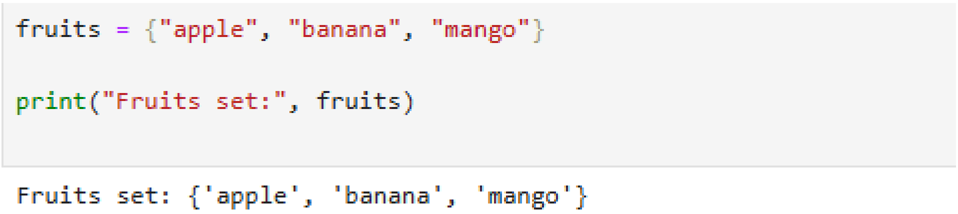


```
for num in range(2, 11, 2):
    print(num)
```

```
2
4
6
8
10
```

3- SET TYPES:

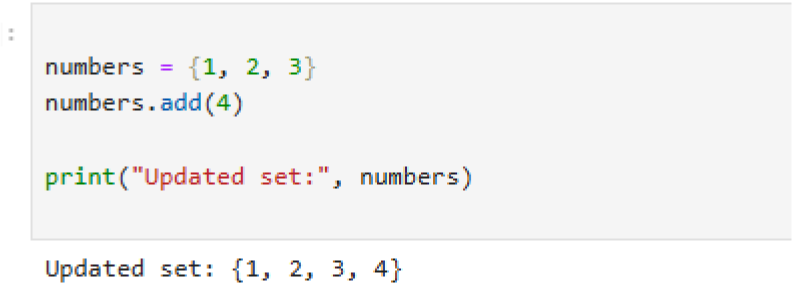
Program:



```
fruits = {"apple", "banana", "mango"}

print("Fruits set:", fruits)
```

```
Fruits set: {'apple', 'banana', 'mango'}
```

Program:

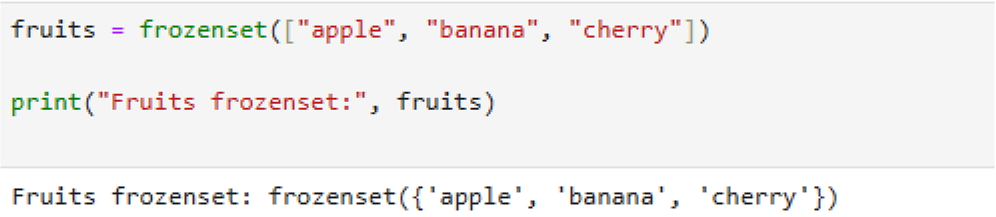


```
numbers = {1, 2, 3}
numbers.add(4)

print("Updated set:", numbers)
```

```
Updated set: {1, 2, 3, 4}
```

- **Frozen set:**

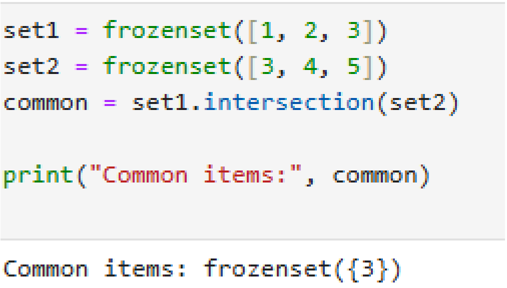
Program:



```
fruits = frozenset(["apple", "banana", "cherry"])

print("Fruits frozenset:", fruits)
```

```
Fruits frozenset: frozenset({'apple', 'banana', 'cherry'})
```

Program:



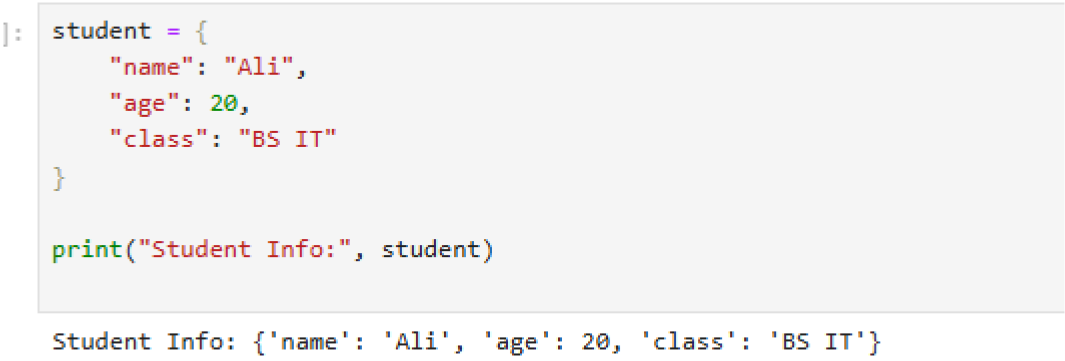
```
set1 = frozenset([1, 2, 3])
set2 = frozenset([3, 4, 5])
common = set1.intersection(set2)

print("Common items:", common)
```

```
Common items: frozenset({3})
```

MAPPING TYPE:

- Dictionary dict:

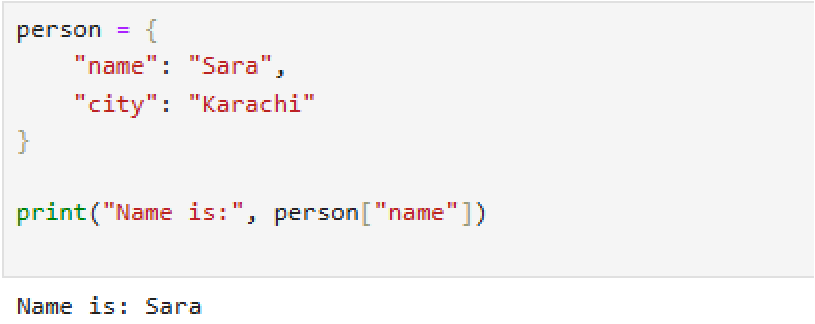
Program:



```
student = {
    "name": "Ali",
    "age": 20,
    "class": "BS IT"
}

print("Student Info:", student)
```

```
Student Info: {'name': 'Ali', 'age': 20, 'class': 'BS IT'}
```

Program:

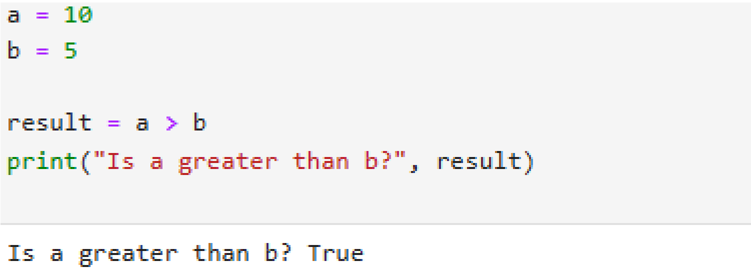


```
person = {
    "name": "Sara",
    "city": "Karachi"
}

print("Name is:", person["name"])
```

```
Name is: Sara
```

BOLEAN TYPE:

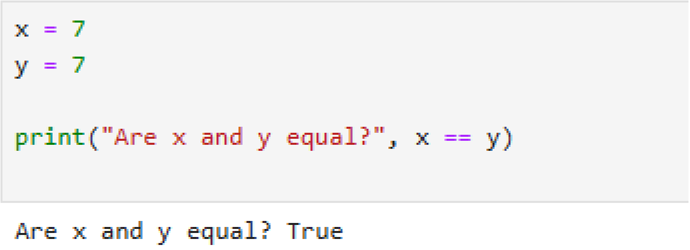
Program:



```
a = 10
b = 5

result = a > b
print("Is a greater than b?", result)
```

```
Is a greater than b? True
```

Program:



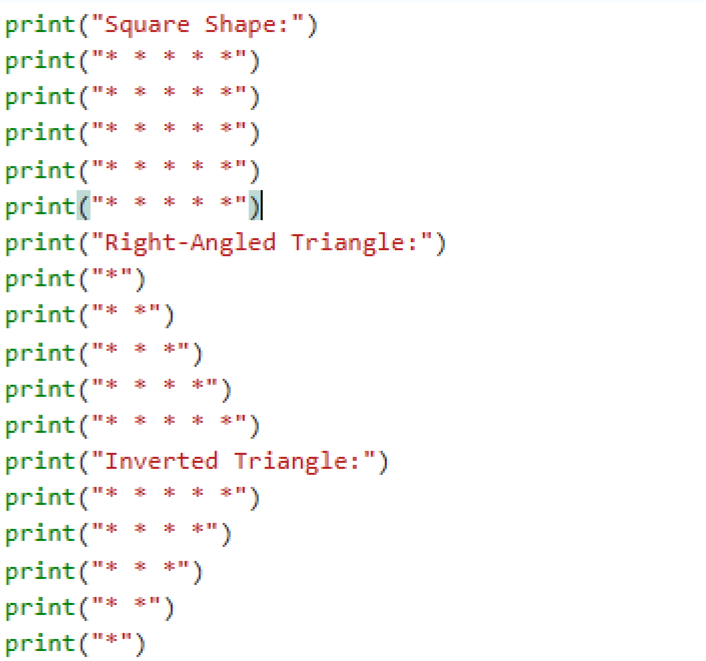
```
x = 7
y = 7

print("Are x and y equal?", x == y)
```

```
Are x and y equal? True
```

TASK NO 02:

Q.2 Make up to 5 shapes programs using *

Programs:



```
print("Square Shape:")
print("* * * * *")
print("* * * * *")
print("* * * * *")
print("* * * * *")
print("* * * * *")
print("Right-Angled Triangle:")
print("*")
print("* *")
print("* * *")
print("* * * *")
print("* * * * *")
print("Inverted Triangle:")
print("* * * * *")
print("* * * *")
print("* * *")
print("* *")
print("*")
```

```
Square Shape:
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
Right-Angled Triangle:
*
* *
* * *
* * * *
* * * * *
Inverted Triangle:
* * * * *
* * * *
* * *
* *
*
```

```python
print("Pyramid Shape:")
print("    *")
print("   * *")
print("  * * *")
print(" * * * *")
print("* * * * *")
print("Diamond Shape:")
print("    *")
print("   * *")
print("  * * *")
print("   * *")
print("    *")
```

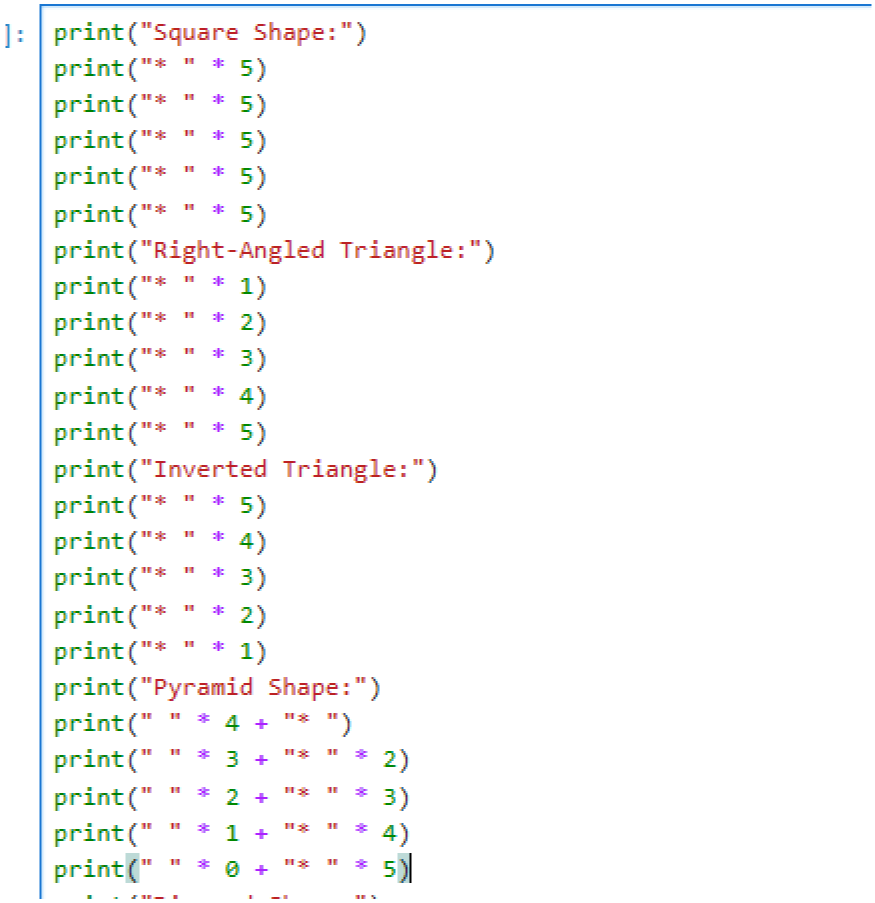
```
Pyramid Shape:
    *
   * *
  * * *
 * * * *
* * * * *
Diamond Shape:
    *
   * *
  * * *
   * *
    *
```

```
]:
```

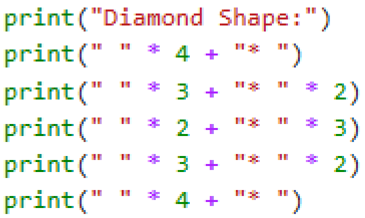
TASK 03

Q.3 Make same shapes you have made in task 2,using * multiple by number.

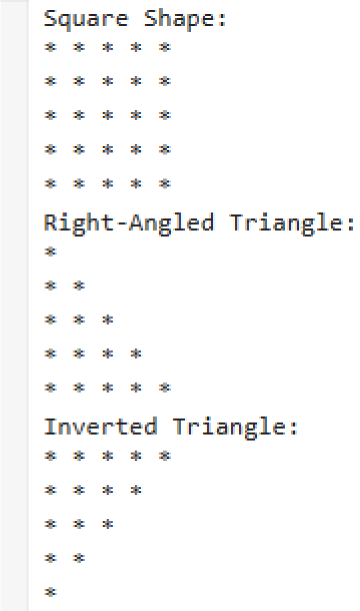
Program:



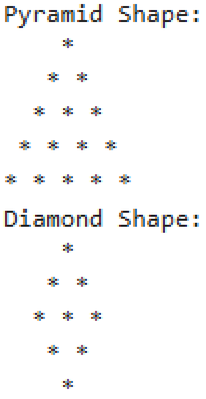
```
]: print("Square Shape:")
   print("* " * 5)
   print("* " * 5)
   print("* " * 5)
   print("* " * 5)
   print("* " * 5)
   print("Right-Angled Triangle:")
   print("* " * 1)
   print("* " * 2)
   print("* " * 3)
   print("* " * 4)
   print("* " * 5)
   print("Inverted Triangle:")
   print("* " * 5)
   print("* " * 4)
   print("* " * 3)
   print("* " * 2)
   print("* " * 1)
   print("Pyramid Shape:")
   print(" " * 4 + "* ")
   print(" " * 3 + "* " * 2)
   print(" " * 2 + "* " * 3)
   print(" " * 1 + "* " * 4)
   print(" " * 0 + "* " * 5)
```



```
print("Diamond Shape:")
print(" " * 4 + "* ")
print(" " * 3 + "* " * 2)
print(" " * 2 + "* " * 3)
print(" " * 3 + "* " * 2)
print(" " * 4 + "* ")
```



```
Square Shape:
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
Right-Angled Triangle:
*
* *
* * *
* * * *
* * * * *
Inverted Triangle:
* * * * *
* * * *
* * *
* *
*
```



```
Pyramid Shape:
    *
   * *
  * * *
 * * * *
* * * * *
Diamond Shape:
    *
   * *
  * * *
   * *
    *
```