EXERCISE 4

1.Write a program to produce the following output using for loop

In [4]:
```
print("  +----+")
for i in range(6):
    if i % 2 == 0:
        print("  \\    /")
    else:
        print("  /    \\")

print("  +----+")
```

```
  +----+
  \    /
  /    \
  \    /
  /    \
  \    /
  /    \
  +----+
```

2. Write a program to produce the following output using the loop

In [5]:
```
for i in range(5):
    print("**********")
```

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

3. Complete the code for the following for loop

In [6]:
```
for i in range(1, 7):
    a = 2 * i - 1
    b = 2 * i
    c = a * a - 3
    d = 10 - b
    e = 3 * i + 1
    f = 93 + i
    g = 12 * i - 18

    print(f"a) {a}\tb) {b}\tc) {c}\td) {d}\te) {e}\tf) {f}\tg) {g}")

    for j in range(1, 7):
        print(f"{j * i}\t", end="")
    print("\n")
```

```
a) 1    b) 2    c) -2   d) 8    e) 4    f) 94   g) -6
1       2       3       4       5       6

a) 3    b) 4    c) 6    d) 6    e) 7    f) 95   g) 6
2       4       6       8       10      12

a) 5    b) 6    c) 22   d) 4    e) 10   f) 96   g) 18
3       6       9       12      15      18

a) 7    b) 8    c) 46   d) 2    e) 13   f) 97   g) 30
4       8       12      16      20      24

a) 9    b) 10   c) 78   d) 0    e) 16   f) 98   g) 42
5       10      15      20      25      30

a) 11   b) 12   c) 118  d) -2   e) 19   f) 99   g) 54
6       12      18      24      30      36
```

4. Write a progeram to produce the following output using for loops. then use a class constant to make it possible to change the number of lines in the figure.

1 22 333 4444 55555 666666 7777777

In [7]:
```
for i in range(1, 8):
    for j in range(i):
        print(i, end="")
    print()
```

```
1
22
333
4444
55555
666666
7777777
```

5.Write a method named pay that accepts two parameters: a real number for a TA's salary, and an integer for the number of hours the TA worked this week. The method should return how much money to pay the TA. For example, the call pay(5.50, 6) should return 33.0.

The TA should receive "overtime" pay of 1 ½ normal salary for any hours above 8. For example, the call pay(4.00, 11) should return (4.00 * 8) + (6.00 * 3) or 50.0

In [11]:
```
def pay(salary, hours_worked):
    regular_hours = 8
    overtime_multiplier = 1.5

    if hours_worked <= regular_hours:
        total_pay = salary * hours_worked
    else:
        regular_pay = salary * regular_hours
        overtime_pay = salary * overtime_multiplier * (hours_worked - regular_h
        total_pay = regular_pay + overtime_pay

    return total_pay

# Testing the function
pay(5.50, 6)
```

Out[11]: 33.0

In [12]:
```
pay(4.00, 11)
```

Out[12]: 50.0

6.Consider the following method for converting milliseconds into days:

// converts milliseconds to days def toDays(millis):
return millis / 1000.0 / 60.0 / 60.0 / 24.0 Write a similar method named area that takes as a parameter the radius of a circle and that returns the area of the circle. For example, the call area(2.0); should return
12.566370614359172.
Recall that area can be computed as π times the radius squared and that Python has a constant called math.pi

In [13]:
```
import math

def area(R):
    #R means radius of the circle
    return math.pi * R**2


# Example
result = area(2.0)
print(result)
```

```
12.566370614359172
```

7. Copy and paste the following code into pythons IDLE script environment.

```
        low = 1          high = 1001          sum = 0          for i in r
    ange(low,high):
            sum += i

        print("sum = " , sum)
```

Modify the code to use a input to prompt the user for the values of low and high. Below is a sample execution in which the user asks for the same values as in the original program (1 through 1000): low? 1 high? 1001 sum = 500500 Below is an execution with different values for low and high: low? 300 high? 5298 sum = 13986903 You should exactly reproduce this format

In [17]:
```
low = int(input("Please input the low? "))
high = int(input("Please input the  high? "))
sum_result = 0

for i in range(low, high):
    sum_result += i

print("sum =", sum_result)
```

```
Please input the low? 1
Please input the  high? 1001
sum = 500500
```

8. Write a program using while loop thta prompts the user for numbers until the user types 0 then output their sum

In [18]:
```
# starting sum equal to 0
sum_result = 0

# Continue prompting for numbers until the user types 0
while True:
    # Take user input
    num = float(input("Enter a number (type 0 to exit): "))

    # Check if the user wants to exit
    if num == 0:
        break

    # Add the entered number to the sum
    sum_result += num

# Output the sum
print("Sum of the numbers:", sum_result)
```

```
Enter a number (type 0 to exit): 6
Enter a number (type 0 to exit): 3
Enter a number (type 0 to exit): 3
Enter a number (type 0 to exit): 32
Enter a number (type 0 to exit): 2
Enter a number (type 0 to exit): 0
Sum of the numbers: 46.0
```

9. Write a program using while loop that prompts the user for numbers until the user types -1, then outputs their sum.

In [20]:

```
sum_result = 0

# Continue prompting for numbers until the user types 0
while True:
    # Take user input
    num = float(input("Enter a number (type -1 to exit): "))

    # Check if the user wants to exit
    if num == -1:
        break

    # Add the entered number to the sum
    sum_result += num

# Output the sum
print("Sum of the numbers:", sum_result)
```

```
Enter a number (type -1 to exit): 3
Enter a number (type -1 to exit): 3
Enter a number (type -1 to exit): 3
Enter a number (type -1 to exit): 0
Enter a number (type -1 to exit): -1
Sum of the numbers: 9.0
```

10.

In [11]:
```
def repl(string, repetitions):
    if repl <= 0:
        return ""
    else:
        return string * repl

repl("hello",4)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-11-8f882b82c85a> in <module>
      5         return string * repl
      6
----> 7 repl("hello",4)
      8

<ipython-input-11-8f882b82c85a> in repl(string, repetitions)
      1 def repl(string, repetitions):
----> 2     if repl <= 0:
      3         return ""
      4     else:
      5         return string * repl

TypeError: '<=' not supported between instances of 'function' and 'int'
```

11

In [26]:
```
def printRange(start, end):
    if start < end:
        # Print increasing sequence
        for num in range(start, end + 1):
            print(num, end=" ")
    elif start > end:
        # Print decreasing sequence
        for num in range(start, end - 1, -1):
            print(num, end=" ")
    else:
        # Numbers are the same
        print(start)


```

In [31]:
```
printRange(5,5)
```

```
5
```

In [33]:
```
(printRange(1,8))
```

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

12. Write a method named smallestLargest that asks the user to enter numbers, then prints the smallest and largest of all the numbers typed in by the user. You may assume the user enters a valid number greater than 0 for the number of numbers to read. Here is an example dialogue:

In [42]:
```
def smallestLargest():
    num_count = int(input("How many numbers do you want to enter? "))

    # Initialize variables for smallest and largest
    smallest = float('inf')
    largest = float('-inf')

    for i in range(1, num_count + 1):
        num = float(input("Number {}: ".format(i)))

        # select the smallest and largest
        smallest = min(smallest, num)
        largest = max(largest, num)

    print("Smallest =", smallest)
    print("Largest =", largest)

# Call the method
smallestLargest()
```

```
How many numbers do you want to enter? 5
Number 1: 2
Number 2: 6
Number 3: 8
Number 4: 9
Number 5: 4
Smallest = 2.0
Largest = 9.0
```

13. Write a method called printAverage that uses a sentinel loop to repeatedly prompt the user for numbers. Once the user types any number less than zero, the method should display the average of all nonnegative numbers typed. Display the average as a double. Here is a sample dialogue with the user:

In [40]:
```
def printAverage():
    total = 0
    count = 0

    while True:
        num = float(input("Type a number: "))

        if num < 0:
            if count > 0:
                average = total / count
                print("Average was", average)
            else:
                print("No non_negative numbers were entered.")
            break

        total += num
        count += 1

# Call the method
printAverage()
```

```
Type a number: 67
Type a number: 3
Type a number: 5
Type a number: -5
Average was 25.0
```

In [41]:
```
printAverage()
```

```
Type a number: -2
No nonnegative numbers were entered.
```

14. Write a method named numUnique that takes three integers as parameters and returns the number of unique integers among the three. For example, the call numUnique(18, 3, 4) should return 3 because the parameters have three different values. By contrast, the call numUnique(6, 7, 6) should return 2 because there are only two unique numbers among the three parameters: 6 and 7.

In [39]:
```
def numUnique(num1, num2, num3):
    unique_numbers = {num1, num2, num3}
    return len(unique_numbers)

numUnique(20,20,4)
```

Out[39]: 2

In [38]:
```
numUnique(20,3,4)
```

Out[38]: 3

15. A Random object generates pseudo-random numbers. Find out how to use the Random class and write a program that simulates rolling of two 6sided dice until their combined result comes up as 7. One possible output can be seen as below:

In [36]:
```
import random

def roll_dice():
    return random.randint(1, 6)

def simulate_dice_rolls():
    tries = 0

    while True:
        dice1 = roll_dice()
        dice2 = roll_dice()
        total = dice1 + dice2

        print("\t" if tries < 10 else "", dice1, "+", dice2, "=", total)

        tries += 1

        if total == 7:
            print("\tYou won after", tries, "tries!")
            break

# Simulate dice rolls
simulate_dice_rolls()
```

```
         6 + 3 = 9
         5 + 6 = 11
         2 + 2 = 4
         5 + 5 = 10
         2 + 1 = 3
         4 + 1 = 5
         5 + 3 = 8
         4 + 5 = 9
         4 + 6 = 10
         6 + 3 = 9
 6 + 5 = 11
 6 + 6 = 12
 4 + 5 = 9
 5 + 3 = 8
 1 + 1 = 2
 2 + 3 = 5
 4 + 6 = 10
 4 + 6 = 10
 5 + 3 = 8
 2 + 5 = 7
       You won after 20 tries!
```