

---

**Program no:01**

**Aim:** Display future leap years from current year to a final year entered by user

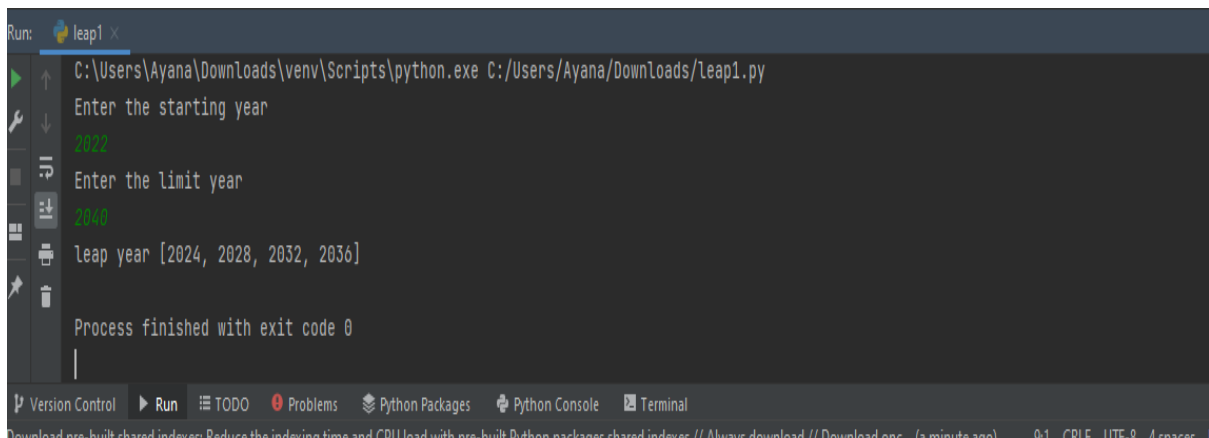
**Source Code:**

```
start_yr=int(input("Enter the starting year\n"))
limit_yr=int(input("Enter the limit year\n"))

lst=[]

for i in range(start_yr,limit_yr):
    if((i%4==0 and i%100!=0) or i%400==0):
        lst.append(i)

print("leap year",lst)
```

**Output:**A screenshot of a Python IDE window titled 'Run: leap1'. The command prompt shows the execution of 'C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/leap1.py'. The user is prompted to 'Enter the starting year' and enters '2022'. Then prompted to 'Enter the limit year' and enters '2040'. The output shows 'leap year [2024, 2028, 2032, 2036]'. At the bottom, it says 'Process finished with exit code 0'. The IDE interface includes a sidebar with icons for Explorer, Search, and Run and Debug, and a bottom bar with tabs for Version Control, Run, TODO, Problems, Python Packages, Python Console, and Terminal.

```
Run: leap1 x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/leap1.py
Enter the starting year
2022
Enter the limit year
2040
leap year [2024, 2028, 2032, 2036]
Process finished with exit code 0
```

---

**Program no:02**

**Aim:** List comprehensions:

- a. Generate positive list of numbers from a given list of integers
- b. Square of N numbers
- c. Form a list of vowels selected from a given word
- d. List ordinal value of each element of a word (Hint: use ord() to get ordinal values)

**Source Code:**

**a.**

```
li=[34,-90,45,-7,-3,21,67,89,100]
pos=[n for n in li if n>0]
print("POSITIVE NUMBERS ARE:",pos)
```

**b.**

```
li=[2,4,6]
sq=[n*n for n in li]
print("SQUARES ARE:",sq)
```

**c.**

```
w=input("ENTER A WORD:")
v=[x for x in w if x=='a' or x=='A' or x=='e' or x=='E' or x=='i' or x=='I' or
x=='o' or x=='O' or x=='u' or x=='U']
print(v)
```

**d.**

```
l=["bat","ball"]
ord=[ord(element)for sub in l for element in sub]
print(ord)
```

## Output:

a.

```
Run: positive1 x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/positive1.py
POSITIVE NUMBERS ARE: [34, 45, 21, 67, 89, 100]

Process finished with exit code 0
```

b.

```
Run: squares1 x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/squares1.py
SQUARES ARE: [4, 16, 36]

Process finished with exit code 0
```

c.

```
Run: vowels1 x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/vowels1.py
ENTER A WORD: string
['i']

Process finished with exit code 0
```

d.

```
Run: ord1 x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/ord1.py
[98, 97, 116, 98, 97, 108, 108]

Process finished with exit code 0
```

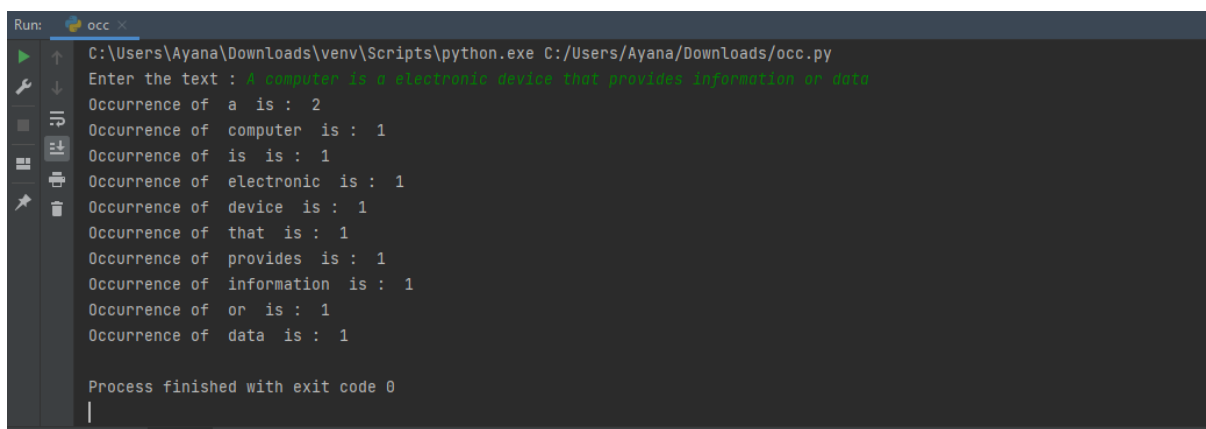
---

**Program no:03**

**Aim:** Count the occurrences of each word in a line of text.

**Source Code:**

```
text = str(input("Enter the text : "))
split = text.lower().split()
checkedList = []
for i in range(len(split)):
    checkedList.append(split[i])
    count = 1
    if checkedList.count(split[i]) <= 1:
        for j in range(i+1, len(split)):
            if split[i] == split[j]:
                count += 1
    print("Occurrence of ", split[i], " is : ", count)
```

**Output:**A screenshot of a Windows command prompt window titled 'Run: occ'. The command executed is 'C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/occ.py'. The input text is 'A computer is a electronic device that provides information or data'. The output shows the occurrence of each word: 'a' is 2, 'computer' is 1, 'is' is 1, 'electronic' is 1, 'device' is 1, 'that' is 1, 'provides' is 1, 'information' is 1, 'or' is 1, and 'data' is 1. The process finished with exit code 0.

```
Run: occ
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/occ.py
Enter the text : A computer is a electronic device that provides information or data
Occurrence of a is : 2
Occurrence of computer is : 1
Occurrence of is is : 1
Occurrence of electronic is : 1
Occurrence of device is : 1
Occurrence of that is : 1
Occurrence of provides is : 1
Occurrence of information is : 1
Occurrence of or is : 1
Occurrence of data is : 1
Process finished with exit code 0
```

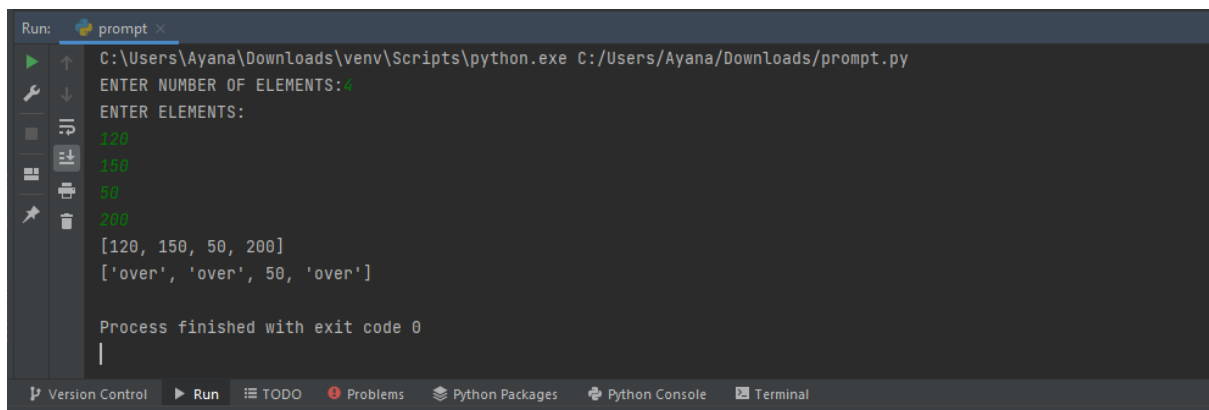
**Program no:04**

**Aim:** Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.

**Source Code:**

```
n=int(input("ENTER NUMBER OF ELEMENTS:"))
print("ENTER ELEMENTS:")
list=[]
res=[]
for i in range (0,n):
    ele=int(input())
    list.append(ele)
print(list)
for i in list:
    if i>100:
        res.append('over')
    else:
        res.append(i)
print(res)
```

---

**Output:**

The screenshot shows a terminal window titled 'Run: prompt' with the following content:

```
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/prompt.py
ENTER NUMBER OF ELEMENTS:4
ENTER ELEMENTS:
120
150
50
200
[120, 150, 50, 200]
['over', 'over', 50, 'over']

Process finished with exit code 0
```

The IDE interface includes a sidebar with icons for Version Control, Run, TODO, Problems, Python Packages, Python Console, and Terminal. The bottom status bar shows 'Run' and 'Terminal' tabs.

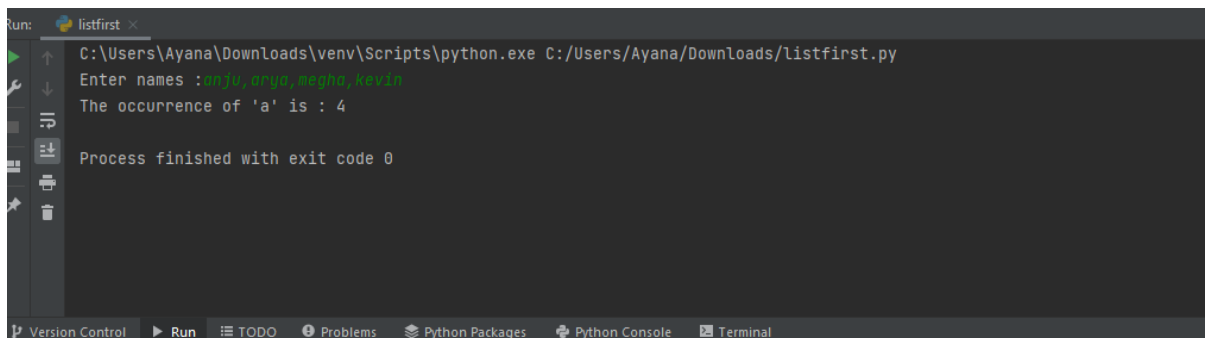
---

**Program no:05**

**Aim:** Store a list of first names. Count the occurrences of 'a' within the list

**Source Code:**

```
list=input("Enter names :")
count=0
for x in list:
    if x == 'a':
        count+=1
print("The occurrence of 'a' is :",count)
```

**Output:**

```
Run: listfirst
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/listfirst.py
Enter names :anju, arun, megha, kevin
The occurrence of 'a' is : 4
Process finished with exit code 0
```

---

**Program no:06**

**Aim:** Enter 2 lists of integers. Check:

- a. Whether list are of same length
- b. whether list sums to same value
- c. whether any value occur in both

**Source Code:**

```
l1=[20,40,30,10,90,15]
```

```
l2=[80,25,50,90,120,12]
```

```
print("length of list1 :",l1)
```

```
print("length of list2 :",l2)
```

```
if len(l1)==len(l2):
```

```
    print("Two lists are of same length")
```

```
total=sum(l1)
```

```
print("sum of list1 :",total)
```

```
total=sum(l2)
```

```
print("sum of list2 :",total)
```

```
if sum(l1)==sum(l2):
```

```
    print("Two lists sum is equal")
```

```
print("Value occur in both:")
```



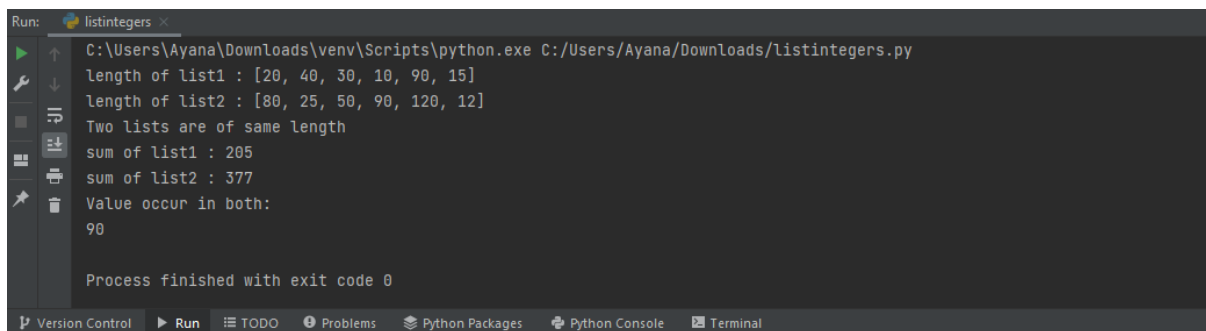
---

```
for i in l1:
```

```
    if i in l2:
```

```
        print(i)
```

## Output:



```
Run: listintegers
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/listintegers.py
length of list1 : [20, 40, 30, 10, 90, 15]
length of list2 : [80, 25, 50, 90, 120, 12]
Two lists are of same length
sum of list1 : 205
sum of list2 : 377
Value occur in both:
90

Process finished with exit code 0
```

---

**Program no:07**

**Aim:** Get a string from an input string where all occurrences of first character replaced with '\$', except first character. [eg: onion -> oni\$n]

**Source Code:**

```
n=(input("enter the string"))
c=n[0]
for i in n:
    if i==c:
        n=n.replace(i,'$')
    n=c+n[1:]
print(n)
```

**Output:**

```
Run: replace1 x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/replace1.py
enter the stringonion
oni$n
Process finished with exit code 0
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download (17)

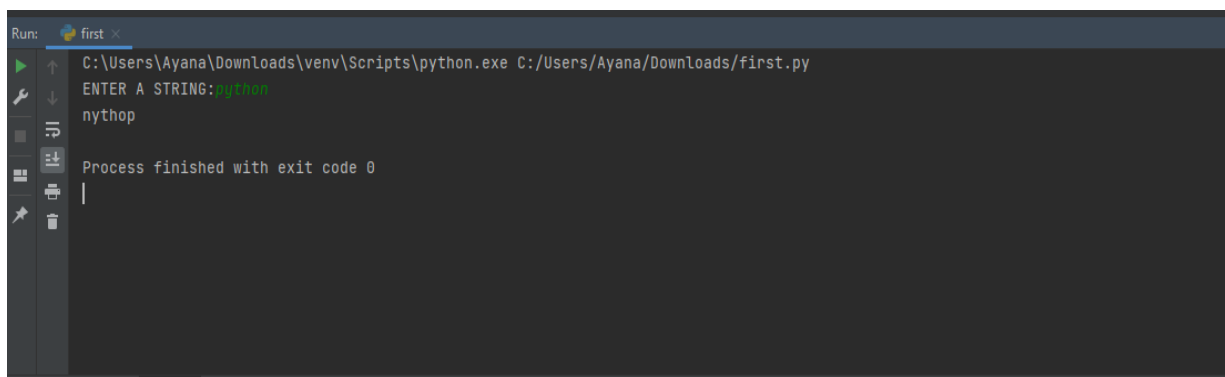
---

**Program no:08**

**Aim:** Create a string from given string where first and last characters exchanged. [eg: python -> nythop]

**Source Code:**

```
str1=input("ENTER A STRING:")  
a=str1[-1]  
b=str1[1:-1]  
c=str1[0]  
print(a+b+c)
```

**Output:**

```
Run: first  
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/first.py  
ENTER A STRING:python  
nythop  
Process finished with exit code 0
```

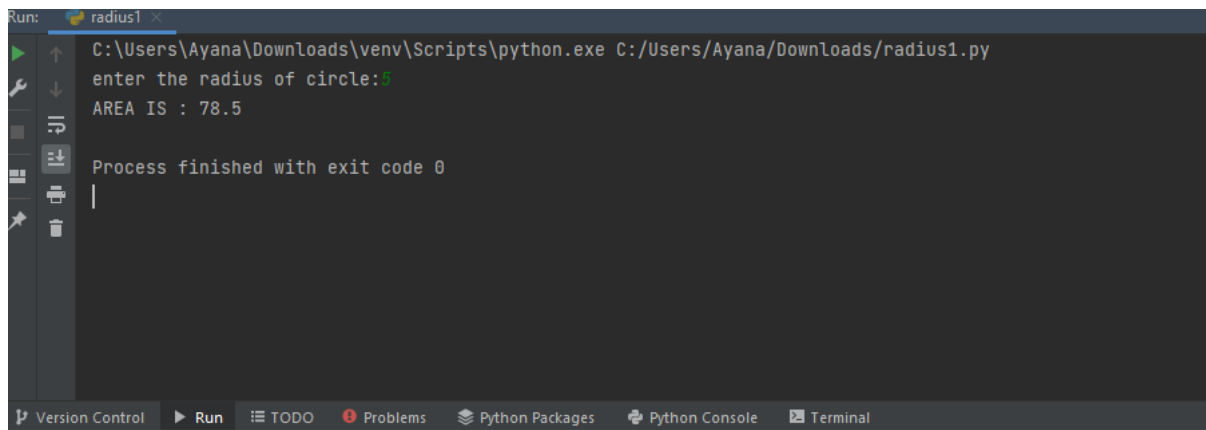
---

**Program no:09**

**Aim:** Accept the radius from user and find area of circle.

**Source Code:**

```
r=int(input("enter the radius of circle:"))  
print("AREA IS :"+3.14*(r*r))
```

**Output:**

The screenshot shows a Python IDE interface. The top bar indicates the file is named 'radius1'. The main editor area contains the following code:  

```
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/radius1.py  
enter the radius of circle:  
AREA IS : 78.5  
  
Process finished with exit code 0
```

  
The bottom status bar shows various IDE features: Version Control, Run, TODO, Problems, Python Packages, Python Console, and Terminal.

---

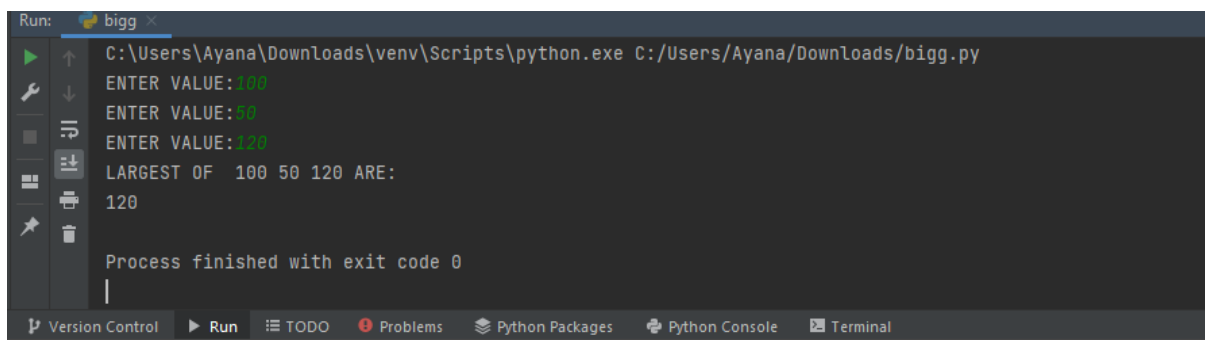
**Program no:10**

**Aim:** Find biggest of 3 numbers entered

**Source Code:**

```
l=int(input("ENTER VALUE:"))  
m=int(input("ENTER VALUE:"))  
n=int(input("ENTER VALUE:"))  
print("LARGEST OF ",l,m,n,"ARE:")  
print(max(l,m,n))
```

**Output:**



The screenshot shows a Python IDE window titled 'Run: bigg'. The command prompt shows the execution of 'C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/bigg.py'. The program prompts for three values: 'ENTER VALUE:100', 'ENTER VALUE:50', and 'ENTER VALUE:120'. The output is 'LARGEST OF 100 50 120 ARE: 120'. The process finished with exit code 0. The IDE interface includes a sidebar with icons for Version Control, Run, TODO, Problems, Python Packages, Python Console, and Terminal.

```
Run: bigg  
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/bigg.py  
ENTER VALUE:100  
ENTER VALUE:50  
ENTER VALUE:120  
LARGEST OF 100 50 120 ARE:  
120  
  
Process finished with exit code 0
```

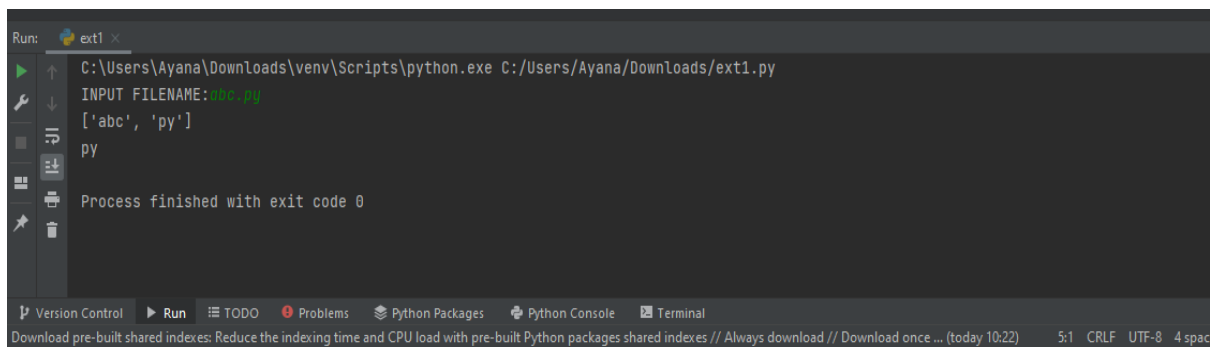
---

**Program no:11**

**Aim:** Accept a file name from user and print extension of that.

**Source Code:**

```
file=input("INPUT FILENAME:")  
fext=file.split(".")  
print(fext)  
print(fext[-1])
```

**Output:**

The screenshot shows a Python IDE interface. The top bar indicates the file is named 'ext1'. The main editor area displays the following code and its output:

```
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/ext1.py  
INPUT FILENAME: abc.py  
['abc', 'py']  
py  
Process finished with exit code 0
```

The bottom status bar shows various IDE features like Version Control, Run, TODO, Problems, Python Packages, Python Console, and Terminal. It also includes a message about downloading pre-built shared indexes and the current file encoding (5:1 CRLF UTF-8 4 spac).

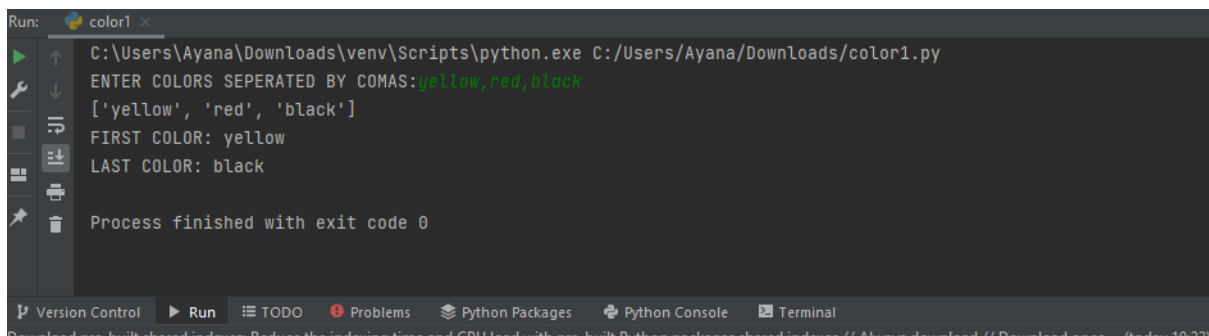
---

**Program no:12**

**Aim:** Create a list of colors from comma-separated color names entered by user. Display first and last colors.

**Source Code:**

```
color=input("ENTER COLORS SEPERATED BY COMAS:")
co_list=color.split(',')
print(co_list)
print("FIRST COLOR:",co_list[0])
print("LAST COLOR:",co_list[-1])
```

**Output:**

```
Run: color1 x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/color1.py
ENTER COLORS SEPERATED BY COMAS:yellow,red,black
['yellow', 'red', 'black']
FIRST COLOR: yellow
LAST COLOR: black
Process finished with exit code 0
```

---

**Program no:13**

**Aim:** Accept an integer n and compute  $n+nn+nnn$ .

**Source Code:**

```
n=int(input("ENTER A NUMBER:"))
a=int(n*n)
b=int(n*n*n)
print("n is:",n)
print("a is:",a)
print("b is:",b)
print(int(n)+a+b)
```

**Output:**

```
Run: int1 x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/int1.py
ENTER A NUMBER: 5
n is: 5
a is: 25
b is: 125
155
Process finished with exit code 0
```



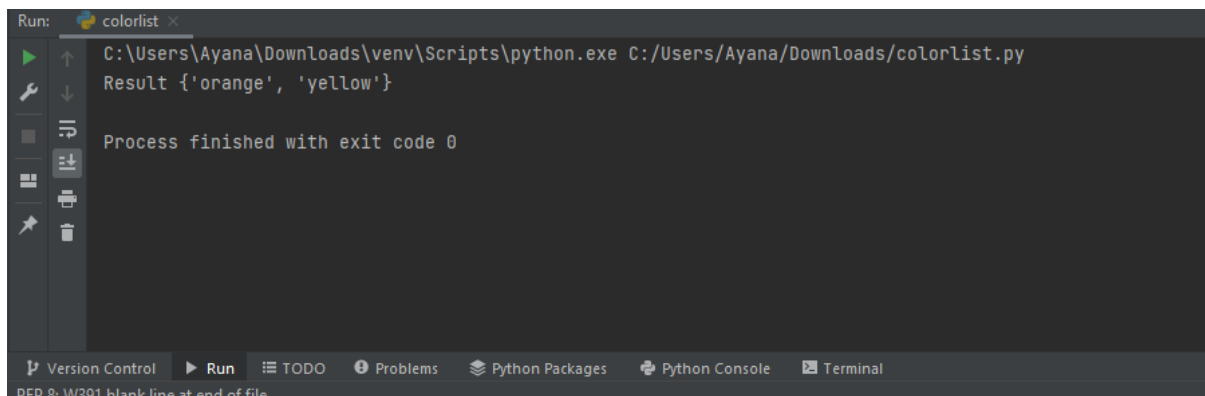
---

**Program no:14**

**Aim:** Print out all colors from color-list1 not contained in color-list2.

**Source Code:**

```
color_list1 = {"red","green","yellow","orange"}  
color_list2 = {"white","red","blue","violet","green"}  
new_list = color_list1.difference(color_list2)  
print("Result",new_list)
```

**Output:**

The screenshot shows a Python IDE window titled 'colorlist'. The command prompt shows the execution of 'C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/colorlist.py'. The output is 'Result {'orange', 'yellow'}'. Below the output, it says 'Process finished with exit code 0'. The IDE interface includes a sidebar with icons for running, debugging, and other functions, and a bottom status bar with tabs for 'Version Control', 'Run', 'TODO', 'Problems', 'Python Packages', 'Python Console', and 'Terminal'.

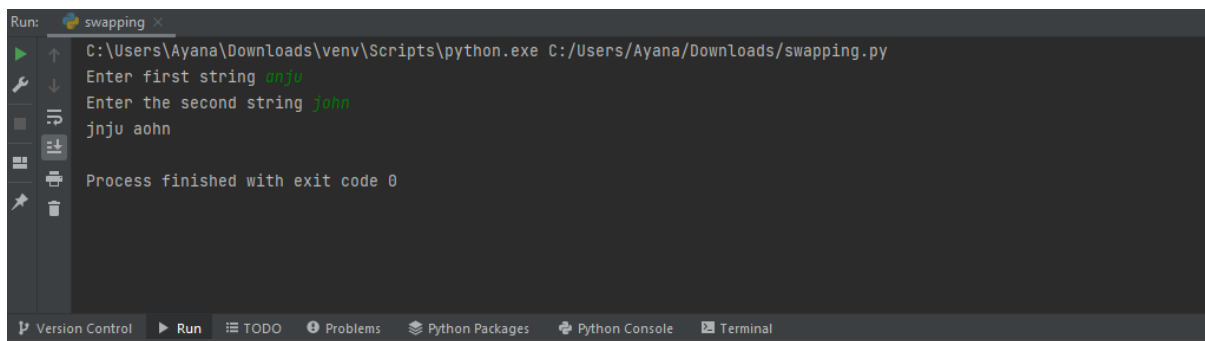
---

**Program no:15**

**Aim:** Create a single string separated with space from two strings by swapping the character at position 1.

**Source Code:**

```
str1 = input("Enter first string ")
str2 = input("Enter the second string ")
a = str2[:1] + str1[1:]
b = str1[:1] + str2[1:]
print(a+' '+b)
```

**Output:**A screenshot of a Python IDE's Run console. The title bar says 'Run: swapping'. The command prompt shows the execution of 'C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/swapping.py'. The user input is shown as 'Enter first string anju' and 'Enter the second string john'. The output is 'jnju aohn'. The console ends with 'Process finished with exit code 0'. The IDE interface includes a sidebar with icons for Version Control, Run, TODO, Problems, Python Packages, Python Console, and Terminal. The bottom status bar shows 'Run', 'TODO', 'Problems', 'Python Packages', 'Python Console', and 'Terminal'.

---

**Program no:16**

**Aim:** Sort dictionary in ascending and descending order.

**Source Code:**

```
y = {'john': 40, 'laly': 2, 'ayana': 1, 'anju': 3, 'aleena':5}
```

```
l = list(y.items())
```

```
l.sort()
```

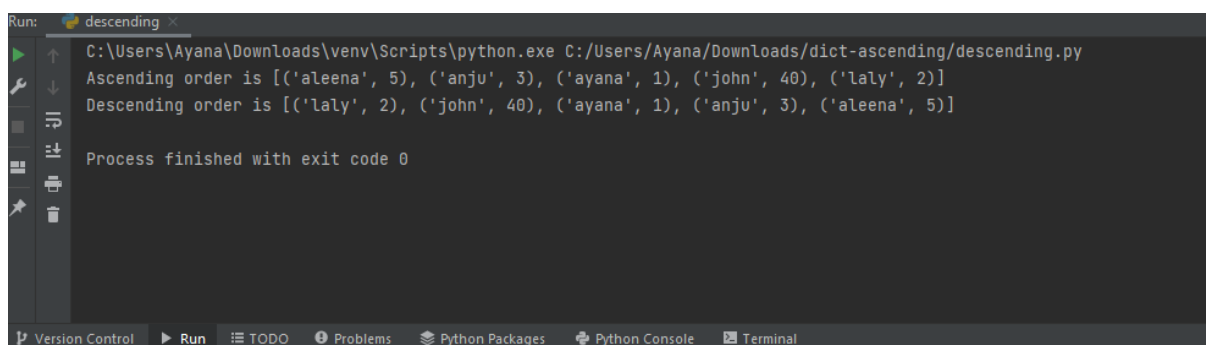
```
print('Ascending order is', l)
```

```
l = list(y.items())
```

```
l.sort(reverse=True)
```

```
print('Descending order is', l)
```

**Output:**



```
Run: descending x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/dict-ascending/descending.py
Ascending order is [('aleena', 5), ('anju', 3), ('ayana', 1), ('john', 40), ('laly', 2)]
Descending order is [('laly', 2), ('john', 40), ('ayana', 1), ('anju', 3), ('aleena', 5)]
Process finished with exit code 0
```

---


**Program no:17**

**Aim:** Merge two dictionaries.

**Source Code:**

```
dic1={'a':1000,'b':2000}
dic2={'u':1000,'v':20}
print("DICTIONARY 1:",dic1)
print("DICTIONARY 2:",dic2)
d=dic1.copy()
d.update(dic2)
print("MERGED ONE IS:",d)
```

**Output:**

A screenshot of a Python IDE's 'Run' console. The console shows the execution of a script named 'merge1.py'. The output is as follows:  
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/merge1.py  
DICTIONARY 1: {'a': 1000, 'b': 2000}  
DICTIONARY 2: {'u': 1000, 'v': 20}  
MERGED ONE IS: {'a': 1000, 'b': 2000, 'u': 1000, 'v': 20}  
Process finished with exit code 0  
The IDE interface includes a sidebar with icons for Run, Debug, and other functions, and a bottom status bar with tabs for Version Control, Run, TODO, Problems, Python Packages, Python Console, and Terminal.

---

**Program no:18**

**Aim:** Find gcd of 2 numbers.

**Source Code:**

```
num1 = int(input("Enter 1st number: "))
num2 = int(input("Enter 2nd number: "))

i = 1

while(i <= num1 and i <= num2):

    if(num1 % i == 0 and num2 % i == 0):

        gcd = i

    i = i + 1

print("GCD is", gcd)
```

**Output:**A screenshot of a Python IDE's Run console. The title bar says 'Run: gcd'. The command prompt shows the execution of a Python script: 'C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/dict-ascending/gcd.py'. The input shows 'Enter 1st number: 80' and 'Enter 2nd number: 100'. The output shows 'GCD is 20'. At the bottom, it says 'Process finished with exit code 0'. The IDE interface includes a sidebar with icons for Run, Debug, and other tools, and a bottom status bar with tabs for Version Control, Run, TODO, Problems, Python Packages, Python Console, and Terminal.

---

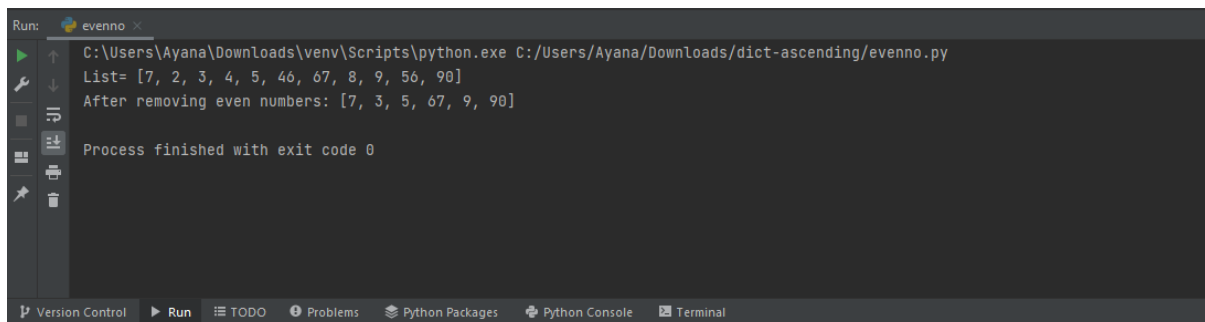
**Program no:19**

**Aim:** From a list of integers, create a list removing even numbers.

**Source Code:**

```
list = [7,2,3,4,5,46,67,8,9,56,90]
print("List=",list)
for x in list:
    if(x%2)==0:
        list.remove(x)

print("After removing even numbers:", list)
```

**Output:**

```
Run: evenno
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/dict-ascending/evenno.py
List= [7, 2, 3, 4, 5, 46, 67, 8, 9, 56, 90]
After removing even numbers: [7, 3, 5, 67, 9, 90]
Process finished with exit code 0
```

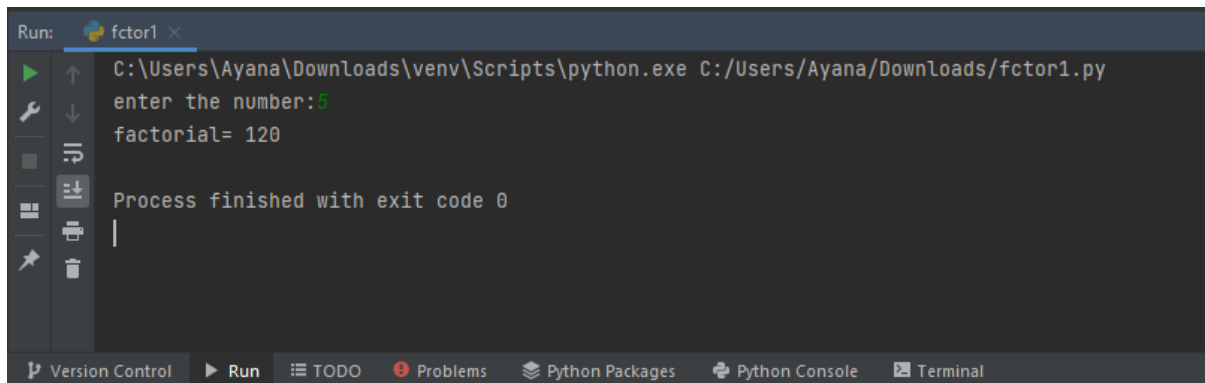
---

**Program no:20**

**Aim:** Program to find the factorial of a number

**Source Code:**

```
a=int(input("enter the number:"))
def fact(num):
    fact = 1
    for i in range(num,0,-1):
        fact=fact*i
    print("factorial=",fact)
fact(a)
```

**Output:**

```
Run: fctor1 x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/fctor1.py
enter the number:5
factorial= 120
Process finished with exit code 0
```

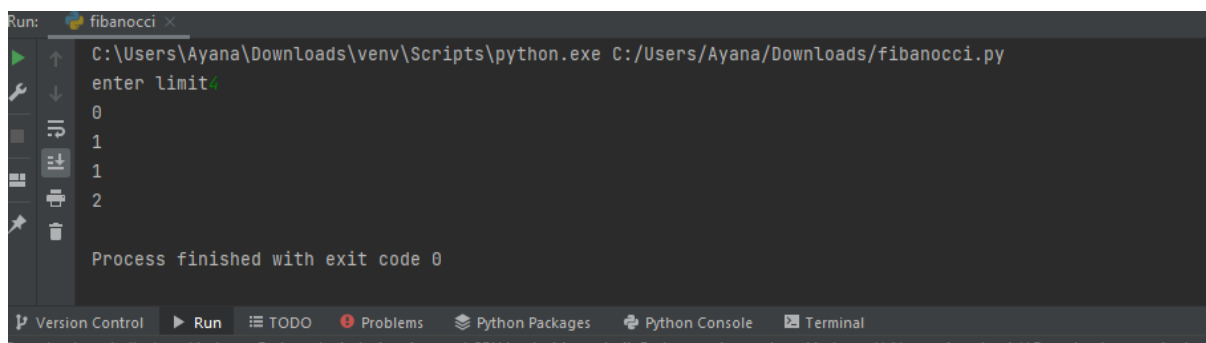
---

**Program no:21**

**Aim:** Generate Fibonacci series of N terms

**Source Code:**

```
n1=0
n2=1
num=int(input("enter limit"))
print(n1)
print(n2)
for i in range(0,num-2):
    n3=n1+n2
    print(n3)
    n1=n2
    n2=n3
```

**Output:**

```
Run: fibanocci x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/fibanocci.py
enter limit
5
0
1
1
2
3
Process finished with exit code 0
```



---

**Program no:22**

**Aim:** Find the sum of all items in a list.

**Source Code:**

```
list=[1,2,3,4]
sum=0
for i in list:
    sum=sum+i
print("sum=",sum)
```

**Output:**



The screenshot shows a Python IDE window titled 'Run: sumall'. The command prompt displays the execution of the script: `C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/sumall.py`. The output is `sum= 10`. Below the output, it states 'Process finished with exit code 0'. The IDE interface includes a sidebar with icons for running, debugging, and other functions, and a bottom status bar with tabs for 'Version Control', 'Run', 'TODO', 'Problems', 'Python Packages', 'Python Console', and 'Terminal'. A message at the bottom reads: 'Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download'.

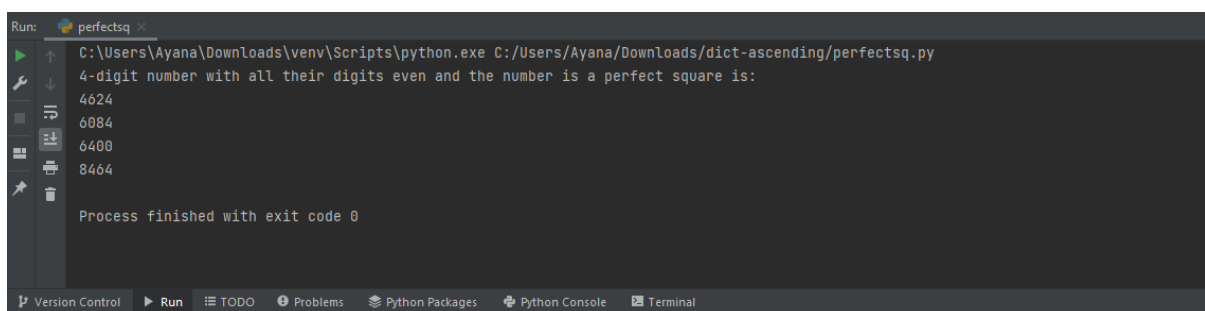
---

**Program no:23**

**Aim:** Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.

**Source Code:**

```
print("4-digit number with all their digits even and the number is a perfect  
square is: ")  
for i in range(1000,10000,1):  
  
    for j in range(32,100,1):  
  
        if i == j*j:  
  
            string = str(i)  
  
            if int(string[0])%2 == 0 and int(string[1])%2 == 0 and  
int(string[2])%2== 0 and int(string[3])%2 == 0:  
  
                print(i)
```

**Output:**

```
Run: perfectsq x  
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/dict-ascending/perfectsq.py  
4-digit number with all their digits even and the number is a perfect square is:  
4624  
6084  
6400  
8464  
Process finished with exit code 0  
Version Control Run TODO Problems Python Packages Python Console Terminal
```

---

**Program no:24**

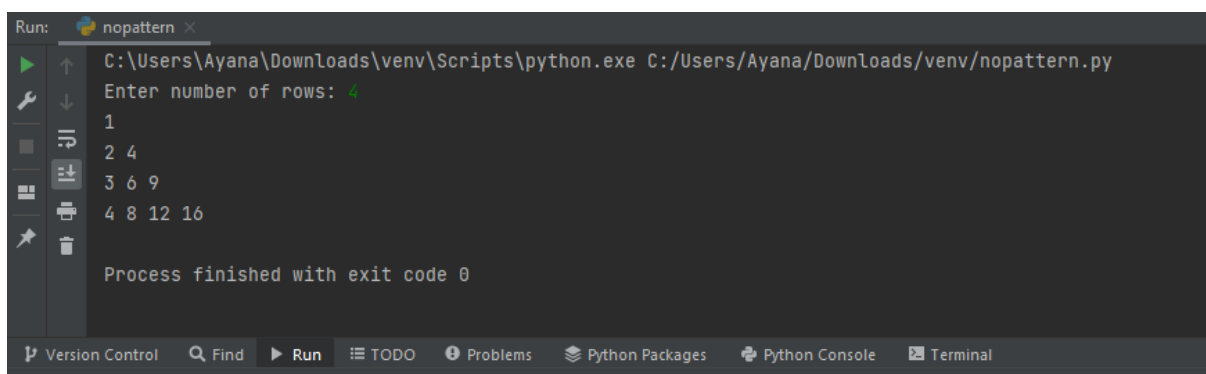
**Aim:** Display the given pyramid with step number accepted from user.

Eg: N=4

```
1
2 4
3 6 9
4 8 12 16
```

**Source Code:**

```
rows = int(input("Enter number of rows: "))
for i in range(1, rows + 1):
    for j in range(1, i + 1):
        square=i*j
        print(i*j, end=' ')
    print("")
```

**Output:**

The screenshot shows a Python IDE window titled 'Run: nopattern'. The command line shows the execution of 'C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:\Users\Ayana\Downloads\venv\nopattern.py'. The input 'Enter number of rows: 4' is shown. The output is a multiplication table pyramid: '1', '2 4', '3 6 9', and '4 8 12 16'. The status bar at the bottom indicates 'Process finished with exit code 0'.

---

**Program no:25**

**Aim:** Count the number of characters (character frequency) in a string

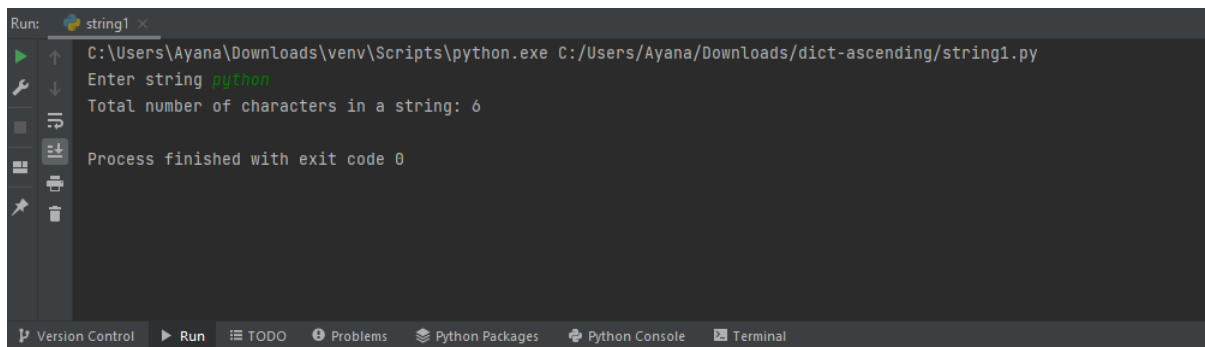
**Source Code:**

```
string = input("Enter string ")
count = 0;

for i in range(0, len(string)):
    if (string[i] != ' '):
        count = count + 1;

print("Total number of characters in a string: " + str(count));
```

**Output:**

A screenshot of a Python IDE window titled 'string1'. The main editor area shows the execution of a Python script. The command prompt shows 'C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/dict-ascending/string1.py'. The user input 'python' is shown in green. The output 'Total number of characters in a string: 6' is shown in white. The status bar at the bottom indicates 'Process finished with exit code 0'. The IDE interface includes a sidebar with icons for Version Control, Run, TODO, Problems, Python Packages, Python Console, and Terminal.

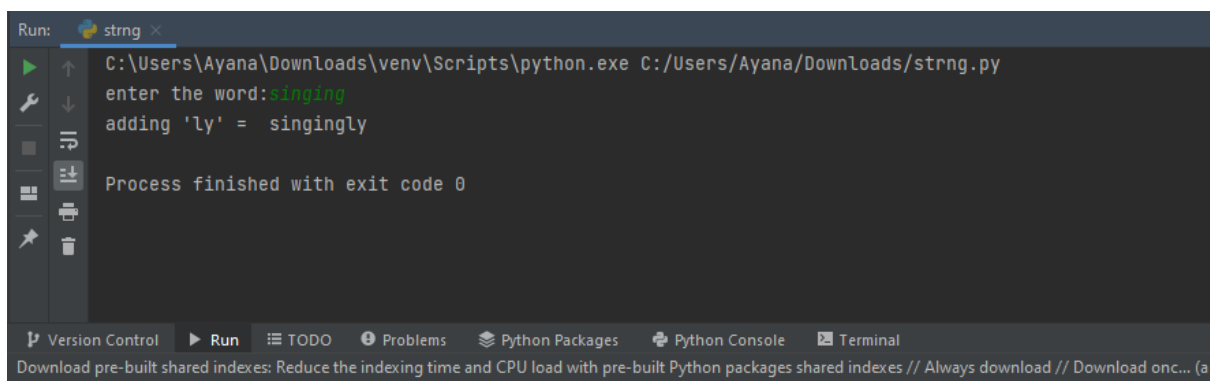
---

**Program no:26**

**Aim:** Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'

**Source Code:**

```
a=input("enter the word:")  
  
def words(word):  
    word1=word[-3:]  
    if word1 != "ing":  
        print("adding 'ing' = ",word+"ing")  
    else:  
        print("adding 'ly' = ",word+"ly")  
words(word=a)
```

**Output:**

```
Run: strng x  
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/strng.py  
enter the word:singing  
adding 'ly' =  singingly  
Process finished with exit code 0  
Version Control Run TODO Problems Python Packages Python Console Terminal  
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download onc... (a
```

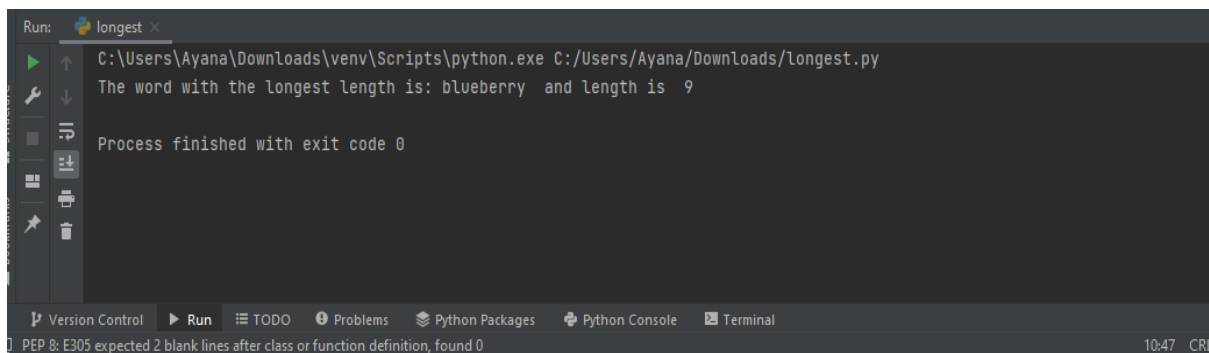
---

**Program no:27**

**Aim:** Accept a list of words and return length of longest word.

**Source Code:**

```
def longestLength(a):  
    max1 = len(a[0])  
    temp = a[0]  
    for i in a:  
        if (len(i) > max1):  
            max1 = len(i)  
            temp = i  
    print("The word with the longest length is:", temp,  
          " and length is ", max1)  
a = ["apple", "orange", "cherry", "pomegranate"]  
longestLength(a)
```

**Output:**

```
Run: longest x  
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/longest.py  
The word with the longest length is: blueberry and length is 9  
Process finished with exit code 0  
Version Control Run TODO Problems Python Packages Python Console Terminal  
PEP 8: E305 expected 2 blank lines after class or function definition, found 0 10:47 CRI
```

---

**Program no:28**

**Aim:** Construct following pattern using nested loop

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

**Source Code:**

```
n = 5;  
for i in range(n):  
    for j in range(i):  
        print('* ', end="")  
    print()  
  
for i in range(n, 0, -1):  
    for j in range(i):  
        print('* ', end="")  
    print()
```

**Output:**

```
Run: pattern x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/dict-ascending/pattern.py

*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*

Process finished with exit code 0
```

Version Control Run TODO Problems Python Packages Python Console Terminal



---

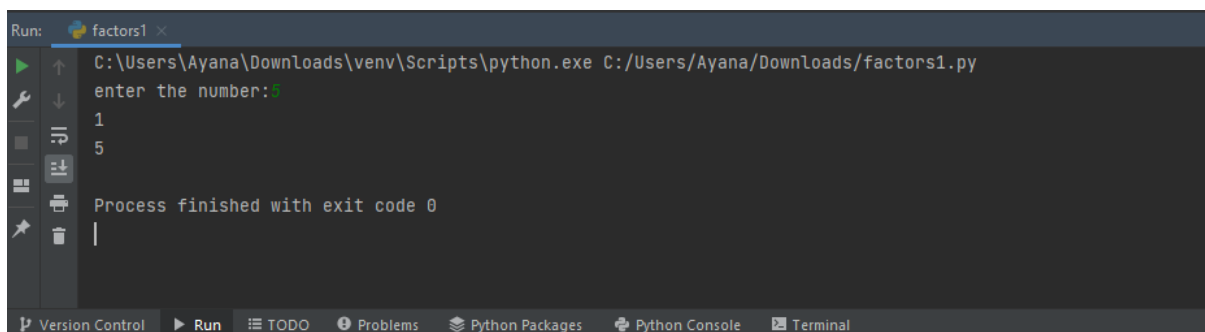
**Program no:29**

**Aim:** Generate all factors of a number.

**Source Code:**

```
a=int(input("enter the number:"))  
  
def factor(num):  
    for i in range(1,num+1):  
        if num % i == 0:  
            print(i)  
  
factor(a)
```

**Output:**



The screenshot shows a Python IDE window titled 'Run: factors1'. The command prompt shows the execution of 'C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:\Users\Ayana\Downloads/factors1.py'. The user input 'enter the number:' is followed by the number '5'. The output shows the factors '1' and '5'. The process finished with exit code 0. The IDE interface includes a sidebar with icons for Version Control, Run, TODO, Problems, Python Packages, Python Console, and Terminal.

---

**Program no:30**

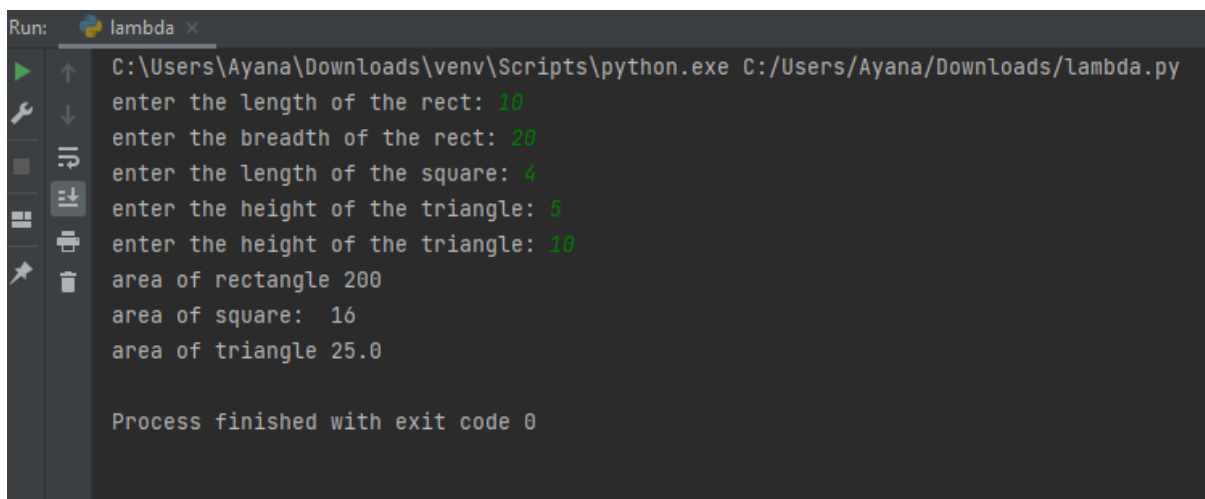
**Aim:** Write lambda functions to find area of square, rectangle and triangle.

**Source Code:**

```
rect=lambda x,y:x*y
sqr=lambda m:m*m
tri=lambda b,h:0.5*b*h

i=int(input("enter the length of the rect: "))
j=int(input("enter the breadth of the rect: "))
k=int(input("enter the length of the square: "))
i1=int(input("enter the height of the triangle: "))
j1=int(input("enter the height of the triangle: "))

print("area of rectangle",rect(i,j))
print("area of square: ",sqr(k))
print("area of triangle",tri(i1,j1))
```

**Output:**

```
Run: lambda x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/Lambda.py
enter the length of the rect: 10
enter the breadth of the rect: 20
enter the length of the square: 4
enter the height of the triangle: 5
enter the height of the triangle: 10
area of rectangle 200
area of square: 16
area of triangle 25.0

Process finished with exit code 0
```

---

**Program no:31**

**Aim:** Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements. (Include selective import of modules and import \* statements)

**Source Code:**

```
def Circle(r):  
    print("Area = ", 3.14*r**2)  
    print("Perimeter = ", 2*3.14*r)  
  
def Rectangle(l,b):  
    print("Area = ", l*b)  
    print("Perimeter = ", (2*l)+(2*b))  
  
def Cuboid(l,w,h):  
    print("Perimeter of cuboid= ", 4*(l+w+h))  
    print("Area of cuboid= ", 2*l*w + 2*l*h + 2*h*w)  
  
def Sphere(r):  
    print("Perimeter of sphere = ", 2*3.14*r)  
    print("Area of sphere= ", 4*3.14*r**2)  
  
import Graphics.Circle  
print("CIRCLE")  
r = int(input("Enter radius "))  
Graphics.Circle.Circle(r)
```

```
import Graphics.Rectangle
print("RECTANGLE")
l = int(input("Enter length "))
b = int(input("Enter breadth "))
Graphics.Rectangle.Rectangle(l,b)
```

```
from Graphics.ThreeDgraphics import Cuboid
print("CUBOID")
l = int(input("Enter length "))
w = int(input("Enter width "))
h = int(input("Enter height "))
Graphics.ThreeDgraphics.Cuboid.Cuboid(l,w,h)
```

```
from Graphics.ThreeDgraphics import Sphere
print("SPHERE")
r = int(input("Enter radius "))
Graphics.ThreeDgraphics.Sphere.Sphere(r)
```

**Output:**

```
CIRCLE
Enter radius 5
Area = 78.5
Perimeter = 31.400000000000002
RECTANGLE
Enter length 8
Enter breadth 4
Area = 32
Perimeter = 24
CUBOID
Enter length 6
Enter width 7
Enter height 8
Perimeter of cuboid= 84
Area of cuboid= 292
SPHERE
Enter radius 5
Perimeter of sphere = 31.400000000000002
Area of sphere= 314.0
```

---

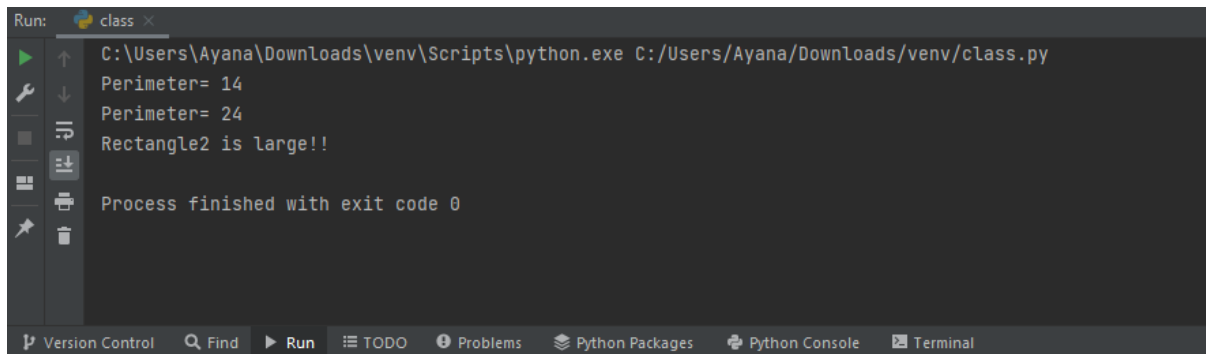
**Program no:32**

**Aim:** Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.

**Source Code:**

```
class rectangle:
    def __init__(self,l1,b1):
        self.length=l1
        self.breadth=b1
    def area(self):
        return(self.length*self.breadth)
    def perimeter(self):
        print("Perimeter=", 2*(self.length+self.breadth))
    def compare(self,obj):
        if(self.area()>obj.area()):
            print("Rectangle1 is large!!")
        else:print("Rectangle2 is large!!")
R1=rectangle(2,5)
R1.area()
R1.perimeter()
R2=rectangle(4,8)
R2.area()
R2.perimeter()
R1.compare(R2)
```

---

**Output:**

```
Run: class
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/venv/class.py
Perimeter= 14
Perimeter= 24
Rectangle2 is large!!
Process finished with exit code 0
```

Version Control Find Run TODO Problems Python Packages Python Console Terminal

---

**Program no:33**

**Aim:** Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.

**Source Code:**

```
class bank:
    def __init__(self,ac,n1,t1,b1):
        self.accntno=ac
        self.name=n1
        self.type=t1
        self.balance=b1

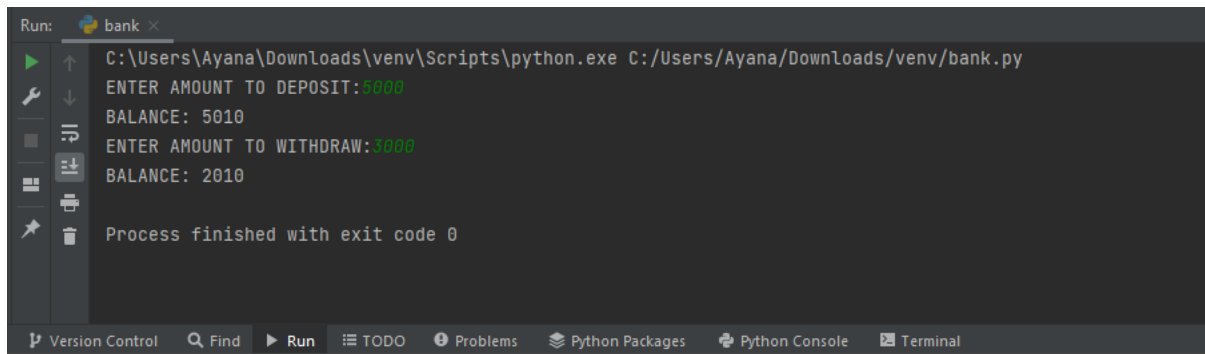
    def deposit(self,amnt):
        self.balance+=amnt
        print("BALANCE:",self.balance)

    def withdraw(self, amnt):
        if(self.balance>amnt):
            self.balance-= amnt
            print("BALANCE:", self.balance)
        else:
            print("SORRY :) BALANCE IS TOO LOW")

b=bank(210200,'ANNU','SAVINGS',10)
a=int(input("ENTER AMOUNT TO DEPOSIT:"))
b.deposit(a)
w=int(input("ENTER AMOUNT TO WITHDRAW:"))
b.withdraw(w)
```



## Output:



The screenshot shows a terminal window titled 'Run: bank'. The command executed is `C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/venv/bank.py`. The program's output is as follows:

```
ENTER AMOUNT TO DEPOSIT: 1000
BALANCE: 5010
ENTER AMOUNT TO WITHDRAW: 3000
BALANCE: 2010
Process finished with exit code 0
```

The IDE interface includes a sidebar with icons for Version Control, Find, Run, TODO, Problems, Python Packages, Python Console, and Terminal. The bottom status bar shows 'Run' as the active tab.

**Program no:34**

**Aim:** Create a class Rectangle with private attributes length and width.

Overload '<' operator to compare the area of 2 rectangles.

**Source Code:**

```
class rectangle:
    def __init__(self,breadth,length):
        self.breadth=breadth
        self.length=length
    def area(self):
        return self.breadth*self.length
a=int(input("Enter the length of rectangle:"))
b=int(input("Enter the breadth of rectangle:"))
obj=rectangle(a,b)
print("Area of rectangle :",obj.area())
c=int(input("Enter the length of 2nd rectangle:"))
d=int(input("Enter the breadth of 2nd rectangle:"))
obj2=rectangle(c,d)
print("Area of rectangle is:",obj2.area())
if obj.area()==obj2.area():
    print("Both equal")
elif obj.area()>obj2.area():
    print("Rectangle 1 is big")
else:
    print("Rectangle 2 is big")
print()
```

## Output:



```
Run: compare
C:\Users\Ayana\Downloads\venv\Scripts\python.exe "C:/Users/Ayana/Downloads/PyCharm Community Edition 2021.2.3/plugins/python-ce/helpers/t
Enter the length of rectangle:20
Enter the breadth of rectangle:10
Area of rectangle : 200
Enter the length of 2nd rectangle:30
Enter the breadth of 2nd rectangle:41
Area of rectangle is: 1230
Rectangle 2 is big

Process finished with exit code 0
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download ... (2 minutes ago) 10:40 CRLF UTF-8 4 spaces Pyt

---

**Program no:35**

**Aim:** Create a class Time with private attributes hour, minute and second.

Overload '+' operator to find sum of 2 time.

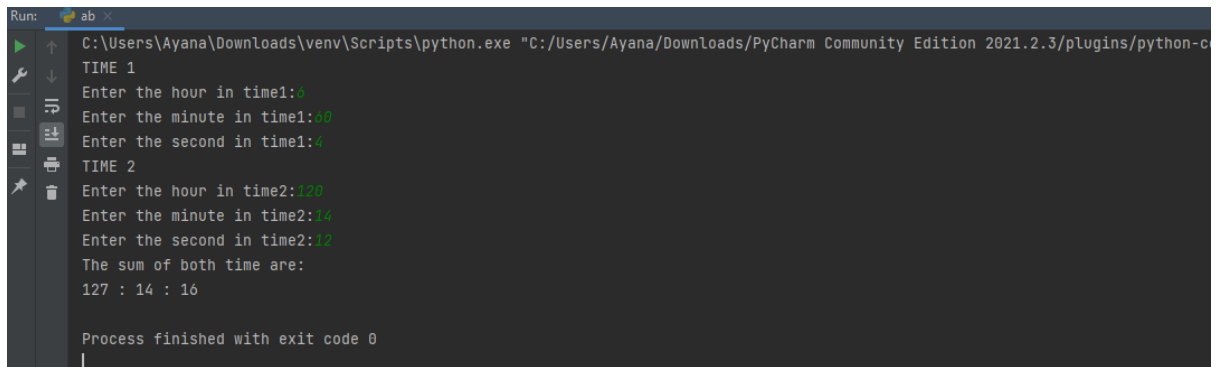
**Source Code:**

```
class Time:
    def __init__(self, h, m, s):
        self._h1 = h
        self._m1 = m
        self._s1 = s

    def __add__(self, x):
        sum1 = self._h1 + x._h1
        sum2 = self._m1 + x._m1
        sum3 = self._s1 + x._s1
        if sum3 >= 60:
            sum3 = sum3 - 60
            sum2 = sum2 + 1
        if sum2 >= 60:
            sum2 = sum2 - 60
            sum1 = sum1 + 1
        print(sum1, ":", sum2, ":", sum3);

print("TIME 1")
h1 = int(input("Enter the hour in time1:"))
m1 = int(input("Enter the minute in time1:"))
s1 = int(input("Enter the second in time1:"))
obj1 = Time(h1, m1, s1)
print("TIME 2")
h2 = int(input("Enter the hour in time2:"))
m2 = int(input("Enter the minute in time2:"))
s2 = int(input("Enter the second in time2:"))
obj2 = Time(h2, m2, s2)
print("The sum of both time are:")
obj1 + obj2
```

---

**Output:**

The screenshot shows a PyCharm console window with the following output:

```
Run: ab x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe "C:/Users/Ayana/Downloads/PyCharm Community Edition 2021.2.3/plugins/python-c
TIME 1
Enter the hour in time1:1
Enter the minute in time1:40
Enter the second in time1:1
TIME 2
Enter the hour in time2:127
Enter the minute in time2:14
Enter the second in time2:16
The sum of both time are:
127 : 14 : 16

Process finished with exit code 0
```

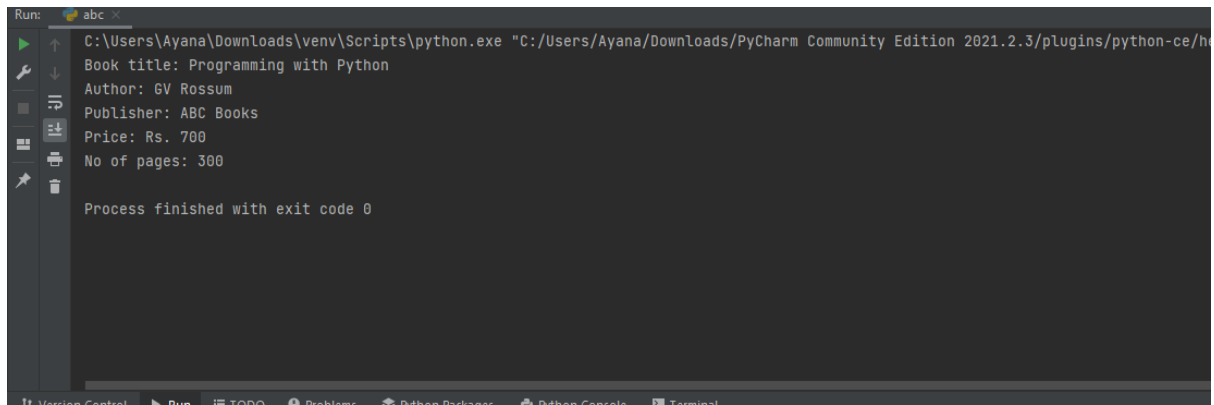
**Program no:36**

**Aim:** Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no\_of\_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

**Source Code:**

```
class Publisher:
    def __init__(self,name1):
        self.name=name1
    def show(self):
        pass
class Book(Publisher):
    def __init__(self,title1,author1,name1):
        self.title=title1
        self.author=author1
        Publisher.__init__(self,name1)
    def show(self):
        pass
class Python(Book):
    def __init__(self,p,no,title1,author1,name1):
        self.price=p
        self.no_of_pages=no
        Book.__init__(self,title1,author1,name1)
    def show(self):
        print('Book title:',self.title)
        print('Author:',self.author)
        print('Publisher:',self.name)
        print('Price: Rs.',self.price)
        print('No of pages:',self.no_of_pages)
P1=Python(700,300,'Programming with Python','GV Rossum','ABC Books')
P1.show()
```

---

**Output:**

The screenshot shows a PyCharm Run console window. The top bar indicates the run configuration is 'abc'. The console output is as follows:

```
C:\Users\Ayana\Downloads\venv\Scripts\python.exe "C:/Users/Ayana/Downloads/PyCharm Community Edition 2021.2.3/plugins/python-ce/h  
Book title: Programming with Python  
Author: GV Rossum  
Publisher: ABC Books  
Price: Rs. 700  
No of pages: 300  
  
Process finished with exit code 0
```

The bottom of the window shows the standard PyCharm interface with tabs for Version Control, Run, TODO, Problems, Other Packages, Other Console, and Terminal.

---

**Program no:37**

**Aim:** Write a Python program to read a file line by line and store it into a list.

**Source Code:****demo.txt**

Python

Interpreted high-level language.

Python is object oriented programming language

**line.py**

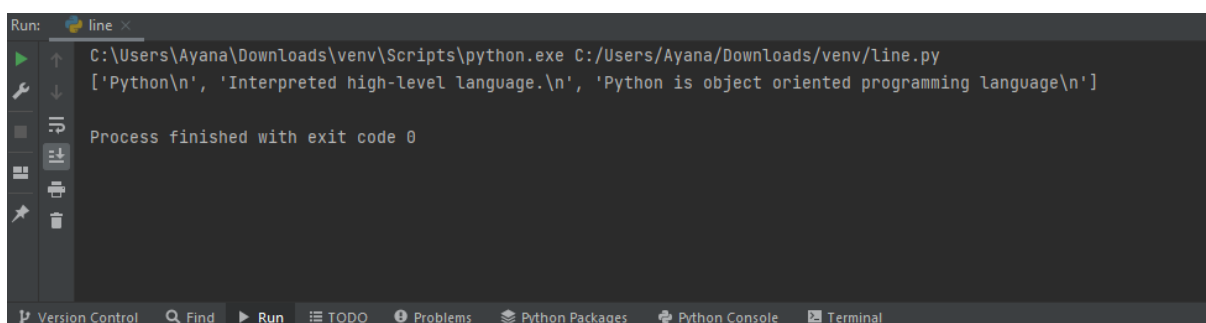
```
def fread(fname):
```

```
    with open(fname) as f:
```

```
        c = f.readlines()
```

```
    print(c)
```

```
fread("demo.txt")
```

**Output:**

```
Run: line x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:\Users\Ayana\Downloads\venv\line.py
['Python\n', 'Interpreted high-level language.\n', 'Python is object oriented programming language\n']
Process finished with exit code 0
```



**Program no:38**

**Aim:** Python program to copy odd lines of one file to other

**Source Code:****demo.txt**

Python

Interpreted high-level language.

Python is object oriented programming language

**line.py**

```
a = open("demo.txt", "r")
```

```
b = open("t", "w")
```

```
c = a.readlines()
```

```
d = len(c)
```

```
for i in range(0, d):
```

```
    if i % 2 == 0:
```

```
        b.write(c[i])
```

```
    else:
```

```
        pass
```

```
b.close()
```

```
b = open("t", "r")
```

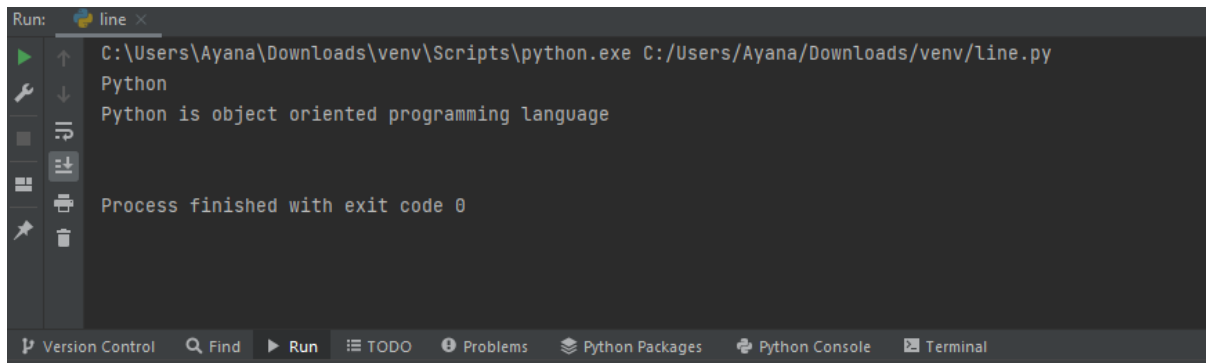
```
e = b.read()
```

```
print(e)
```

```
a.close()
```

```
b.close()
```

---

**Output:**

The screenshot shows a terminal window titled 'Run: line x'. The command executed is `C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/venv/line.py`. The output consists of two lines: `Python` and `Python is object oriented programming language`. Below the output, it states `Process finished with exit code 0`. The terminal interface includes a left sidebar with icons for file operations and a bottom status bar with tabs for Version Control, Find, Run, TODO, Problems, Python Packages, Python Console, and Terminal.

```
Run: line x
C:\Users\Ayana\Downloads\venv\Scripts\python.exe C:/Users/Ayana/Downloads/venv/line.py
Python
Python is object oriented programming language

Process finished with exit code 0
```

## Program no:39

**Aim:** Write a Python program to read each row from a given csv file and print a list of strings.

### Source Code:

#### CSV

```
Series_reference,Period,Data_value,Suppressed,STATUS,UNITS,Magnitude,Subject,Group,Series_title_1,Series_title_2,Series_1
BDCQ.SF1AA2CA,2016.06,1116.386,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2016.09,1070.874,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2016.12,1054.408,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2017.03,1010.665,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2017.06,1233.7,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2017.09,1282.436,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2017.12,1290.82,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2018.03,1412.007,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2018.06,1488.055,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2018.09,1497.678,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2018.12,1570.507,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2019.03,1393.749,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2019.06,1517.143,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2019.09,1381.514,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2019.12,1370.985,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2020.03,1073.017,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2020.06,1131.445,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2020.09,1440.101,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA3CA,2016.06,1189.735,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),"
BDCQ.SF1AA3CA,2016.09,1144.938,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),"
BDCQ.SF1AA3CA,2016.12,1180.550,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),"
```

#### line.py

```
import csv
```

```
with open("csv", newline="") as csvfile:
```

```
    d = csv.reader(csvfile, delimiter=' ', quotechar='|')
```

```
    for i in d:
```

```
        print(' '.join(i))
```

## Output:

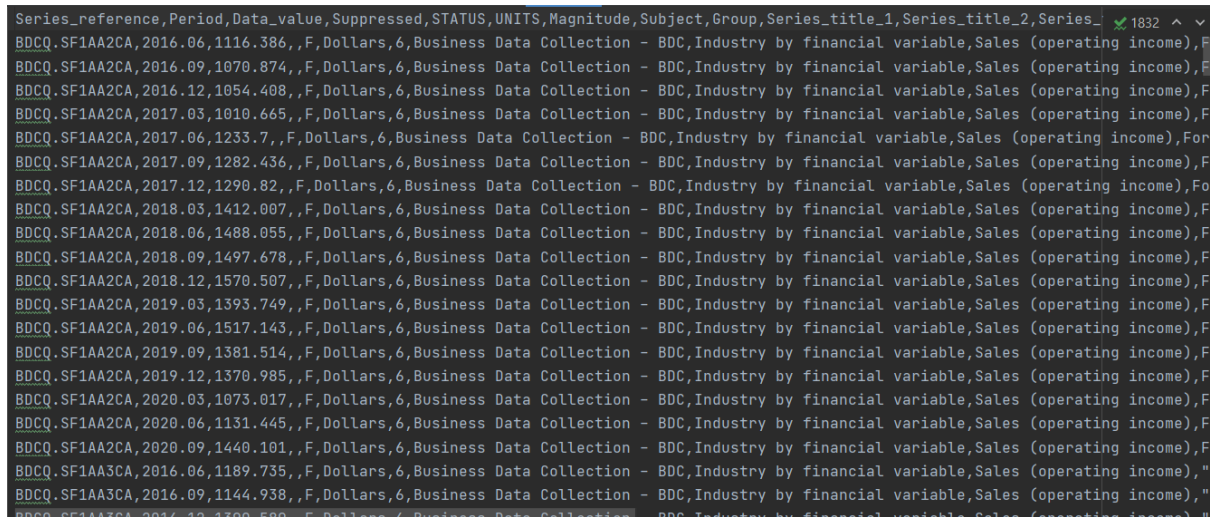
```
"C:\Program Files\Python39\python.exe" "C:/Users/Hp/Desktop/python programs/New folder/28-01-22/File/3.py"
Series_reference,Period,Data_value,Suppressed,STATUS,UNITS,Magnitude,Subject,Group,Series_title_1,Series_title_2,Series_title_3,Series_title_4,Series_title_5
BDCQ.SF1AA2CA,2016.06,1116.386,,F,Dollars,6,Business,Data,Collection,-,BDC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BDCQ.SF1AA2CA,2016.09,1070.874,,F,Dollars,6,Business,Data,Collection,-,BDC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BDCQ.SF1AA2CA,2016.12,1054.408,,F,Dollars,6,Business,Data,Collection,-,BDC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BDCQ.SF1AA2CA,2017.03,1010.665,,F,Dollars,6,Business,Data,Collection,-,BDC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BDCQ.SF1AA2CA,2017.06,1233.7,,F,Dollars,6,Business,Data,Collection,-,BDC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BDCQ.SF1AA2CA,2017.09,1282.436,,F,Dollars,6,Business,Data,Collection,-,BDC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BDCQ.SF1AA2CA,2017.12,1290.82,,F,Dollars,6,Business,Data,Collection,-,BDC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BDCQ.SF1AA2CA,2018.03,1412.007,,F,Dollars,6,Business,Data,Collection,-,BDC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BDCQ.SF1AA2CA,2018.06,1488.055,,F,Dollars,6,Business,Data,Collection,-,BDC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BDCQ.SF1AA2CA,2018.09,1497.678,,F,Dollars,6,Business,Data,Collection,-,BDC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BDCQ.SF1AA2CA,2018.12,1570.507,,F,Dollars,6,Business,Data,Collection,-,BDC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BDCQ.SF1AA2CA,2019.03,1393.749,,F,Dollars,6,Business,Data,Collection,-,BDC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BDCQ.SF1AA2CA,2019.06,1517.143,,F,Dollars,6,Business,Data,Collection,-,BDC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BDCQ.SF1AA2CA,2019.09,1381.514,,F,Dollars,6,Business,Data,Collection,-,BDC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BDCQ.SF1AA2CA,2019.12,1370.285,,F,Dollars,6,Business,Data,Collection,-,BDC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
```

## Program no:40

**Aim:** Write a Python program to read specific columns of a given CSV file and print the content of the columns.

### Source Code:

#### csv



#### line.py

```
import csv
```

```
with open("csv", newline="") as csvfile:
```

```
    d = csv.DictReader(csvfile)
```

```
    print("Period    Subject")
```

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

```
    for i in d:
```

```
        print(i['Period'], i['Subject'])
```

---

**Output:**

```
Period      Subject
-----
2016.06 Business Data Collection - BDC
2016.09 Business Data Collection - BDC
2016.12 Business Data Collection - BDC
2017.03 Business Data Collection - BDC
2017.06 Business Data Collection - BDC
2017.09 Business Data Collection - BDC
2017.12 Business Data Collection - BDC
2018.03 Business Data Collection - BDC
2018.06 Business Data Collection - BDC
```

## Program no:41

**Aim:** Write a Python program to write a Python dictionary to a csv file. After writing the CSV file read the CSV file and display the content.

### Source Code:

#### csv

```
Series_reference,Period,Data_value,Suppressed,STATUS,UNITS,Magnitude,Subject,Group,Series_title_1,Series_title_2,Series_1
BDCQ.SF1AA2CA,2016.06,1116.386,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2016.09,1070.874,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2016.12,1054.408,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2017.03,1010.665,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2017.06,1233.7,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2017.09,1282.436,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2017.12,1290.82,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2018.03,1412.007,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2018.06,1488.055,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2018.09,1497.678,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2018.12,1570.507,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2019.03,1393.749,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2019.06,1517.143,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2019.09,1381.514,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2019.12,1370.985,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2020.03,1073.017,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2020.06,1131.445,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA2CA,2020.09,1440.101,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA3CA,2016.06,1189.735,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA3CA,2016.09,1144.938,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
BDCQ.SF1AA3CA,2016.12,1180.550,,F,Dollars,6,Business Data Collection - BDC,Industry by financial variable,Sales (operating income),F
```

#### line.py

```
import csv
```

```
field_name = ['No', 'Company', 'Car Model']
```

```
car = [{'No': 1, 'Company': 'Ferrari', 'Car Model': 'GH'},
```

```
      {'No': 2, 'Company': 'BMW', 'Car Model': 'X5'},
```

```
      {'No': 3, 'Company': 'Maruti Suzuki', 'Car Model': 'Swift'},
```

```
      {'No': 4, 'Company': 'Audi', 'Car Model': 'RS7'},
```

```
      {'No': 5, 'Company': 'Toyota', 'Car Model': 'Fortuner'}]
```

```
with open('b.csv', 'w') as csvfile:
```

```
    write = csv.DictWriter(csvfile, fieldnames=field_name)
```

```
    write.writeheader()
```

---

```
write.writerows(car)
```

```
with open('b.csv', newline='') as csvfile:
```

```
    d = csv.reader(csvfile, delimiter=',')
```

```
    for r in d:
```

```
        print(','.join(r))
```

### Output:

```
No,Company,Car Model
```

```
1,Ferrari,GH
```

```
2,BMW,X5
```

```
3,Maruti Suzuki,Swift
```

```
4,Audi,RS7
```

```
5,Toyota,Fortuner
```

```
Process finished with exit code 0
```