Print Function

print() function comes with a parameter called 'end'. By default, the value of this parameter is '\n', i.e. the new line character.

In [2]:
```
for i in range (3):
    print(i)
```

```
0
1
2
```

In [3]:
```
for i in range(3):
    print(i, end='   ')
```

```
0   1   2
```

In [4]:
```
for i in range(3):
    print(i, end='*')
```

```
0*1*2*
```

In [5]:
```
for i in range(3):
    print(i, end='')
```

```
012
```

Homework 3: print triangle with *

In [ ]:
```
Using nested loops
```

In [64]:
```
max_star = int(input("please input maximum number of stars in a row:"))
for row in range(1,  max_star+1):
    #print from one start a row to max_star in a row
    for num_star in range(0,  row):
        print('*', end='')

    # outside the for loop, start a new line
    print()

for row in range(max_star-1, 0, -1):
    # print * from max_star-1 in a row to one star in a row
    for num_star in range(0, row):
        print('*', end='')

    print()
```

```
please input maximum number of stars in a row:5
*
**
***
****
*****
****
***
**
*
```

In [ ]:
```
Using string multiplication
```

In [65]:
```
max_star = int(input("please input maximum number of stars in a row:"))
for row in range(1,  max_star+1):
    print('*'*row)

for row in range(max_star-1, 0, -1):
    print('*'*row)
```

```
please input maximum number of stars in a row:5
*
**
***
****
*****
****
***
**
*
```

In [ ]:
```
## Print a Diamond Shape of starts of odd number of input
```

In [70]:
```
while True:
    width = int(input("please input maximum number of stars in a row:"))
    if width%2 != 0:
        break
    print('Please input an odd number.')

spaces = width//2
for row in range(1,  width+1, 2):
    print(' '*spaces, end='')
    print('*'*row)
    spaces -=1

# the bottom half of diamond, do not print middle row again (starts from widt
spaces = 1
for row in range(width-2, 0, -2):
    print(' '*(spaces), end='')
    print('*'*row)
    spaces +=1
```

```
please input maximum number of stars in a row:8
Please input an odd number.
please input maximum number of stars in a row:7
   *
  ***
 *****
*******
 *****
  ***
   *
```

Print the diamond in the middle of a row

In [68]:
```
# defind an offset to print the diamond in the middle of the row
offset = 10
print('Offset is ', offset)

while True:
    width = int(input("please input maximum number of stars in a row:"))
    if width%2 != 0:
        break
    print('Please input an odd number.')

spaces = width//2
for row in range(1,  width+1, 2):
    print(' '*(offset+spaces), end='')
    print('*'*row)
    spaces -=1

# the bottom half of diamond, do not print middle row again (starts from widt
spaces = 1
for row in range(width-2, 0, -2):
    print(' '*(offset+spaces), end='')
    print('*'*row)
    spaces +=1
```

```
Offset is  10
please input maximum number of stars in a row:7
            *
           ***
          *****
         *******
          *****
           ***
            *
```

Swap the value of two variables

In [92]:
```
a = 2
b = 5
print('before swapping ', a, b)
a, b = b, a
print('after swapping ', a, b)
```

```
before swapping  2 5
after swapping  5 2
```

Without using Python's swap assignment directly

In [93]:
```
a = 2
b = 5
print('before swapping ', a, b)
c = a
a = b
b = c
print('after swapping ', a, b)
```

```
before swapping  2 5
after swapping  5 2
```

Homework: Print odd numbers

Print all of the odd numbers between 80 to 101. Print as the following format:

each odd number in seperate line, with a sequence number in front of it. For example:

1) 81

2) 83

3) 85

In [24]:
```
i=1
for even in range(81, 101, 2):
    print("%d) %d" % (i, even))
    i=i+1
```

```
1) 81
2) 83
3) 85
4) 87
5) 89
6) 91
7) 93
8) 95
9) 97
10) 99
```

Homework check if a given year is leap year or not

Using nested conditions

A leap year is exactly divisible by 4 except for century years (years ending with 00). The century year is a leap year only if it is perfectly divisible by 400. For example,

In [ ]:
```
year = int(input("Enter a year: "))
# Write your program below to check if the value of the variable 'year' is le

if (year % 4) == 0:
   if (year % 100) == 0:
       if (year % 400) == 0:
           print("{0} is a leap year".format(year))
           # or you can print in the following way
           # print("%d is a leap year" % year)
       else:
           print("{0} is not a leap year".format(year))
   else:
       print("{0} is a leap year".format(year))
else:
   print("{0} is not a leap year".format(year))
```

Median Number

Write a Python program to find the median of three numbers from the user input.

In [52]:
```
a = int(input("Input first number: "))
b = int(input("Input second number: "))
c = int(input("Input third number: "))
if a > b:
    if a < c:
        median = a
    elif b > c:
        median = b
    else:
        median = c
else:
    if a > c:
        median = a
    elif b < c:
        median = b
    else:
        median = c

print("The median is ", median)
```

```
Input first number: 10
Input second number: 20
Input third number: 15
The median is  15
```

Compute the greatest common divisor (GCD) of two positive integers

In [84]:
```
a = int(input("Input first integer: "))
b = int(input("Input second integer: "))

if a > b:
    divisor = b
else:
    divisor = a

for gcd in range(divisor, 0, -1):

    if a%gcd == 0 and b%gcd == 0:
        print('The GCD of %d and %d is %d' % (a, b, gcd))
        break
```

```
Input first integer: 16
Input second integer: 16
The GCD of 16 and 16 is 16
```

Get the least common multiple (LCM) of two positive integers

In [87]:
```
a = int(input("Input first integer: "))
b = int(input("Input second integer: "))

if a > b:
    multiple = a
else:
    multiple = b

while multiple%a!=0 or multiple%b!=0:
    multiple = multiple + 1

print('The least common multiple of %d and %d is %d' % (a, b, multiple))
```

```
Input first integer: 12
Input second integer: 24
The least common multiple of 12 and 24 is 24
```

Use while True

In [89]:
```
a = int(input("Input first integer: "))
b = int(input("Input second integer: "))

if a > b:
    multiple = a
else:
    multiple = b

while True:
    if multiple%a==0 and multiple%b==0:
        print('The least common multiple of %d and %d is %d' % (a, b, multipl
        break

    multiple = multiple + 1
```

```
Input first integer: 8
Input second integer: 12
The least common multiple of 8 and 12 is 24
```

Find Required Numbers

Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 2200 (both included). The numbers obtained should be printed in a comma-separated sequence on a single line.

In [91]:
```
l=[]
for i in range(2000, 2201):
    if (i%7==0) and (i%5!=0):
        print(i, end=', ')
```

```
2002, 2009, 2016, 2023, 2037, 2044, 2051, 2058, 2072, 2079, 2086, 2093, 210
7, 2114, 2121, 2128, 2142, 2149, 2156, 2163, 2177, 2184, 2191, 2198,
```

Check if a user input number is a Prime Number

In [12]:
```
# take input from the user
num = int(input("Enter a number: "))
# prime numbers are greater than 1
if num > 1:
    # check for factors
    for i in range(2,int(num/2)):
        if (num % i) == 0:
            print(num,"is not a prime number")
            print(i,"times",int(num/i),"is",num)
            break
    else:
        print(num,"is a prime number")

# if input number is less than or equal to 1, it is not prime
else:
   print(num,"is not a prime number")
```

```
Enter a number: -7
-7 is not a prime number
```

without using 'for loop else'

In [21]:
```
# take input from the user
num = int(input("Enter a number: "))
# prime numbers are greater than 1
if num > 1:
    # check for factors
    for i in range(2,int(num/2)):
        if (num % i) == 0:
            print(num,"is not a prime number")
            print(i,"times",int(num/i),"is",num)
            break
    print('After for loop, i=', i)
    if i+1 >= int(num/2):
        print(num,"is a prime number")

# if input number is less than or equal to 1, it is not prime
else:
   print(num,"is not a prime number")
```

```
Enter a number: 19
After for loop, i= 8
19 is a prime number
```

Find 10 prime numbers that are bigger than 12345

In [16]:
```
# setup a counter
count = 0
# initial number
num = 12345
while count < 10:
    # check for factors
    for i in range(2,int(num/2)):
        if (num % i) == 0:
            break
    else:
        print(count, ') ', num,"is a prime number")
        count = count + 1
    num = num +1
```

```
0 )  12347 is a prime number
1 )  12373 is a prime number
2 )  12377 is a prime number
3 )  12379 is a prime number
4 )  12391 is a prime number
5 )  12401 is a prime number
6 )  12409 is a prime number
7 )  12413 is a prime number
8 )  12421 is a prime number
9 )  12433 is a prime number
```

Use 'while True'

In [17]:
```
# setup a counter
count = 0
# initial number
num = 12345
while True:
    # check for factors
    for i in range(2,int(num/2)):
        if (num % i) == 0:
            break
    else:
        print(count, ') ', num,"is a prime number")
        count = count + 1
    if count == 10:
        break
    num = num +1
```

```
0 )  12347 is a prime number
1 )  12373 is a prime number
2 )  12377 is a prime number
3 )  12379 is a prime number
4 )  12391 is a prime number
5 )  12401 is a prime number
6 )  12409 is a prime number
7 )  12413 is a prime number
8 )  12421 is a prime number
9 )  12433 is a prime number
```

Chicken Story -1: 鸡兔同笼

Solving the famous ancient Chinese math problem called “Chickens and Rabbits in a Cage.”

A farmer put chickens and rabbits in a cage. There are totally 49 animals and 124 legs in the cage. How many rabbits are there in the cage?

Your output should be:

There are totally ?? rabbits in the cage!

In [28]:
```
total = 245
legs = 620

# Write your program at below

for numRabbit in range(total+1):
    numChicken = total - numRabbit
    if ((numRabbit*4 + numChicken*2) == legs):
        print("There are totally %d rabbits and %d chickens in the cage!" % (
        break
else:
    print('No sulution can be found')
```

```
There are totally 65 rabbits and 180 chickens in the cage!
```

Chicken Story -2: Only Chickens

```
A rooster cost 5 dollars
A hen cost 3 dollars
Three baby chikens cost 1 dollars
```

Now you need to spend exactly $100 to buy exactly 100 chickens, without any money left. How many rooster, hen and baby chikens should you buy? Note: There might be more than one solutions

In [51]:
```
# r for Rooster
# h for hen
# c for baby chikens
total_num = 100
total_cost = 100
for r in range(total_num):
    for h in range(total_num-r):
        c = total_num-r-h
        if c%3 == 0:
            if (5*r + h*3 + c/3) == total_cost:
                print("Roosters: %d,  Hens: %d,  Baby chickens: %d" % (r, h,
```

```
Roosters: 2,  Hens: 34,  Baby chickens: 114
Roosters: 6,  Hens: 27,  Baby chickens: 117
Roosters: 10,  Hens: 20,  Baby chickens: 120
Roosters: 14,  Hens: 13,  Baby chickens: 123
Roosters: 18,  Hens: 6,  Baby chickens: 126
```

The Smart Monkeys

A group of monkeies picked up some peaches on the first day. They ate half of them and gave another peach from the other half left to a baby monkey. On the second day, they ate half of what had left from previous day, and give another peach from the other half to a baby monkey. Every day they did the same, until on the 15th day, they found they only have 1 peach left.

Make a program, print out how many peaches those monkey had on each day, from day 15 to day 1.

For Example, your program should print out like the following:

```
day 15: 1
day 14: 4
day 13: 10
...
...
```

Note: make a program to do the calculation, do NOT do the calculation yourself.

In [37]:
```
n = 1
# Last day, print out directly
print ("day 15: 1 peach")

# loop from Day 29 to Day 1
for i in range(14,0,-1):
     n = (n+1)*2
     print("day %2d: %7d peaches" % (i, n))
```

```
day 15: 1 peach
day 14:       4 peaches
day 13:      10 peaches
day 12:      22 peaches
day 11:      46 peaches
day 10:      94 peaches
day  9:     190 peaches
day  8:     382 peaches
day  7:     766 peaches
day  6:    1534 peaches
day  5:    3070 peaches
day  4:    6142 peaches
day  3:   12286 peaches
day  2:   24574 peaches
day  1:   49150 peaches
```

Square Root

This problem is optional for students who do not know what is "square root".

There is a positive integer X. After adding 100 to x, we got y, which is equal to the square of an integer. Now we add 168 to y, again the result is equal to the square of another integer.

Make a program to find out the smallest possible value of X.

Note: make a program to do the calculation, do NOT do the calculation yourself.

Hint: to get the square root, use: math.sqrt( )

In [38]:
```
import math

num = 1
while True:
    sqrt1 = math.sqrt(num + 100)
    sqrt2 = int(math.sqrt(num + 268)) # Note: use int() to convert the square
    # the following line demoes two different ways to check if a number is ar
    if sqrt1-int(sqrt1) == 0 and sqrt2*sqrt2 == (num+268):
        print(num)
        break
    num += 1
```

```
21
```

In [ ]:
```

```