Question 1 : Write a program that asks the user to enter their age in years as input (assume that the user enters a positive integer) and calculates and prints how old the user is in terms of days. Assume that there are 365 days in a year.

For example: if the user enters 22

Sample Output → You are 8030 days old

In [3]:
```
age = eval(input('how old are you : '))
print(f'your days old is {age*365}')
```

```
how old are you : 17
your days old is 6205
```

Question 2: Write a Python program to display the current date and time

In [1]:
```
import datetime
now = datetime.datetime.now()
print('Current date and time:')
print(f'{now: %A, %d/%m/%Y, %H:%M:%S}')
```

```
Current date and time:
 Sunday, 01/01/2023, 12:23:33
```

Question 3 : Suppose the cover price of a book is $24.95, but bookstores get a 40% discount. Shipping costs $3 for the first copy and 75 cents for each additional copy. Write a program to find the total wholesale cost for 60 copies.

In [4]:
```
book = 24.95*0.6
copy = 60
total = book*copy + 3 + 0.75*(copy - 1)
print(total)
```

```
945.4499999999999
```

Question4: If I leave my house at 06:52:00 am and run 1 mile at an easy pace (00:08:15 per mile), then 3 miles at tempo (00:07:12 per mile) and 1 mile at easy pace again, what time do I get home for breakfast? Write a program to solve this problem. (3points)

In [8]:
```
start = 6*3600 + 52*60
easy = 2*(8*60 + 15)
tempo = 3*(7*60 + 12)
home = start + easy + tempo
hour = home//3600
minute = (home - hour*3600)//60
second = (home - hour*3600 - minute*60)
print(hour,':',minute,':',second)
```

```
7 : 30 : 6
```

Question 5: Modify the following program to display the sales tax with two digits after the decimal point.

```
# Input purchase amount
purchaseAmount = eval(input("Enter purchase amount: "))
# Compute sales tax
tax = purchaseAmount * 0.06
print("Sales tax is", tax)
```

In [11]:
```
1 purchaseamount = eval(input('Enter purchase amount: '))
2 tax = purchaseamount*0.06
3 print('sale tax is', (tax*100)/100)
4 print(f'sale tax is {tax: .2f}')
```

```
Enter purchase amount: 123
sale tax is 7.38
sale tax is  7.38
```

Qustion 6: Complete the code below

```
a = "+"
b = "-"
c = a + b
# Create the 45-character string below, store it in a variable called s
# "+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+"

# print s and the length of s
print(s,"\n",len(s))
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 45
```

In [13]:
```
1 a = '+'
2 b = '-'
3 c = a+b
4 s = c*22 + a
5 print(s, '\n',len(s))
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 45
```

Question 7: Write a Python program which accepts a list of comma-separated integer numbers from user and display length, sum of items in list

In [3]:
```
1 n = eval(input('Enter a list: '))
2 print(f'Length of list is {len(n)}')
3 print(f'sum is {sum(n)}')
```

```
Enter a list: [1,2,3,4,5]
Length of list is 5
sum is 15
```

Question 8: Compose a program that writes five uniform random floats between 0 and 1, their average value, and their minimum and maximum value.

In [13]:
```
import random
r = random.randrange(1,10)#sinh ra 1 số nguyên theo start và stop
print(r)
#Read the documentation for randrange() function and generate a random odd number less than 100

r2 = random.randrange(1,100,2)#randrange(start, stop=None, step=1) method of random.Random instance
print(r2)
r4 = random.randint(1,100)#sinh ra 1 số nguyên bao gồm cả a và b
print(r4)
r5 = random.uniform(0,1)
print(r5)

```

In [4]:
```
import random
r1 = random.uniform(0,1)
r2 = random.uniform(0,1)
r3 = random.uniform(0,1)
r4 = random.uniform(0,1)
r5 = random.uniform(0,1)
print(f'Average is {(r1+r2+r3+r4+r5)/5}')
print(f'Maximum is {max(r1,r2,r3,r4,r5)}')
print(f'Minimum is {min(r1,r2,r3,r4,r5)}')
```

```
Average is 0.3372426347875665
Maximum is 0.6448036476948044
Minimum is 0.07998392861758319
```

Question 9: Write a program that lets the user enter a year and then determines whether it is a leap year. All leap years are divisible by four, but multiples of 100 are not leap years unless they are also a multiple of 400.

***For example*:**
2016 was a leap year, but 1900 was not
2000 was a leap year, as it is a multiple of 400, but 2100 will not be a leap year.

In [6]:
```
n = eval(input('Enter a year: '))
if (n%4==0 and n%100 != 0 ) or (n%400 == 0):
    print(f'{n} is a leap year')
else:
    print(f'{n} is not a leap year')
```

```
Enter a year: 2022
2022 is not a leap year
```

Question 10: Given different scored marks of students. We need to find grades. The final grade will be out of 100%. The grading breakdown is listed below

Weekly Homework:	20%
Quizzes:	10%
Assignments :	20%
Final Exam:	50%

Grade will be calculated according to:

1. score >= 9.0 : "A+"
2. 8.5 <= score < 9.0 : "A"
3. 8.0 <= score < 8.5 : "B+"
4. 7.0 <= score < 8.0 : "B"
5. 6.5 <= score < 7.0 : "C+"
6. 6.0 <= score < 6.5 : "C"
7. 5.0 <= score < 6.0 : "D+"
8. 4.0 <= score < 5.0 : "D"

In [5]:

```
a = eval(input('Enter Weekly Homework: '))
b = eval(input('Enter Quizzes: '))
c = eval(input('Enter Assignments: '))
d = eval(input('Enter Final Exam: '))
score = a*0.2 + b*0.1 + c*0.2 + d*0.5
if a<0 or b<0 or c<0 or d<0:
    print('invalid data')
if score>=9.0:
    print('A+')
elif score>= 8.5:
    print('A')
elif score >= 8.0:
    print('B+')
elif score>=7.0:
    print('B')
elif score>=6.5:
    print('C+')
elif score>=6.0:
    print('C')
elif score>=5.0:
    print('D+')
else:
    print('D')
```

```
Enter Weekly Homework: 1
Enter Quizzes: 2
Enter Assignments: 3
Enter Final Exam: 4
D
```

Question 11: Develop a program to play a lottery. The program randomly generates a two-digit number, prompts the user to enter a two-digit number, and determines whether the user wins according to the following rules:

a. If the user's input matches the lottery in the exact order, the award is $10,000.

b. If all the digits in the user's input match all the digits in the lottery number the award is $3,000

c. If one digit in the user's input matches a digit in the lottery number, the award is $1,000.

In [9]:
```
import random
lottery = random.randrange(10,99)
guess = eval(input('Enter the number: '))
lottery1 = lottery//10
lottery2 = lottery%10
guess1 = guess//10
guess2 = guess%10
if lottery == guess:
    print('you have $10,000')
elif lottery1 == guess2 and lottery2 == guess1:
    print('You have $3,000')
elif lottery1 == guess1 or lottery1==guess2\
or lottery2==guess1 or lottery2==guess2:
    print('You have $1000')
else:
    print('Try again')
```

```
Enter the number: 27
You have $1000
```

Question 12: Write a program that lets user enter a integer number N and calculate the sum of EVEN numbers from 0 to N-1.

In [12]:
```
n = eval(input('Enter N: '))
s= 0
for i in range(0,n,2):
    s= s+i
print(s)
```

```
Enter N: 12
30
```

Question 13: Create a simple guessing game. Program generates a random integer number between 1 and 50. The program prompts the user to enter numbers continuously until it matches the randomly generated number. For each user input, the program reports whether it is too low or too high, so the user can choose the next input intelligently

```
import random
n = random.randint(1,50)
while True:
    a = eval(input("Enter number: "))
    if a>n:
        print('too high')
    elif a<n:
        print('too low')
    else:
        print('Congatulation')
        break
```

Question 14: Find all divisors of a natural number.

In [7]:
```
n = eval(input('Enter n: '))
print(f'all divisor {n} are')
for i in range(1, n +1 ):
    if n%i == 0:
        print(i, end= ' ')
```

```
Enter n: 12
all divisor 12 are
1 2 3 4 6 12
```

Question 15: Write a program to find out the Chinese zodiac sign for a given year. The Chinese zodiac sign is based on a 12-year cycle, and each year in this cycle is represented by an animal — monkey, rooster, dog, pig, rat, ox, tiger, rabbit, dragon, snake, horse, and sheep

In [4]:
```
year = eval(input('The year is: '))
if year%12 == 0:
    print('Monkey')
elif year%12 ==1:
    print('rooster')
elif year%12 == 2:
    print('Dog')
elif year%12 == 3:
    print('Pig')
elif year%12 == 4:
    print('Rat')
elif year%12 ==5:
    print('Ox')
elif year%12 == 6:
    print('Tiger')
elif year%12 == 7:
    print('Rabbit')
elif year%12 == 8:
    print('Dragon')
elif year%12 == 9:
    print('Snake')
elif year%12 == 10:
    print('Horse')
else:
    print('Sheep')
```

```
The year is: 1966
Horse
```

Question 16: Write a program to display all odd numbers between a and b.

In [8]:
```
while True:
    a = eval(input('Enter a:'))
    b = eval(input('Enter b: '))
    if a>=b:
        print('a must be greater than b')
    else:
        break
for i in range(a+1,b):
    if i%2 !=0:
        print(i, end = ' ')
```

```
Enter a:1
Enter b: 20
3 5 7 9 11 13 15 17 19
```

Question 17: Write a program to create a simple calculator that can add, subtract, multiply or divide depending upon the input from the user.

```
# Print menu
q = 'Y'
while q == 'Y':
    c = int(input("Please choose operations (1,2,3,4):"))
    # check user's choice is valid or invalid?
    # if choice is invalid, ask user to re-enter other choice
    # if choice is valid, prompt user to input two numbers
    # Use "if" statement to execute operations
    q = input("Do you want to continue (Y/N)?")
```

Sample output:

```
1.Add
2.Subtract
3.Multiply
4.Divide
Please choose operations (1,2,3,4):5
Your choice is invalid
Please choose operations (1,2,3,4):3
Enter two numbers: 2,3
2 * 3 = 6
Do you want to continue (Y/N)?Y
Please choose operations (1,2,3,4):1
Enter two numbers: 3,4
3 + 4 = 7
Do you want to continue (Y/N)?N
```

In [1]:

```
print(f'1.Add', '2.Subtract', '3.Multiply', '4.Divide', sep = '\n')
q = 'Y'
while q =='Y':
    c = int(input(('Please choose operations (1,2,3,4): ')))
    if c<1 or c>4:
        print('invalid data, please choose again')
        continue
    a,b = eval(input('Enter a and b: '))
    if c == 1:
        print(f'{a}+{b} = {a+b}')
    elif c == 2:
        print(f'{a}-{b} = {a - b}')
    elif c == 3:
        print(f'{a}*{b} = {a*b}')
    else:
        if b==0:
            print('zero division error')
        else:
            print(f'{a}/{b} = {a/b}')
    q = input('Do you want to continue (Y/N)?')
```

Question 18: Write a compare function that returns 1 if a > b, 0 if a == b, and -1 if a < b.

In [3]:
```
def compare(a,b):
    if a>b:
        return 1
    elif a==b:
        return 0
    else:
        return -1
a= int(input('Enter a: '))
b = int(input('Enter a: '))
compare(a,b)
```

```
Enter a: 1
Enter a: 2
```

Question 19: Write a program that contains two functions:

a. **Calculate area of rectangle (hình chữ nhật)**
b. **Calculate area of circle**

In [8]:
```
def rectangle(a,b):
    if a<0 or b<0:
        return -1
    else:
        area = a*b
        return area

import math
def circle(c):
    if c<0:
        return -1
    else:
        area = c*c*math.pi
        return area
a = int(input('Enter length: '))
b = int(input('Enter wildth: '))
print(f'area of reactangle is {rectangle(a,b)}')

c = eval(input('Enter radius: '))
print(f'area of circle is {circle(c): .3f}')
```

```
Enter length: 2
Enter wildth: 3
area of reactangle is 6
Enter radius: 4
area of circle is  50.265
```

Question 20: Write a function to reverse digits of a number.

In [12]:
```
def reversenum(a):
    rev = 0
    while a!=0:
        i = a%10
        a = a//10
        rev = (rev*10)+i
    return rev
a = int(input('Enter a: '))
print(f'reverse digits of number ís {reversenum(a)}')

```

```
Enter a: 123456789
reverse digits of number ís 987654321
```

Question 21: Write a function to show greeting based on current time:

In [15]:
```
def currenttime(n):
    if n<0 or n>=24:
        print('Invalid data')
    elif n<12:
        print('Good morning')
    elif n<16:
        print('Good afternoon')
    elif n<21:
        print('Good evening')
    else:
        print('Good night')

import time
n = time.localtime()
currenttime(n.tm_hour)
```

```
Good night
```

String Indexing

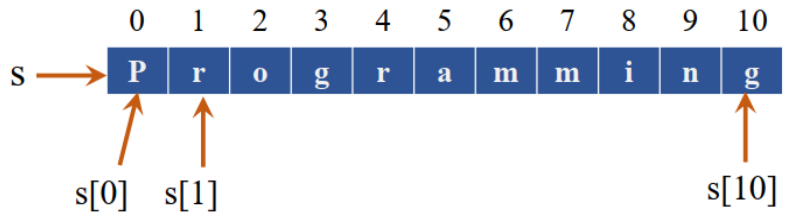
Phần này lưu ý đọc kĩ , ghi nhớ sâu

chức năng truy xuất kí tự

- A character in the string can be accessed through the index operator using the syntax:

  s[<index>]

- For example: s = “Programming”



- The indexes are 0 based:
  indexes range from 0 to len(s) – 1

Thực chất việc từ 0 → len(s) -1 là do vị trí của str được tính từ 0 trở đi còn hàm len là bắt đầu đếm từ 1 nên mới có việc len(s)-1

```
1  s = "Programming"
2  print(s[0])
3  print(s[1])
4  print(len(s))
5  print(s[len(s)-1])
6  #print(s[len(s)]) --> IndexError: string index out of range
```

```
P
r
11
g
```

A+ A-

- Python also allows the use of negative numbers as indexes → indexing occurs from the end of the string backward:

-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
P	**r**	**o**	**g**	**r**	**a**	**m**	**m**	**i**	**n**	**g**
0	1	2	3	4	5	6	7	8	9	10

```
1  s = "Programming"
2  print(s[-len(s)])
3  print(s[-1])
```

```
P
g
```

String Slicing

- The slicing operator returns a slice of the string using the syntax:

s[start : end : step] cú pháp giống với hàm range

- The slice is a substring from index **start** to index **end – 1.**
- The *starting index or ending index may be omitted*. In this case, by default the starting index is 0 and the ending index is the last index

```
1  s = "Programming"
2  print(s[1:4])
3  print(s[:4])
4  print(s[4:])
5  print(s[1:-1])
6  print(s[3:1]) # the slice is empty
```

```
rog
Prog
ramming
rogrammin
```


Iterating a String

duyệt chuỗi

- A string is iterable → can use a for loop to traverse all characters in the string sequentially.

```
1  s = "Programming"
2  for ch in s: duyệt chuỗi
3      print(ch,end='-')
```

```
P-r-o-g-r-a-m-m-i-n-g-
```

```
1  # displays characters at odd-numbered positions
2  s = "Programming"
3  for i in range(1,len(s),2):
4      print(s[i],end='-')
```

duyệt vị trí của chuỗi

```
r-g-a-m-n-
```

String Functions

Case Conversion

- **s.lower()**: return a copy of string in lowercase
- **s.upper()**: return a copy of string in uppercases
- **s.capitalize()**: returns a copy of string with the first character converted to uppercase. in hoa chữ cái đầu
- **s.title()**: returns a copy of this string with the first letter capitalized in each word. viết hoa chữ đầu
- **s.swapcase()**: Returns a copy of this string in which lowercase letters are converted to uppercase and uppercase to lowercase chuyển chữ hoa thành thường và thường thành hoa

c

```
1  s1 = "Welcome to Programming"
2  print(s1.lower())
3  print(s1.upper())
4  print(s1.swapcase())
5
```

```
welcome to programming
WELCOME TO PROGRAMMING
wELCOME TO pROGRAMMING
```


Stripping characters from a String

- **s.strip([<char>])**: removes any character from the beginning or the end loại bỏ bất kì kí tự nào nếu muốn
  - By default, char is **None** --> ***whitespace characters are removed***

```
1  s = " Welcome to Python \n"
2  s.strip()
```

```
'Welcome to Python'
```

```
1  s = "***10***"
2  s.strip('*')
```

```
'10'
```

Find and Replace

trả ra vị trí nhỏ nhất hoặc đầu tiên nếu chữ đó lặp nhiều lần

- **s.find(sub[,start[,end]])**: returns the lowest index in s where substring is found, otherwise return -1 trả về kí tự gần nhất
- **s.replace(old, new)**: Returns a new string that replaces all the occurrences of the old string with a new string

thay thế chữ nếu như chữ đó có trong chuỗi còn nếu không có thì nó sẽ trả lại về chuỗi ban đầu không thay đổi

```
print(s.find("come"))
print(s.find("become"))
```

```
3
-1
```

```
s1 = s.replace("Python","Java")
print(s1)
```

```
Welcome to Java
```

Character Classification

- **s.isalnum()**: returns **True** if s is non-empty and all its characters are alphanumeric (either a letter or a number) bao gồm chữ cái hoặc số
- **s.isalpha()**: returns **True** if s is non-empty and all its characters are alphabetic tất cả kí tự là chữ cái
- **s.isdigit()**: returns **True** if s is non-empty and all its characters are numeric digits tất cả kí tự chữ số

```
s = "python3"
print(s.isalnum())
```

```
True
```

```
s = "abc"
print(s.isalpha())
```

```
True
```

```
s = "2019"
print(s.isdigit())
```

```
True
```

Character Classification

trả về true khi nó non empty và thuộc hệ chữ cái aphabetic đồng thời nó đang đc viết ở dạng chữ thường không in hoa

- **s.islower()**: returns **True** if s is non-empty and all the alphabetic characters it contains are lowercase.
- **s.isupper()**: returns **True** if s is non-empty and all the alphabetic characters it contains are uppercase tương tự cái trên nhưng phải đc viết ở dạng in hoa tất cả thì mới trả về true

```
s = "Hello"
print(s.islower())
print(s.isupper())
```

```
False
False
```

Split

tác ra thành các chuỗi con

- **s.split(sep=None, maxsplit=-1)**: Splits a string into a list of substrings.
  - **Sep**: delimiter according which to split the string, default value is **None** (whitespace) nó sẽ coi mỗi dấu cách trắng là mốc để tách từng chữ ra
  - **Maxsplit**: maximum number of splits to do, default value is -1 (no limit)

```
1  s = "This is split function"
2  items = s.split()
3  print(items)
```

```
['This', 'is', 'split', 'function']
```

```
1  date = "20/11/2011"
2  items_1 = date.split(sep='/')
3  print(items_1)
4
5  items_2 = date.split(sep='/',maxsplit=1)
6  print(items_2)
```

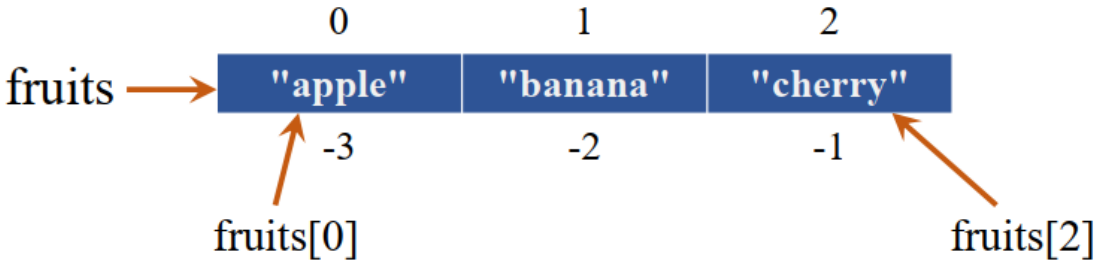
```
['20', '11', '2011']
['20', '11/2011']
```

List Indexing

- We can access the elements of a list using an integer index.

  myList[index]

- For example, fruits= ["apple", "banana", "cherry"]




```
1  fruits= ["apple","banana","cherry"]
2  print(fruits[0])
3  print(fruits[-1])
```

```
apple
cherry
```

- A list can contain sublists.
- To access the elements in a sublist, simply append an additional index:

```
1 hands = [['J','Q','K'],['2','2','2'],['6','A','K']]
2 print(hands[0])
3 print(hands[0][1])
4 print(hands[2][1])
```

```
['J', 'Q', 'K']
Q
A
```

- The slicing operator returns a slice of the list using the syntax:

  mylist[start : end]

- The slice is a sublist from index **start** to index **end – 1**

```
1 primes = [2, 3, 5, 7,11,13]
2 print(primes[2 : 4])
3 print(primes[:2]) # two first primes
4 print(primes[3:]) # all the primes from index 3
5 print(primes[:]) # all the primes
6 print(primes[1:-1]) # All the primes except the first and last
```

```
[5, 7]
[2, 3]
[7, 11, 13]
[2, 3, 5, 7, 11, 13]
[3, 5, 7, 11]
```

```
1  intNum = [6, 5, -9 ,0]
2  print(intNum[2])
3  # get last element in float num
4  print(intNum[-1])
5  print(matrix[1][0])
6  # get 3 frist element in ds
7  print(ds[:3])
8  print(ds[-3:])
9  print(ds[2:6])
10 # create a sublist that contains elements in even posoition
11 print(ds[::2])
12 print(ds[:])
```

```
-9
0
6
[0, 1, 2]
[7, 8, 9]
[2, 3, 4, 5]
[0, 2, 4, 6, 8]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

- Lists are "mutable", meaning we can change their elements.
- To change the value of a specific element, refer to the index number:

```
fruits= ["apple","banana","cherry"]
fruits[1] = "grapes"
print(fruits)
```

```
['apple', 'grapes', 'cherry']
```

```
fruits[1:] = ["kiwi","durian"]
print(fruits)
```

```
['apple', 'kiwi', 'durian']
```

```
fruits[1:] = [] # remove elements from 1 to last
print(fruits)
```

```
['apple']
```



```
number = [1,3,5,7,9]
number[1]=11
print(number)
number[-2:] = [13,17]
print(number)
number [-2:] = [8]
print(number)
number[:2] = [0]
print(number)
number[1:2]= []
print(number)
```

```
[1, 11, 5, 7, 9]
[1, 11, 5, 13, 17]
[1, 11, 5, 8]
[0, 5, 8]
[0, 8]
```

- Using **for** loop, which enables you to traverse the list sequentially without using an index variable. duyệt chuỗi không qua biến số

```
fruits= ["apple","banana","cherry"]
for f in fruits:
    print(f,end=" ")
```

```
apple banana cherry
```

- Use an index variable if you wish to traverse the list in a different order or change the elements in the list. duyện qua biến số

```
fruits= ["apple","banana","cherry"]
size = len(fruits)
for i in range(size-1, -1, -1): # i = size-1 --> 0
    print(fruits[i],end=" ")
```

```
cherry banana apple
```

```
numbers = [4,2,8,7]
s = 0
for num in numbers:
    s = s + num
print("Sum all elements in list: ",s)
```

```
Sum all elements in list:  21
```

```
# Print all positive elements in list
numbers = eval(input("Enter a list of integer: "))
flag = False
for num in numbers:
    if num > 0:
        flag = True
        print(num)
if not flag:
    print("The list has no positive elements!")
```

```
Enter a list of integer: [-9,0,4,-2,5]
4
5
```

Operator	Description
in	True if element x is in list.
not in	True if element x is not in list.
+	Concatenates list1 and list2.
*	n copies of list concatenated.

```
countries = ["Norway","Germany","Canada"]
"Canada" in countries
```

```
True
```

```
list1 = [1,2,3]
list2 = [4,5,6]
list3 = list1 + list2
print(list3)
print(list1*2)
```

```
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 1, 2, 3]
```

- **Comparing Lists**
  - Using the comparison operators (>, >=, <, <=, ==,!=)
  - The comparison uses ordering: the first two elements are compared, and if they differ this determines the outcome of the comparison; if they are equal, the next two elements are compared, and so on.

```
color1 = ["green", "red", "blue"]
color2 = ["red", "blue", "green"]
list1 = [1,2,3]
print(color1 != color2)
print(color1 > color2)
#color1 > list1 --> Error
```

```
True
False
```

cùng kiểu mới so sánh đc

- **mylist.extend(<iterable>)**: Extend list by appending elements from the iterable, behaves like the + operator.

những dữ liệu có nhiều phần tử mới xài đc extend và hàm này sẽ tách từng kí tự để chèn vào

```
1 list1 = [2,3,4]
2 list1.extend([5,6])
3 print(list1)
4 list1.extend("Python")
5 print(list1)
```

```
[2, 3, 4, 5, 6]
[2, 3, 4, 5, 6, 'P', 'y', 't', 'h', 'o', 'n']
```

- **mylist.count(<object>)**: Returns the number of times element <object> appears in the list

hàm đếm

```
1 list1 = [3,1,4,5,4]
2 list1.count(4)
```

```
2
```

- **mylist.index(<object>)**: Returns the index of the first occurrence of element <object> in the list.

vị trí xuất hiện đầu tiên của số cần tìm

```
1 list1 = [3,4,1,5,4]
2 list1.index(4)
```

```
1
```

- **mylist.insert(<index>, <object>)**: Inserts an element <object> at a given index.

vị trí giá trị

chèn vô 1 vị trí nào đó

```
1 list1 = [1,5,7,9]
2 list1.insert(1,3)
3 print(list1)
```

```
[1, 3, 5, 7, 9]
```

- **mylist.remove(<object>)**: Removes the first occurrence of element x from the list.

xóa phần tử xuất hiện đầu tiên trong list

```
1 list1 = [3,4,1,5,4]
2 list1.remove(4)
3 print(list1)
4 #List1.remove(6) --> Error: 6 not in List
```

```
[3, 1, 5, 4]
```

- **mylist.pop(index = -1)**: Removes the element at the given position and returns it. dùng để xóa tại vị trí cho trước

để trống thì mặc định xóa cái cuối

```
1  list1 = [3,4,1,5,4]
2  list1.pop(1) # return 4
3  print(list1)
4  list1.pop()
5  print(list1)
```

```
[3, 1, 5, 4]
[3, 1, 5]
```


- **mylist.reverse()**: Reverses the elements in the list.

hàm đảo ngược danh sách

```
1  odd = [1,3,5,7,9]
2  odd.reverse()
3  print(odd)
```

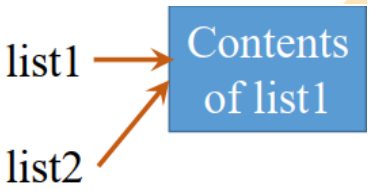
```
[9, 7, 5, 3, 1]
```

- **mylist.sort()**: Sorts the elements in the list in ascending order.

hàm sắp xếp theo thứ tự tăng dần

```
1  odd = [5,1,7,3,9]
2  odd.sort()
3  print(odd)
```

```
[1, 3, 5, 7, 9]
```



- list2 = list1 → Do not copy the contents
- So, get a duplicate copy of list1 into list2,

you have to copy individual elements from the source list to the target list using list comprehension or operator + or copy() function

```
1  list1 = [1,3,5,7]
2
3  list2 = [x for x in list1]
4
5  print(list2)
6  print(list2 is list1)
```

```
[1, 3, 5, 7]
False
```

Or simply

```
1  list1 = [1,3,5,7]
2
3  list2 = [] + list1
4
5  print(list2)
6  print(list2 is list1)
```

```
[1, 3, 5, 7]
False
```

Question 22: Write a program that prompts the user to enter two strings s1, s2 and sorts them in increasing order.

```
:
1 s1 = str(input('Enter s1: '))
2 s2 = str(input('Enter s2: '))
3 if len(s1)<=len(s2):
4     print(s1, s2)
5 else:
6     print(s2, s1 )
```

```
Enter s1: hello
Enter s2: hi
hi hello
```

Question 23: Write a program that prompts the user to enter a string and tests whether the string is a palindrome or not.

```
18]:
1 s = str(input('Enter a string: '))
2 a = s[::-1]
3 if a == s:
4     print('String ís a palindrome')
5 else:
6     print('String is not a palindrome')
```

```
Enter a string: malayalam
String ís a palindrome
```

Question 24: Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string.

```
In [27]:
1 def string(s):
2     s1 = s[0:2] + s[-2:]
3     if len(s1)>2:
4         return s1
5     else:
6         return -1
7 s = str(input('Enter a string: '))
8 print(string(s))
```

```
Enter a string: vu
vuvu
```

Question 25: Write a function named mismatch that accepts two strings as input arguments and returns:

- 0 if the two strings match exactly

- 1 if the two strings have the same length and mismatch in only one character.

- 2 if the two strings do not have the same length or mismatch in two or more characters.

Capital letters are considered the same as lower case letters. Here are some examples:

S1 = "Python", S2 = "Java" → return 2

S1 = "Hello There", S2 = "helloothere" → return 1

S1 = "dog" , S2 = "Dog" → return 0

In [32]:

```
def mismatch(s1,s2):
    if len(s1) == len(s2):
        return 0
    elif len(s1) == len(s2):
        count = 0
        for i in range(len(s1)):
            if s1[i]!=s2[i]:
                count = count +1
                return 1
        if count == 1:
            return 1
        else:
            return 2
    else:
        return 2
s1 = str(input('Enter s1: '))
s2 = str(input('Enter s2: '))
print(mismatch(s1,s2))

```

```
Enter s1: dog
Enter s2: Dog
0
```

Question 26: Write a function that checks whether a string is a valid password. Suppose the password rules are as follows:

- **A password must have at least eight characters.**
- **A password must consist of only letters and digits.**
- **A password must contain at least two digits.**

In [21]:

```
    def vaildpass(n):

        if len(n)<8:
            return False
        elif n.isalnum() == False:
            return False
        else:
            count = 0
            for e in n:
                if e.isdigit() == True:
                    count = count + 1
            if count < 2:
                return False
        return True
n = input('Enter n: ')
print(vaildpass(n))
```

```
Enter n: QUAN2004
True
```

Question 27: Write a program that prompts the user to enter a Social Security number in the format ddd-dd-dddd, where d is a digit. The program displays Valid SSN for a correct Social Security number or Invalid SSN otherwise.

In [24]:
```
ssn =input('Enter ssn: ')
flag = 1
if len(ssn) != 11:
    flag = 0
elif ssn[3] != '-' or ssn[6] != '-' or ssn[:3].isdigit == False\
or ssn[4:6] == False or ssn[-4:] == False:
    flag = 0
if flag == 1:
    print('Valid ssn')
else:
    print('Invalid ssn')
```

```
Enter ssn: 123-45-6789
Valid ssn
```

Question 28: Given list of numbers $a_0, a_1, \ldots, a_{n-1}$. Write a program to:

a. **Print all the negative elements in the list**
b. **Find sum of all the negative elements in the list**
c. **Check whether all elements in the list are positive or NOT**
d. **Sort the List in ascending /descending order**.

In [13]:
```
#Given list of numbers a0, a1,…., an-1.
ds = eval(input('Enter a list: '))
flag = False
s= 0
for e in ds:
    if e < 0:
        flag = True
        s = s+e
        print(e, end = ' ')

if flag == False:
    print('List has no negative elelementz')
else:
    print()
    print(s)
```

```
Enter a list: [-2,-4]
-2 -4
-6
```

Question 29: Write a program that prompts the user to enter a list of integers:

a. **Sum all elements in the list**
b. **Find minimum and maximum element in the list**
c. **Search an X element in the list**
d. **Count occurrences of an element X in the list**
e. **Print index of the first occurrence of element X in the list.**

In [1]:
```
ds = eval(input('Enter a list: '))
print(sum(ds))
print(min(ds))
print(max(ds))
x = eval(input('x= '))
if x in ds:
    print(f'{x} is in the list')
    print('fist occurence in the list', ds.index(x))
else:
    print(f'{x} is not in the list')
print(ds.count(x))

```

```
Enter a list: [1,2,3,4]
10
1
4
x= 3
3 is in the list
fist occurence in the list 2
1
```

Question 30: Write a program that lets user enter a integer number N and display:

a. **A box like one below:**

```
Enter a number: 5
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

```
Enter a number: 5
* * * * *
*       *
*       *
*       *
* * * * *
```

In [4]:
```
n = eval(input('Enter numbers: ')) # cách 1
for i in range(1, n + 1):
    print(n*' * ')
```

```
Enter numbers: 5
 *  *  *  *  *
 *  *  *  *  *
 *  *  *  *  *
 *  *  *  *  *
 *  *  *  *  *
```

In [14]:
```
n = eval(input('Enter numbers: '))
for i in range (1,n+1):
    if i==1 or i==n:
        print(n*"* ")
    else:
        print('*'+ (2*n - 3)*' '+'*')
```

```
Enter numbers: 5
* * * * *
*       *
*       *
*       *
* * * * *
```

b.

Enter a number: 5 * ** *** **** *****	Enter a number: 5 ***** **** *** ** *	Enter a number: 5 * ** *** **** *****

In [18]:
```
n = eval(input('Enter numbers: ')) # cách 1
for i in range(1, n + 1):
    print(i*' *')
```

```
Enter numbers: 5
 *
 * *
 * * *
 * * * *
 * * * * *
```

In [21]:
```
n = eval(input('Enter numbers: ')) # cách 1
for i in range(n, 0, -1 ):
    print(i*' * ')
```

```
Enter numbers: 5
 *  *  *  *  *
 *  *  *  *
 *  *  *
 *  *
 *
```

In [26]:
```
n = eval(input('Enter numbers: ')) # cách 1
for i in range(1, n + 1):
    print((n - i)*' '+i*'*')
```

```
Enter numbers: 5
    *
   **
  ***
 ****
*****
```

c. A triangle like one below:

```
Enter a number: 5
      *
     ***
    *****
   *******
  *********
```

In [38]:

```
n = eval(input('Enter numbers: ')) # cách 1
for i in range(1, n + 1):
    print(((n - i)*' '+(2*i-1)*'*'))
```

```
Enter numbers: 5
    *
   ***
  *****
 *******
*********
```

d. Shape “K” like one below:

```
Enter a number: 5
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

In [56]:

```

n = eval(input('Enter numbers: ')) # cách 1
for i in range(n, 0, -1 ):
    for j in range(1, i + 1 ):
        print(j,end= ' ')
    print()
for i in range(2, n+1 ):
    for j in range(1, i + 1 ):
        print(j,end= ' ')
    print()


```

```
Enter numbers: 5
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```