CSE1021 – Introduction to Problem Solving and Programming

LAB MANUAL

Class No. BL2023240500384

Slot: D11+D12+D13

By

Name: Devam Pandey

Regd No: 23BCE10731

Course Instructor

Dr Mayuri AVR

Associate Professor

School of Computing Science and Engineering
VIT Bhopal University, Sehore
Madhya Pradesh

Table of Contents

Problem Solving – Algorithm and Flowchart using Raptor Tool

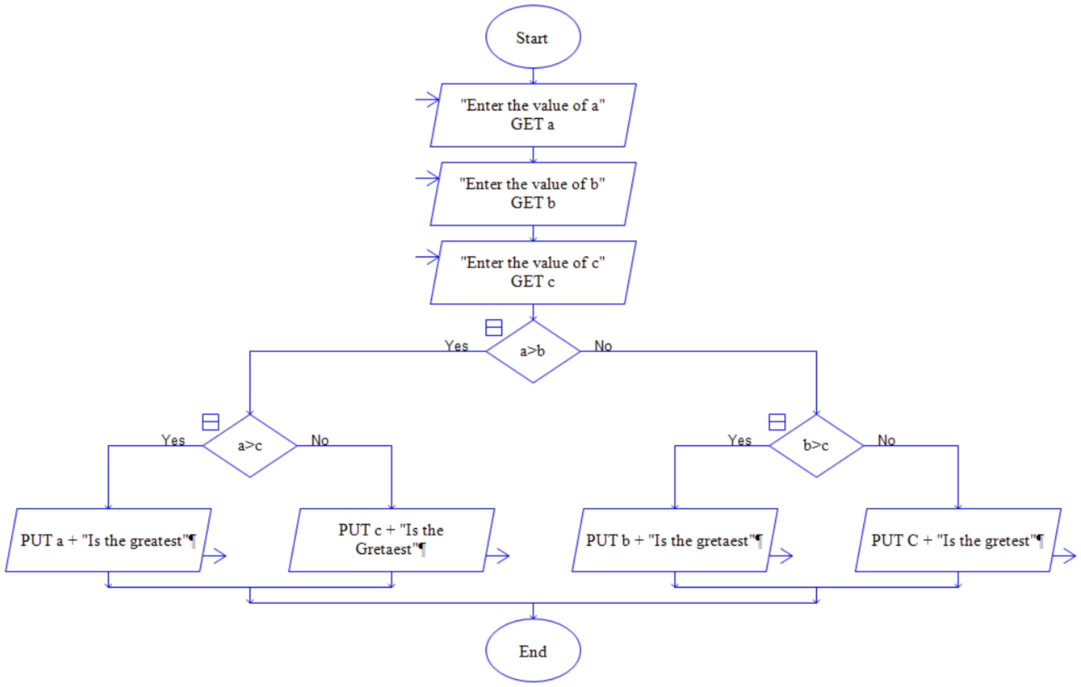
Question No. 1:

Write the algorithm and draw the flowchart to find the maximum of three numbers.

Algorithm:

Step 1: Start
Step 2: Take 3 numbers. (a,b,c)
Step 3: Check a is greater than b
Step 4: if True check go to step 5 else go to step 7
Step 5: check if a is greater than c
Step 6: if true go to Print (a is greater) Else print (C is greater)
Step 7: Check if b is greater than c
Step 8: If true print (B is greater) else print (C is greater)
Step 9: End

Flowchart:



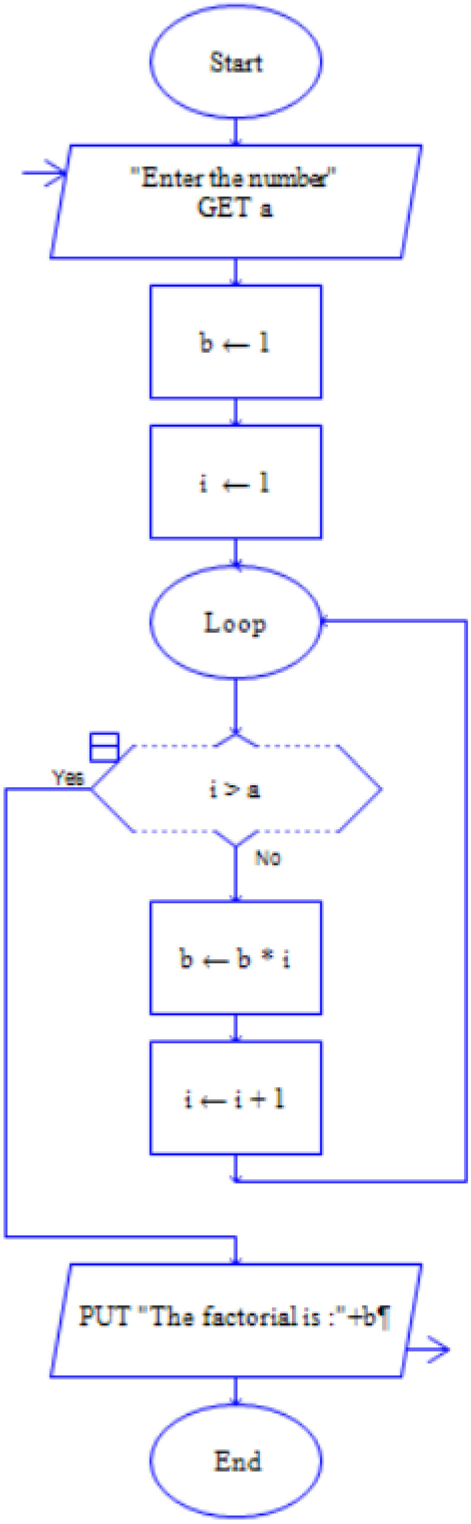
Question 2:

`Write the algorithm and draw the flowchart to find the factorial of the given number`

Algorithm

Step 1 : Start
Step 2: Take an Input (a)
Step 3:Run a loop from 'a' times from 1 to a
Step 4: Multiply all the numbers
Step 5: Print the number
Step 6: End

Flowchart


Start
"Enter the number"
GET a
b ← 1
i ← 1
Loop
Yes
i > a
No
b ← b * i
i ← i + 1
PUT "The factorial is :"+b¶
End

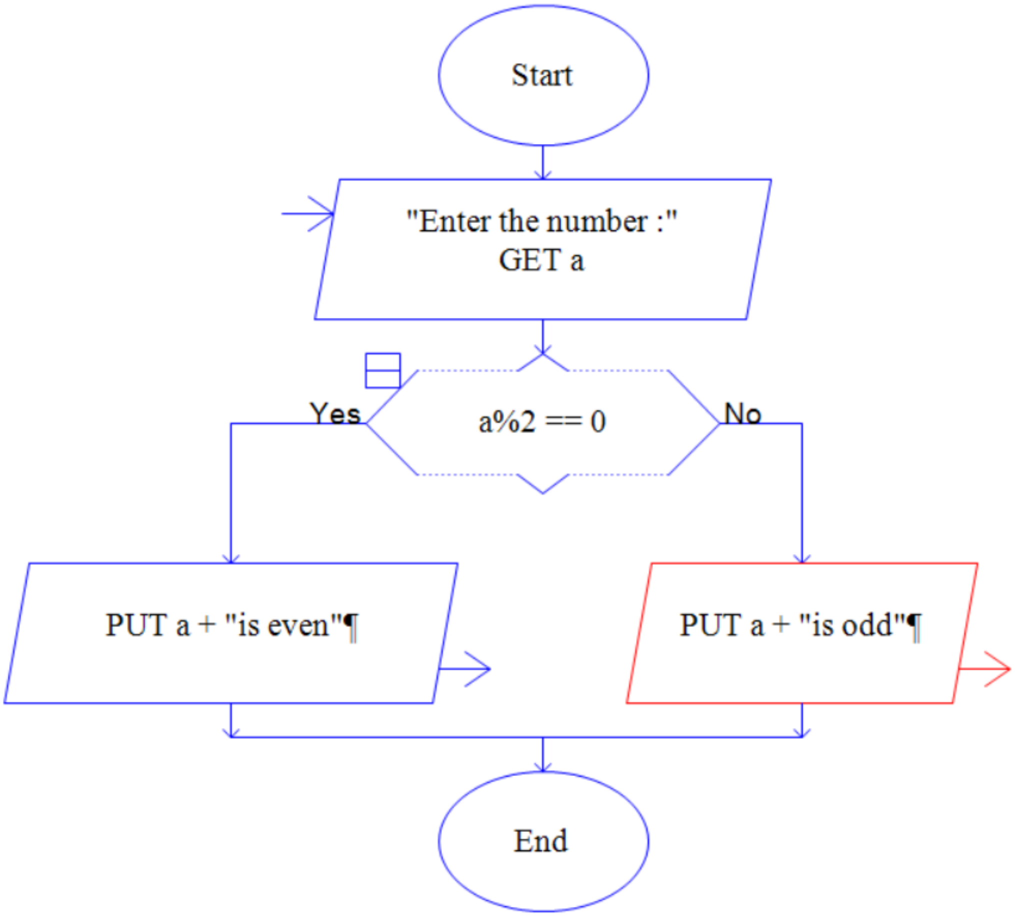
Question 3:

`Write the algorithm and draw the flowchart to find whether the given number is odd or even`

Algorithm:

Step 1 : Start
Step 2 :take an integer as an input (a)
Step 3 : check if a divided by 2 gives remainder 0
Step 4 : If true print a is even else print a is odd
Step 5 : End

Flowchart



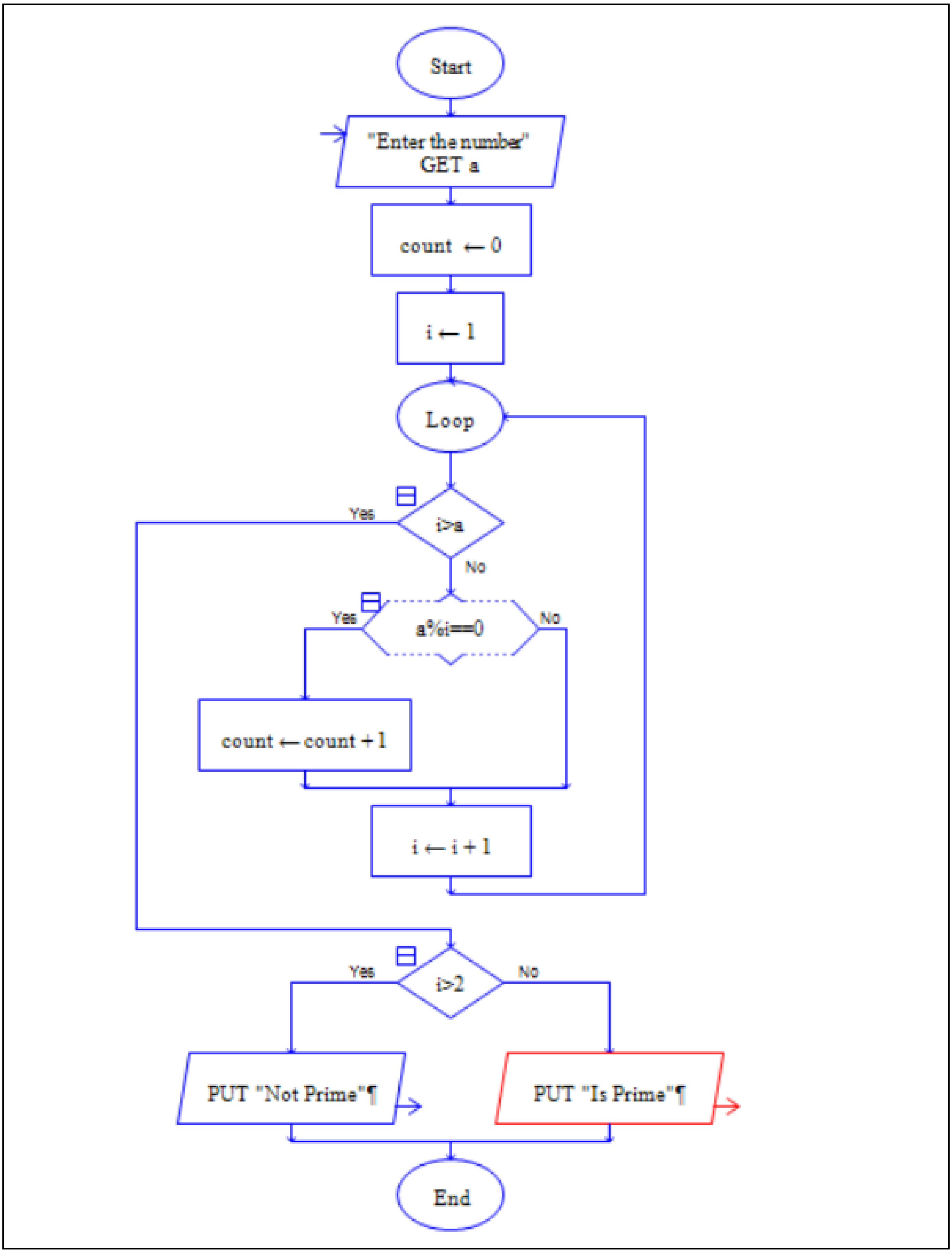
Question 4:

Write the algorithm and draw the flowchart to find whether the given number is prime number or not

Algorithm

Step 1: Start
Step 2:Take input (a)
Step 3: Run a loop a times from 1 to a
Step 4: Divide a with each number and count every time the remainder is 0
Step 5: if the counted value is more than 2 then its a prime number else it is not
Step 6: End

FlowChart


Start
"Enter the number"
GET a
count ← 0
i ← 1
Loop
i>a
Yes
No
a%i==0
Yes
No
count ← count + 1
i ← i + 1
i>2
Yes
No
PUT "Not Prime"¶
PUT "Is Prime"¶
End

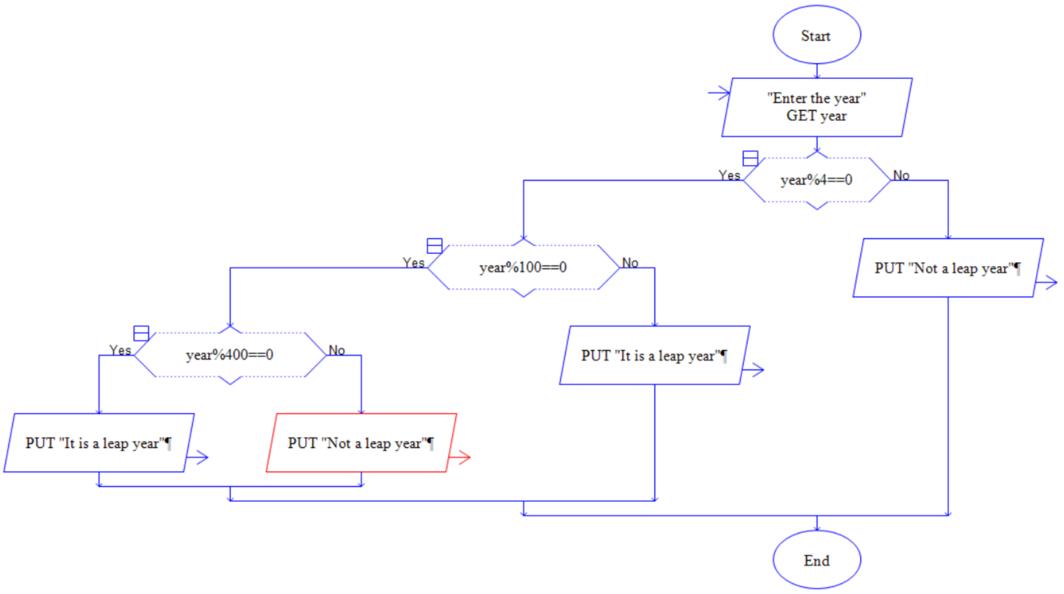
Question 5:

Write the algorithm and draw the flowchart to find the given year is leap year or not.

Algorithm

Step 1: Start
Step 2: Take an input
Step 3: check if dividing it with 4 gives a remainder 0
Step 4: if true go to step 5 else print not a leap year
Step 5: check if dividing it with 100 gives remainder 0
Step 6:if true go to step 7 else print it is a leap year
Step 7: check if dividing it by 400 gives a remainder 0
Step 8: if true it is a leap year else it is not
Step 9: End

Flowchart



Programs on Basics of Python Programming

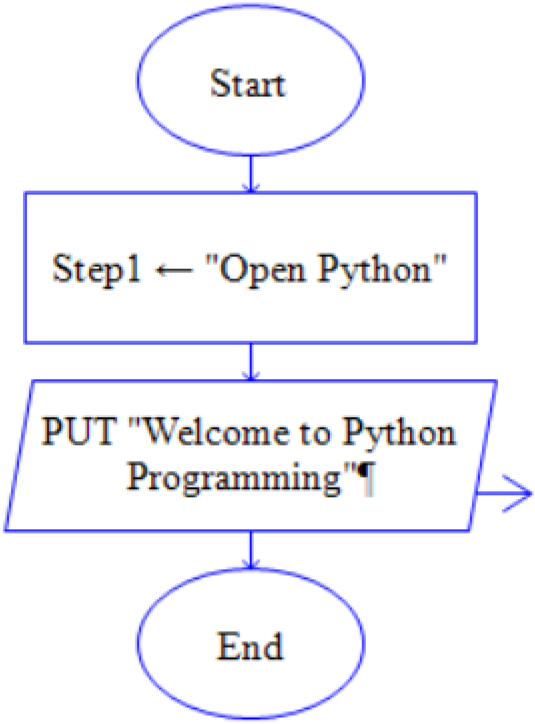
Question no. 1:

`Write a python program to print "Welcome to Python Programming"`

Algorithm :

Step 1: start
Step 2: Open any Python development environment
step 3: use the print command and input "Welcome to Python Programming"
Step 4: end

Flowchart :



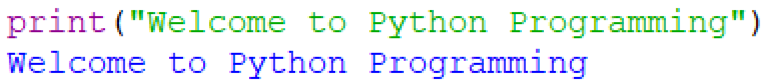
Source Code :

```
print("Welcome to Python Programming")
```

Sample input and output :

```
print("Welcome to Python Programming")
Welcome to Python Programming
```

Screenshot of Execution:



```
print("Welcome to Python Programming")
Welcome to Python Programming
```

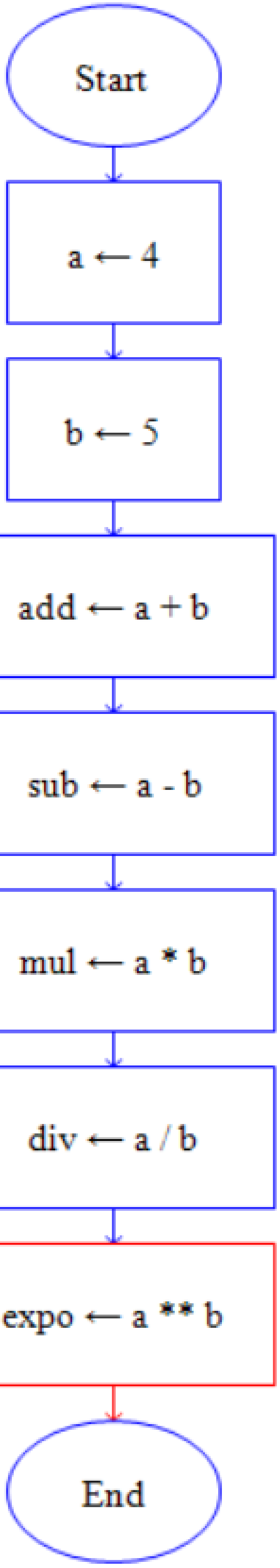
Question no. 2:

Write a python program to perform arithmetic operations

Algorithm :

```
Step1: Start
Step2: Take 2 numbers
Step3: Perform various Arithmetic OPerations
Step4: Print the result
Step5 : End
```

Flowchart :



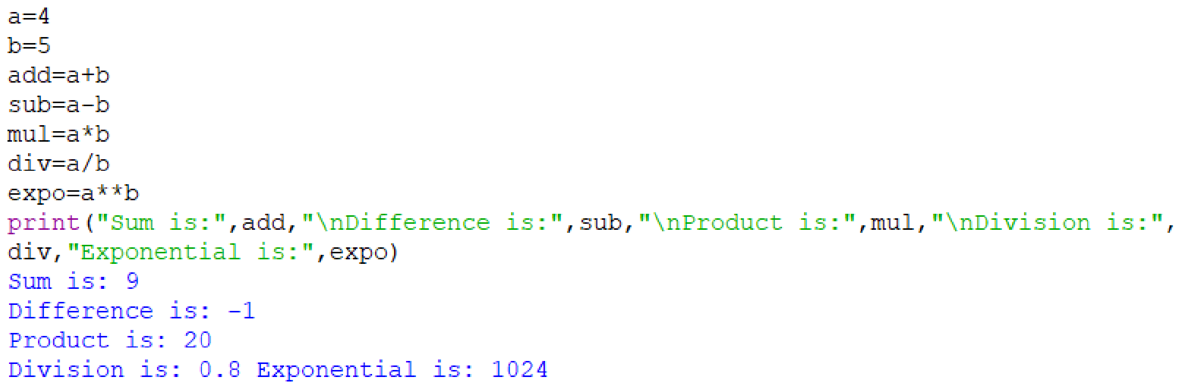
Source Code :

```
a=4
b=5
add=a+b
sub=a-b
mul=a*b
div=a/b
expo=a**b
print("Sum is:",add,"\nDifference is:",sub,"\nProduct is:",mul,"\nDivision is:",div,"Exponential is:",expo)
```

Sample input and output :

```
a=4
b=5
add=a+b
sub=a-b
mul=a*b
div=a/b
expo=a**b
print("Sum is:",add,"\nDifference is:",sub,"\nProduct is:",mul,"\nDivision is:",div,"Exponential is:",expo)
Sum is: 9
Difference is: -1
Product is: 20
Division is: 0.8 Exponential is: 1024
```

Screenshot of Execution:



```
a=4
b=5
add=a+b
sub=a-b
mul=a*b
div=a/b
expo=a**b
print("Sum is:",add,"\nDifference is:",sub,"\nProduct is:",mul,"\nDivision is:",
div,"Exponential is:",expo)
Sum is: 9
Difference is: -1
Product is: 20
Division is: 0.8 Exponential is: 1024
```

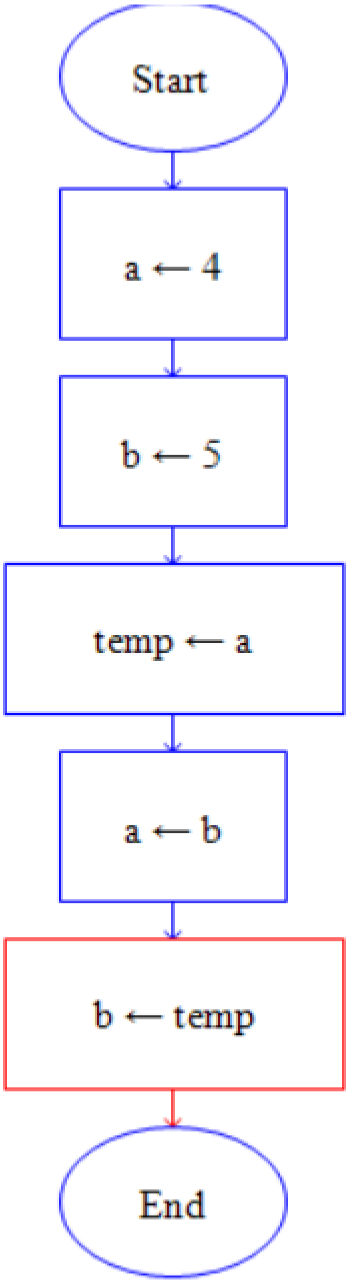
Question no. 3:

Write a python program to swap two numbers using temporary variable

Algorithm :

Step1: Start
Step2: Take 2 numbers (a,b)
Step3: Take a temporary variable (temp)
Step4: Assign 1 of the values to the temporary variable
Step5: Assign the other value to the first variable
Step6: NOw assign the value of the temporary variable to the second variable
Step7: End

Flowchart :



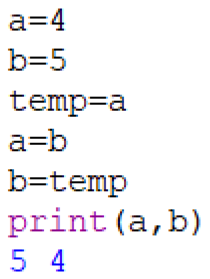
Source Code :

```
a=4
b=5
temp=a
a=b
b=temp
print(a,b)
```

Sample input and output :

```
a=4
b=5
temp=a
a=b
b=temp
print(a,b)
5 4
```

Screenshot of Execution:



```
a=4
b=5
temp=a
a=b
b=temp
print(a,b)
5 4
```

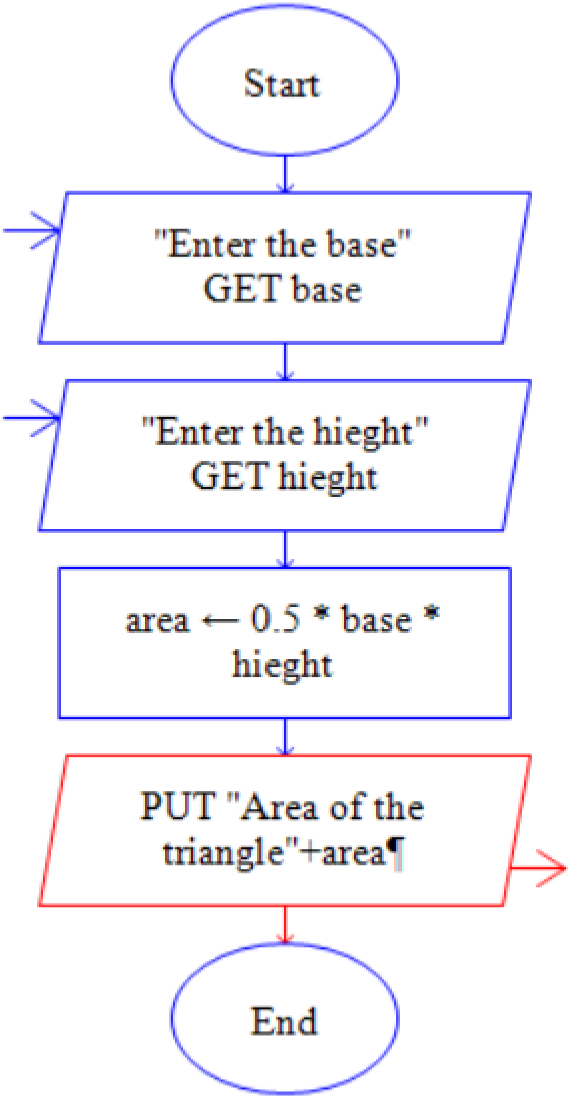
Question no. 4:

`Write a python program to print area of a triangle`

Algorithm :

```
Step1: Start
Step2: Input the base of the triangle(base)
Step3: input the height of the triangle (height)
Step4: Apply the formula (0.5*area*height)
Step5: Print the output
Step6: End
```

Flowchart :



Source Code :

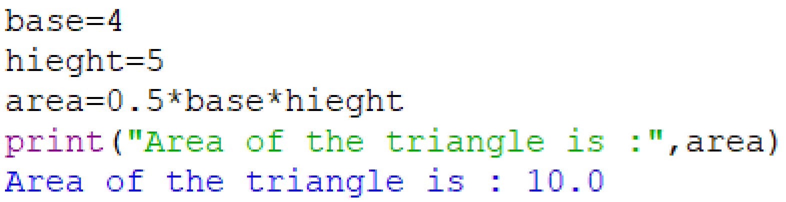
```
base=4
hieght=5
area=0.5*base*hieght
print("Area of the triangle is :",area)
```

Sample input and output :

```
base=4
hieght=5
area=0.5*base*hieght
print("Area of the triangle is :",area)
```

Area of the triangle is : 10.0

Screenshot of Execution:



```
base=4
hieght=5
area=0.5*base*hieght
print("Area of the triangle is :",area)
Area of the triangle is : 10.0
```

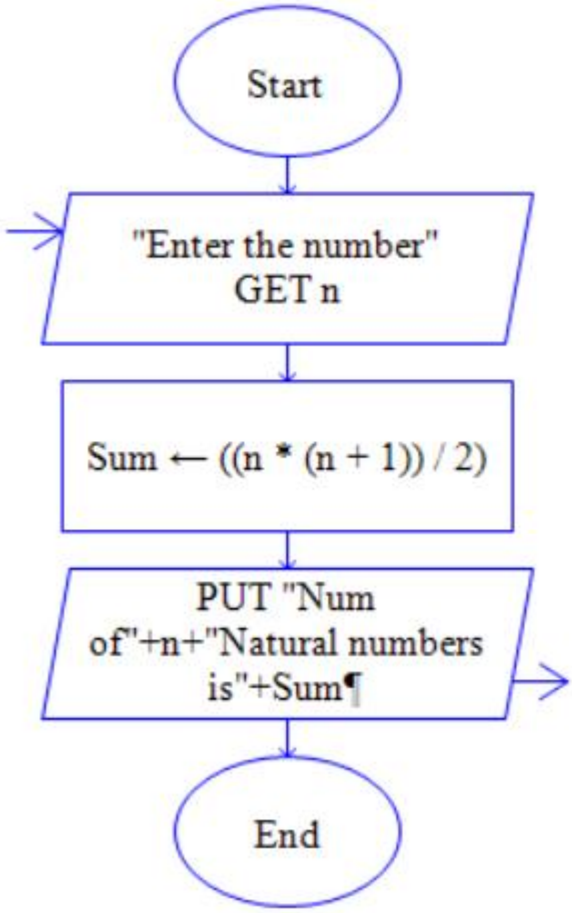
Question no. 5:

```
Write a python program to find the sum of n natural numbers using
mathematical formula (n*(n+1))/2
```

Algorithm :

step1: Start
Step2: Input the number till where we need the sum(n)
Step3: Apply the Formula((n*(n+1))/2)

Flowchart :


Start
"Enter the number"
GET n
Sum ← ((n * (n + 1)) / 2)
PUT "Num of"+n+"Natural numbers is"+Sum¶
End

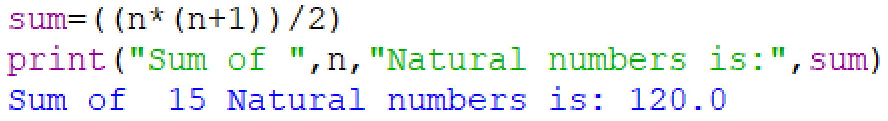
Source Code :

```
n=15
sum=((n*(n+1))/2)
print("Sum of ",n,"Natural numbers is:",sum)
```

Sample input and output :

```
n=15
sum=((n*(n+1))/2)
print("Sum of ",n,"Natural numbers is:",sum)
Sum of  15 Natural numbers is: 120.0
```

Screenshot of Execution:



```
sum=((n*(n+1))/2)
print("Sum of ",n,"Natural numbers is:",sum)
Sum of  15 Natural numbers is: 120.0
```

Python Programs on Conditional Control Structures

Question No 1:

A school has following rules for grading system:

a. Below 25 - F

b. 25 to 45 - E

c. 45 to 50 - D

d. 50 to 60 - C

e. 60 to 80 - B

f. Above 80 – A.

Ask user to enter marks and print the corresponding grade.

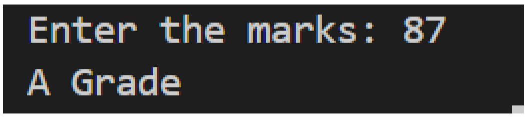
Source Code:

```
a = int(input("Enter the marks: "))
if a <= 25:
   print ("F Grade")
elif a <= 45:
   print ("E Grade")
elif a <= 50:
   print ("D Grade")
elif a <= 60:
   print ("C Grade")
elif a <= 80:
   print ("B Grade")
elif a > 80:
   print ("A Grade")
else:
   print ("Invalid Input")
```

Sample Input and Output:

```
Enter the marks: 87
A Grade
```

Screenshot of the Output:



```
Enter the marks: 87
A Grade
```

Question No 2:

Write a Python program that checks if a given year is a leap year or not. A leap year is divisible by 4, except for years divisible by 100 but not divisible by 400.

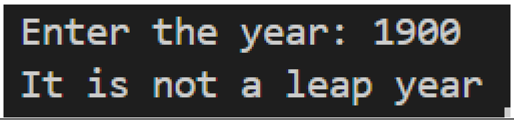
Source Code:

```
year = int(input("Enter the year: "))
if year % 4 == 0:
    if year % 100 == 0:
        if year % 400 == 0:
            print ("It is a leap year")
        else:
            print ("It is not a leap year")
    else:
        print ("It is a leap year")
else:
    print ("It is not a leap year")
```

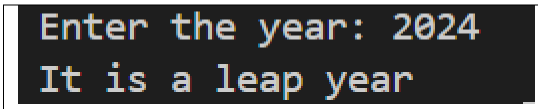
Sample Input and Output:

```
Enter the year: 1900
It is not a leap year
Enter the year: 2024
It is a leap year
```

Screenshot of the Output:



```
Enter the year: 1900
It is not a leap year
```



```
Enter the year: 2024
It is a leap year
```

Question No 3:

Accept three sides of a triangle and check whether the triangle is possible or not. (Hint: Triangle is possible only when sum of any two sides is greater than 3rd side).

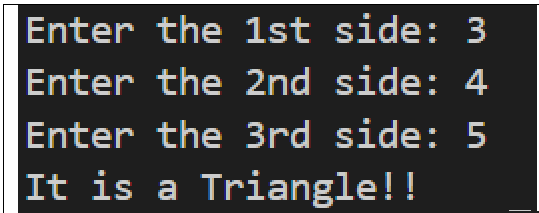
Source Code:

```
a = int(input("Enter the 1st side: "))
b = int(input("Enter the 2nd side: "))
c = int(input("Enter the 3rd side: "))
if a+b > c:
   if a+c > b:
      if b+c > a:
         print ("It is a Triangle!!")
      else:
         print ("It is not a Triangle!!")
   else:
      print ("It is not a Triangle!!")
else:
   print ("It is not a Triangle!!")
```

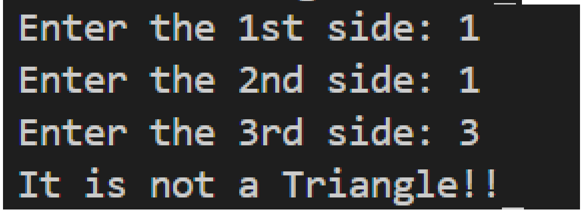
Sample Input and Output:

```
Enter the 1st side: 3
Enter the 2nd side: 4
Enter the 3rd side: 5
It is a Triangle!!

Enter the 1st side: 1
Enter the 2nd side: 1
Enter the 3rd side: 3
It is not a Triangle!!
```

Screenshot of the Output:



```
Enter the 1st side: 3
Enter the 2nd side: 4
Enter the 3rd side: 5
It is a Triangle!!
```

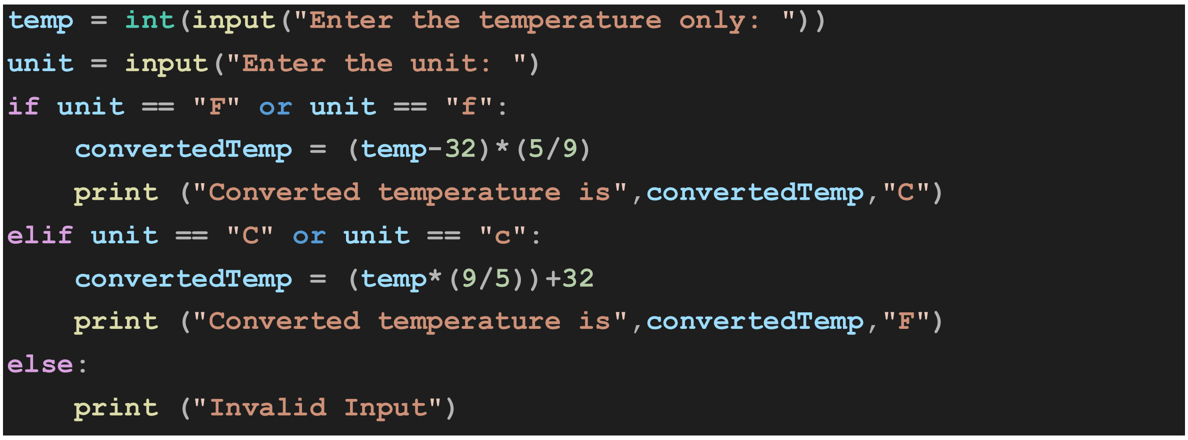


```
Enter the 1st side: 1
Enter the 2nd side: 1
Enter the 3rd side: 3
It is not a Triangle!!
```

Question No 4:

Write a Python program that converts temperatures between Celsius and Fahrenheit. The user should input the temperature and its unit (C or F), and the program should convert it to the other unit.

Source Code:

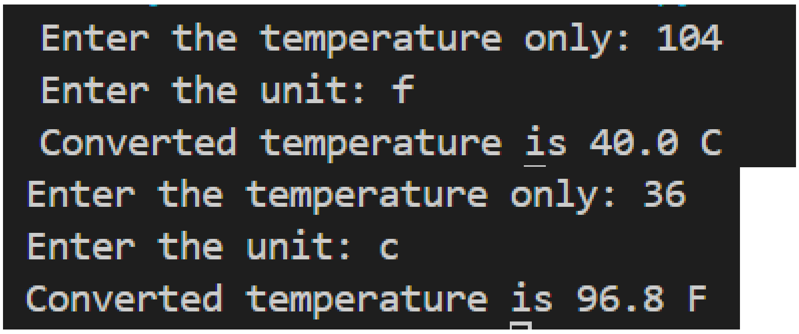


```python
temp = int(input("Enter the temperature only: "))
unit = input("Enter the unit: ")
if unit == "F" or unit == "f":
    convertedTemp = (temp-32)*(5/9)
    print ("Converted temperature is",convertedTemp,"C")
elif unit == "C" or unit == "c":
    convertedTemp = (temp*(9/5))+32
    print ("Converted temperature is",convertedTemp,"F")
else:
    print ("Invalid Input")
```

Sample Input and Output:

```
Enter the temperature only: 104
Enter the unit: f
Converted temperature is 40.0 C

Enter the temperature only: 36
Enter the unit: c
Converted temperature is 96.8 F
```

Screenshot of the Output:



```
Enter the temperature only: 104
Enter the unit: f
Converted temperature is 40.0 C
Enter the temperature only: 36
Enter the unit: c
Converted temperature is 96.8 F
```

Question No 5:

A student will not be allowed to sit in exam if his/her attendance is less than 75%. Take following input from user -Number of classes held, Number of classes attended and print percentage of class attended. Allow the student to sit for the exam if he/she has medical case. Ask user if he/she has medical cause or not (‘Y’ or ‘N’) and print accordingly. Finally Print whether the student is allowed to sit in exam or not.

Source Code:

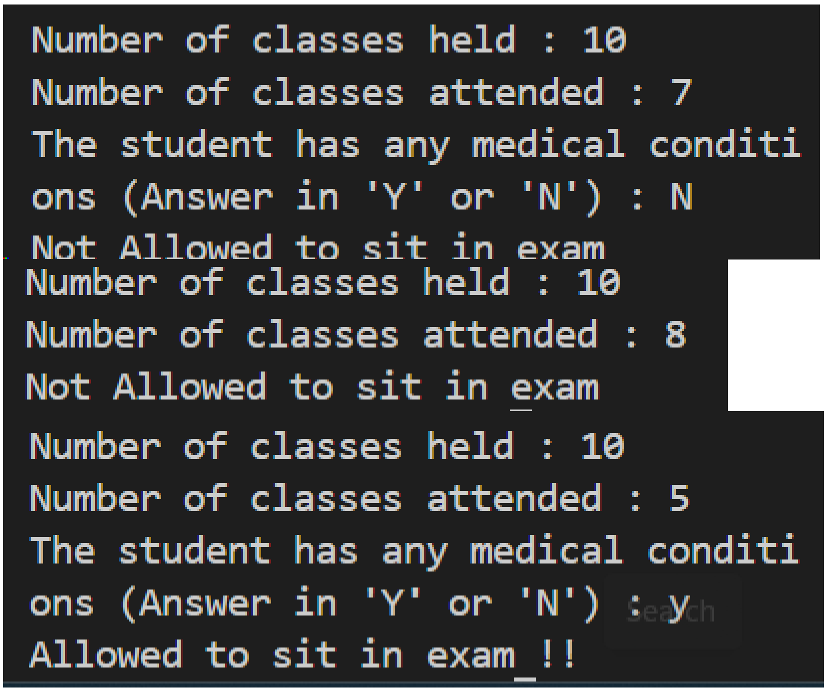
```
clHeld = int(input("Number of classes held : "))
clAttended = int(input("Number of classes attended : "))
attendance = (clAttended/clHeld)*100
if attendance < 75:
   medCondition = input("The student has any medical conditions (Answer in 'Y' or 'N') : 
")
   if medCondition == "Y" or medCondition == "y":
      print ("Allowed to sit in exam !!")
   elif medCondition == "N" or medCondition == "n":
      print ("Not Allowed to sit in exam")
else:
   print ("Not Allowed to sit in exam")
```

Sample Input and Output:

```
Number of classes held : 10
Number of classes attended : 7
The student has any medical conditions (Answer in 'Y' or 'N') : N
```

```
Not Allowed to sit in exam

Number of classes held : 10
Number of classes attended : 8
Not Allowed to sit in exam

Number of classes held : 10
Number of classes attended : 5
The student has any medical conditions (Answer in 'Y' or 'N') : y
Allowed to sit in exam !!
```

Screenshot of the Output:



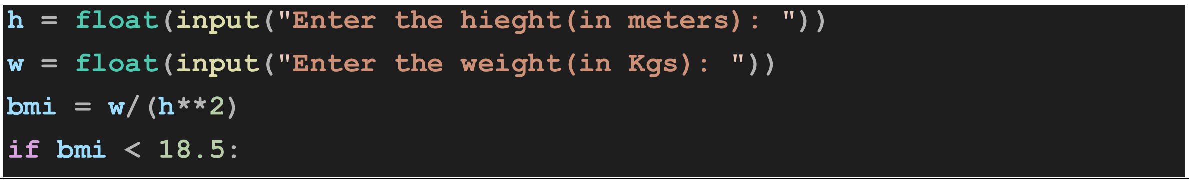
```
Number of classes held : 10
Number of classes attended : 7
The student has any medical conditi
ons (Answer in 'Y' or 'N') : N
Not Allowed to sit in exam
Number of classes held : 10
Number of classes attended : 8
Not Allowed to sit in exam
Number of classes held : 10
Number of classes attended : 5
The student has any medical conditi
ons (Answer in 'Y' or 'N') : y
Allowed to sit in exam !!
```

Question No 6:

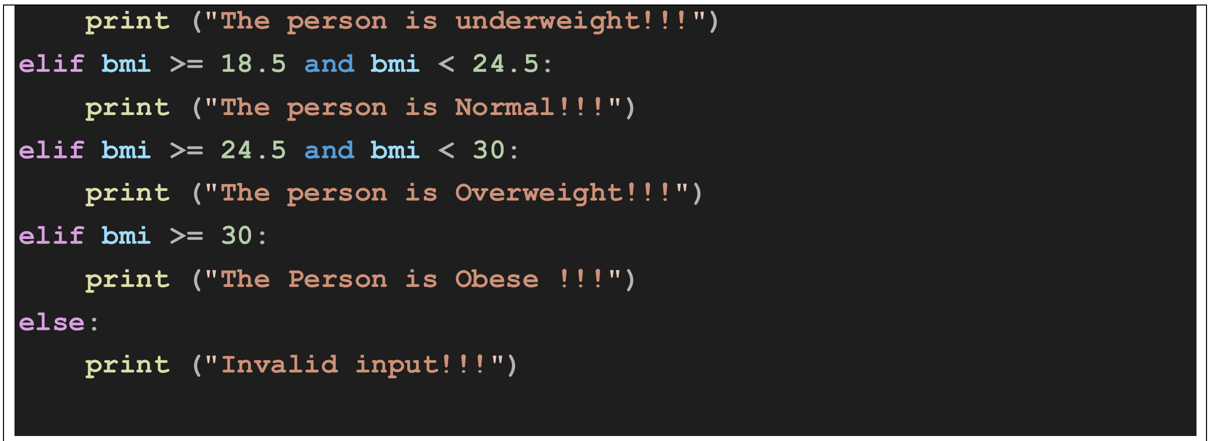
Create a Python program that calculates and categorizes a person's Body

Mass Index (BMI) based on their height and weight.

Source Code:



```python
h = float(input("Enter the hieght(in meters): "))
w = float(input("Enter the weight(in Kgs): "))
bmi = w/(h**2)
if bmi < 18.5:
```

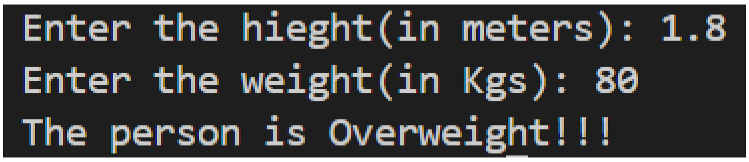


```
    print ("The person is underweight!!!")
elif bmi >= 18.5 and bmi < 24.5:
    print ("The person is Normal!!!")
elif bmi >= 24.5 and bmi < 30:
    print ("The person is Overweight!!!")
elif bmi >= 30:
    print ("The Person is Obese !!!")
else:
    print ("Invalid input!!!")
```

Sample Input and Output:

```
Enter the hieght(in meters): 1.8
Enter the weight(in Kgs): 80
The person is Overweight!!!
```

Screenshot of the Output:



```
Enter the hieght(in meters): 1.8
Enter the weight(in Kgs): 80
The person is Overweight!!!
```

Question No 7:

Create a Python program for a movie theatre that calculates ticket prices based on age and time of day. Tickets for children (age < 12) are $5, adults (age >= 12) are $10, and seniors (age >= 60) are $7. For evening shows (after 5 PM), there's an additional $2 surcharge.

Source Code:

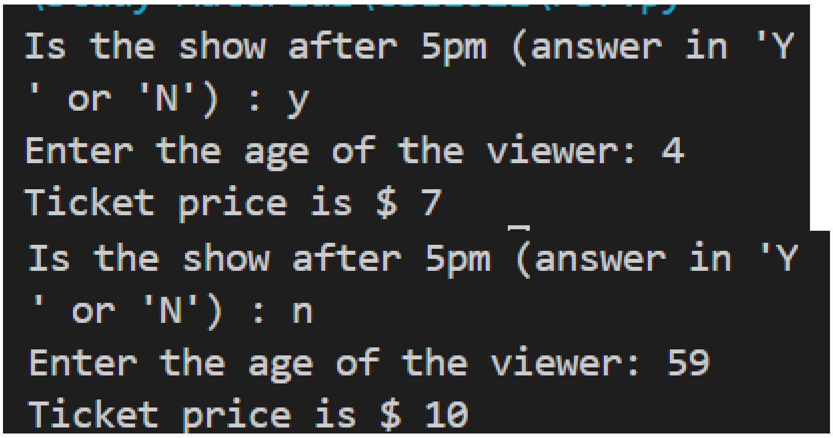
```
time = input("Is the show after 5pm (answer in 'Y' or 'N') : ")
ticketPrice = 0
if time == "y" or time == "Y":
   ticketPrice = 2
age = int(input("Enter the age of the viewer: "))
if age < 12:
   ticketPrice = ticketPrice + 5
```

```
elif age >= 12 and age < 60:
   ticketPrice = ticketPrice + 10
elif age >= 60:
   ticketPrice = ticketPrice + 7
else:
   print ("Invalid Input")
print ("Ticket price is $", ticketPrice)
```

Sample Input and Output:

```
Is the show after 5pm (answer in 'Y' or 'N') : y
Enter the age of the viewer: 4
Ticket price is $ 7

Is the show after 5pm (answer in 'Y' or 'N') : n
Enter the age of the viewer: 59
Ticket price is $ 10
```

Screenshot of the Output:



```
Is the show after 5pm (answer in 'Y
' or 'N') : y
Enter the age of the viewer: 4
Ticket price is $ 7
Is the show after 5pm (answer in 'Y
' or 'N') : n
Enter the age of the viewer: 59
Ticket price is $ 10
```

Question No 8:

A company decided to give bonus of 10% to employee if his/her year of service is more than 3 years. Ask user for their salary and year of service and print the net bonus amount.

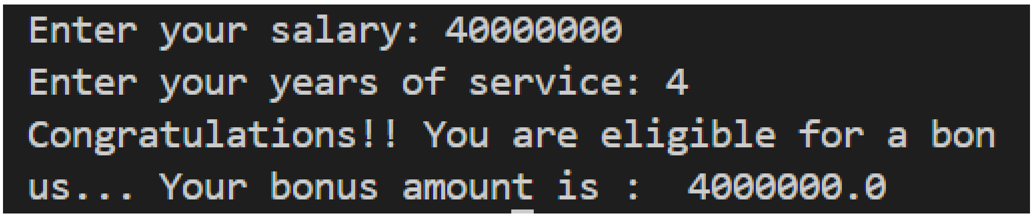
Source Code:

```
salary = int(input("Enter your salary: "))
serviceYears = int(input("Enter your years of service: "))
if serviceYears >3:
   print ("Congratulations!! You are eligible for a bonus... Your bonus amount is : ",
salary/10)
else:
   print ("Sorry you are not eligible for a bonus")
```

Sample Input and Output:

```
Enter your salary: 40000000
Enter your years of service: 4
Congratulations!! You are eligible for a bonus... Your bonus amount is :  4000000.0
```

Screenshot of the Output:



```
Enter your salary: 40000000
Enter your years of service: 4
Congratulations!! You are eligible for a bon
us... Your bonus amount is :  4000000.0
```

Question No 9:

Ask user to enter age, gender (M or F), marital status (Y or N) and then using following rules print their place of service.

If employee is female, then she will work only in urban areas.

If employee is a male and age is in between 20 to 40 then he may work in anywhere.

If employee is male and age is in between 40 to 60 then he will work in urban areas only.

And any other input of age should print "ERROR".

Source Code:

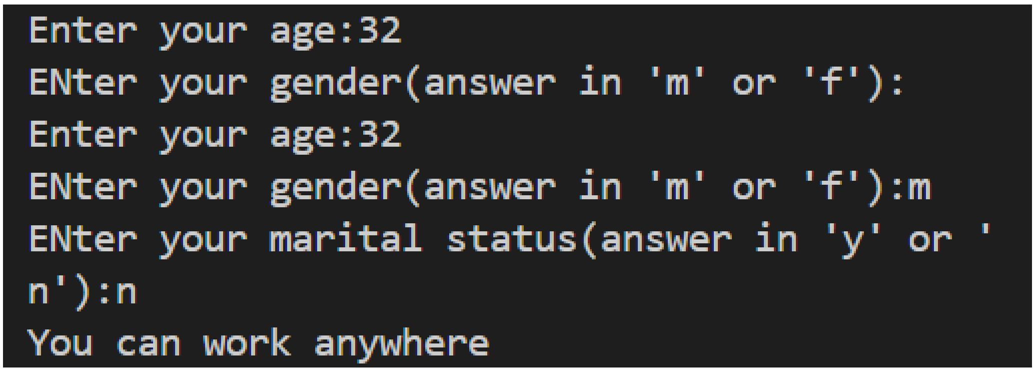
```
age = int(input("Enter your age:"))
gender = input("ENter your gender(answer in 'm' or 'f'):")
marritalStatus = input("ENter your marital status(answer in 'y' or 'n'):")
if gender == "f"or gender == "F":
   print ("You will only  work in urban areas")
```

```
elif gender == "M" or gender == "m":
   if age >= 20 and age <=40:
      print ("You can work anywhere")
   elif age >= 40 and age <=60:
      print ("You can work only in urban areas")
   else:
      print ("ERROR")
else:
   print ("Error: Unknown gender")
```

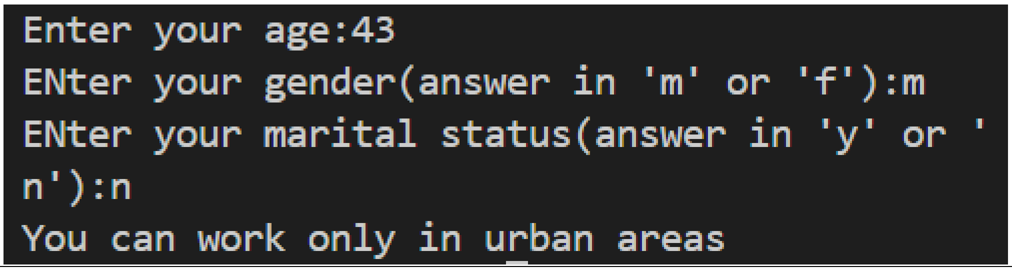
Sample Input and Output:

```
Enter your age:32
ENter your gender(answer in 'm' or 'f'):
Enter your age:32
ENter your gender(answer in 'm' or 'f'):m
ENter your marital status(answer in 'y' or 'n'):n
You can work anywhere

Enter your age:43
ENter your gender(answer in 'm' or 'f'):m
ENter your marital status(answer in 'y' or 'n'):n
You can work only in urban areas
```

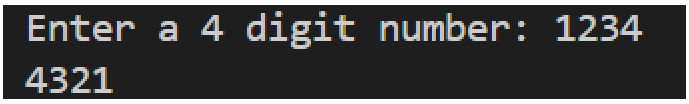
Screenshot of the Output:



```
Enter your age:32
ENter your gender(answer in 'm' or 'f'):
Enter your age:32
ENter your gender(answer in 'm' or 'f'):m
ENter your marital status(answer in 'y' or '
n'):n
You can work anywhere
```



```
Enter your age:43
ENter your gender(answer in 'm' or 'f'):m
ENter your marital status(answer in 'y' or '
n'):n
You can work only in urban areas
```

Question No 10: A 4 digit number is entered through keyboard. Write a program to print a new number with digits reversed as of orignal one. E.g.- INPUT: 1234 OUTPUT: 4321 INPUT: 5982 OUTPUT: 2895
Source Code: num = input("Enter a 4 digit number: ") print (num[::-1])
Sample Input and Output: Enter a 4 digit number: 1234 4321
Screenshot of the Output:  Enter a 4 digit number: 1234 4321

Question No 11:

What will be the output of the following?

```
if i < j:
   if j < k:
      i = j
   else:
      j = k
else:
    if j < k:
        j = i
    else:
        i = k
print(i,j,k)
```

(a) i = 3, j = 5, k = 7

(b) i = -2, j = -5, k = 9

(c) i = 8, j = 15, k = 12

(d) i = 13, j = 15, k = 13

(e) i = 3, j = 5, k = 17

(f) i = 25, j = 15, k = 17

Source Code:

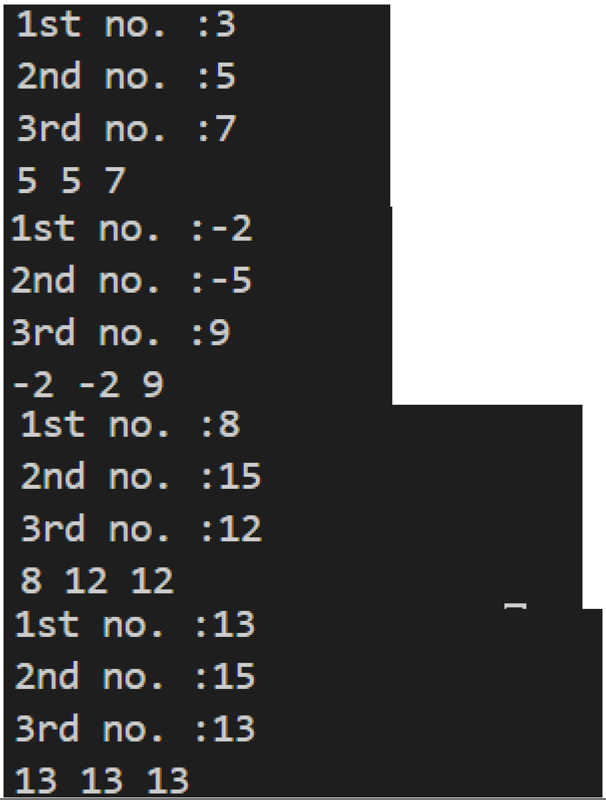
```
i = int(input("1st no. :"))
j = int(input("2nd no. :"))
k = int(input("3rd no. :"))
if i < j:
    if j < k:
        i = j
    else:
        j = k
else:
    if j < k:
        j = i
    else:
        i = k
print(i,j,k)
```

Sample Input and Output:

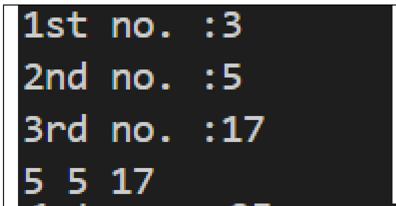
```
1st no. :3
2nd no. :5
3rd no. :7
5 5 7

1st no. :-2
2nd no. :-5
3rd no. :9
-2 -2 9
```

```
1st no. :8
2nd no. :15
3rd no. :12
8 12 12

1st no. :13
2nd no. :15
3rd no. :13
13 13 13

1st no. :3
2nd no. :5
3rd no. :17
5 5 17

1st no. :25
2nd no. :15
3rd no. :17
25 25 17
```

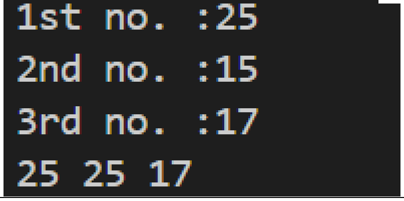
Screenshot of the Output:



```
1st no. :3
2nd no. :5
3rd no. :7
5 5 7
1st no. :-2
2nd no. :-5
3rd no. :9
-2 -2 9
1st no. :8
2nd no. :15
3rd no. :12
8 12 12
1st no. :13
2nd no. :15
3rd no. :13
13 13 13
```



```
1st no. :3
2nd no. :5
3rd no. :17
5 5 17
```



```
1st no. :25
2nd no. :15
3rd no. :17
25 25 17
```

Question No 12:

Write a Python program that takes a user's input for their exam score and provides the corresponding grade along with remarks. Consider the following grading scale:

A: 90-100 (Perfect score marked as 'A' with the remark 'Perfect score!')

B: 80-89 (Remark: 'Good job!')

C: 70-79 (Remark: 'Average performance.')

D: 60-69 (Remark: 'Below average.')

F: Below 60 (Remark: 'Failed.').

Include additional remarks for a perfect score (100) and excellent performance (grades A and B). Make use of if, elif, and else statements, as well as nested if and else statements in the same program.

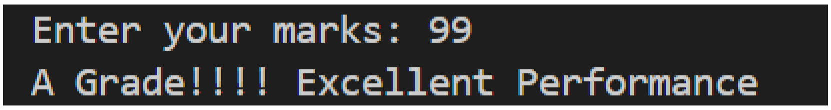
Source Code:

```
marks = int(input("Enter your marks: "))
if marks < 60:
   print ("F Grade!!!! Falied")
elif marks < 69:
   print ("D Grade!!!! Below average")
elif marks < 79:
   print ("C Grade!!!! Average Performance")
elif marks < 89:
   print ("B Grade!!!! Good Job! Excellent performance")
elif marks >89:
   if marks == 100:
       print ("A Grade!!!! Perfect score Excellent Performance")
   else:
       print ("A Grade!!!! Excellent Performance")
else:
   print ("Invalid Input")
```

Sample Input and Output:

```
Enter your marks: 99
A Grade!!!! Excellent Performance
```

Screenshot of the Output:



```
Enter your marks: 99
A Grade!!!! Excellent Performance
```

Question No 1:

Write a Python program to print whether a given number is an odd number or an even number using if-else conditional statements

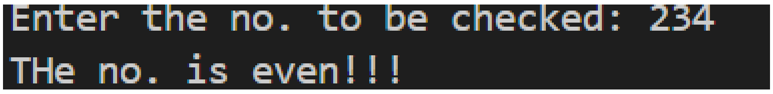
Source Code:

```
check = int(input("Enter the no. to be checked: "))
if check%2==0:
    print ("THe no. is even!!!")
else:
    print("The no. is odd!!")
```

Sample Input and Output:

```
Enter the no. to be checked: 234
THe no. is even!!!
```

Screenshot of the Output:



```
Enter the no. to be checked: 234
THe no. is even!!!
```

Question No 2:

Write a Python program to print the grade of a student by considering the marks of 5 subjects scored in his exam using chained conditionals

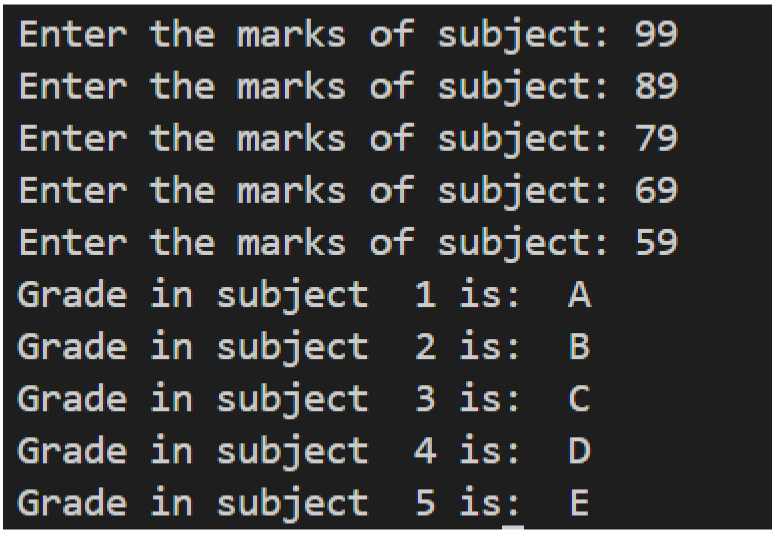
Source Code:

```
l = []
for i in range(5):
   x = int(input("Enter the marks of subject: "))
   if x >= 90:
      grade = "A"
   elif x >= 80:
      grade = "B"
   elif x >= 70:
      grade = "C"
   elif x >= 60:
      grade = "D"
   elif x >= 50:
      grade = "E"
   else:
      grade = "F"
   l.append(grade)
for j in range(5):
   print ("Grade in subject ", j+1 , "is: ",l[j])
```

Sample Input and Output:

```
Enter the marks of subject: 99
Enter the marks of subject: 89
Enter the marks of subject: 79
Enter the marks of subject: 69
Enter the marks of subject: 59
Grade in subject  1 is:  A
Grade in subject  2 is:  B
Grade in subject  3 is:  C
Grade in subject  4 is:  D
Grade in subject  5 is:  E
```

Screenshot of the Output:



```
Enter the marks of subject: 99
Enter the marks of subject: 89
Enter the marks of subject: 79
Enter the marks of subject: 69
Enter the marks of subject: 59
Grade in subject  1 is:  A
Grade in subject  2 is:  B
Grade in subject  3 is:  C
Grade in subject  4 is:  D
Grade in subject  5 is:  E
```

Question No 3:

Write a Python program to find the greatest of three numbers using nested if else conditionals.

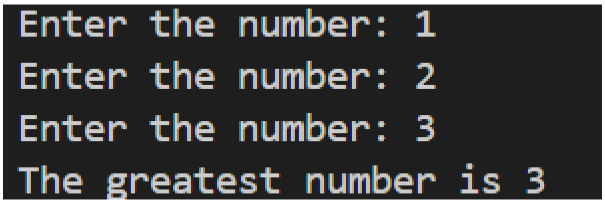
Source Code:

```
a  = int(input("Enter the number: "))
b  = int(input("Enter the number: "))
c  = int(input("Enter the number: "))
if c == a and b == a:
   print ("The numbers are equal")
else:
   if a>b:
      if a>c:
         print ("The greatest number is", a)
      elif c > a:
         print ("The greatest number is",c)
      else:
         print ("The greatest number is",a)
   elif b>a:
      if b>c:
         print ("The greatest number is",b)
      elif c > b:
         print ("The greatest number is",c)
      else:
         print ("The greatest number is",c)
   else:
      if a>c:
         print ("The greatest number is",a)
      elif c>a:
         print ("The greatest number is",c)
```

Sample Input and Output:

```
Enter the number: 1
Enter the number: 2
Enter the number: 3
The greatest number is 3
```

Screenshot of the Output:



```
Enter the number: 1
Enter the number: 2
Enter the number: 3
The greatest number is 3
```

Question No 4:

Write a Python script to assign a number to a variable num. Use the if elif else statement to show the corresponding day. 1 is for Monday, 2 is for Tuesday, and so on. if the number assigned is other than 1 to 7, show a message of ‘Invalid input’.

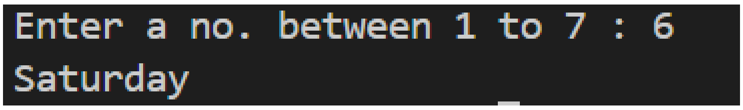
Source Code:

```
num = int(input("Enter a no. between 1 to 7 : "))
if num == 1:
    print ("Monday")
elif num == 2:
    print ("Tuesday")
elif num == 3:
    print ("Wednesday")
elif num == 4:
    print ("Thursday")
elif num == 5:
    print ("Friday")
elif num == 6:
    print ("Saturday")
elif num == 7:
    print ("Sunday")
else:
    print ("Invalid input")
```

Sample Input and Output:

```
Enter a no. between 1 to 7 : 6
Saturday
```

Screenshot of the Output:



```
Enter a no. between 1 to 7 : 6
Saturday
```

Question No 5:

Write a Python program to implement the roots of the quadratic equation

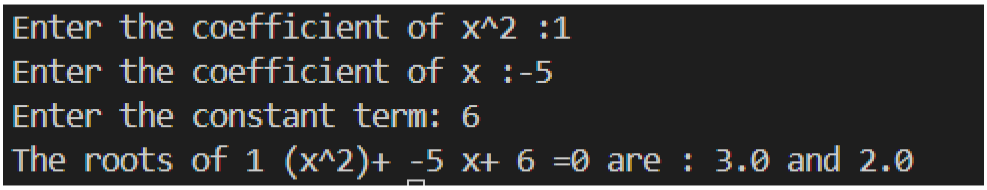
Source Code:

```
a = int(input("Enter the coefficient of x^2 :"))
b = int(input("Enter the coefficient of x :"))
c = int(input("Enter the constant term: "))
r1 = (-b+(((b**2) - 4*a*c)**(-1)))/2*a
r2 = (-b-(((b**2) - 4*a*c)**(-1)))/2*a
print ("The roots of",a,"(x^2)+",b,"x+",c,"=0 are :",r1,"and",r2)
```

Sample Input and Output:

```
Enter the coefficient of x^2 :1
Enter the coefficient of x :-5
Enter the constant term: 6
The roots of 1 (x^2)+ -5 x+ 6 =0 are : 3.0 and 2.0
```

Screenshot of the Output:



```
Enter the coefficient of x^2 :1
Enter the coefficient of x :-5
Enter the constant term: 6
The roots of 1 (x^2)+ -5 x+ 6 =0 are : 3.0 and 2.0
```

Python Programs on Looping Statements

Question No 1:

Write a Python program to find whether the given number is Armstrong or not using a while loop

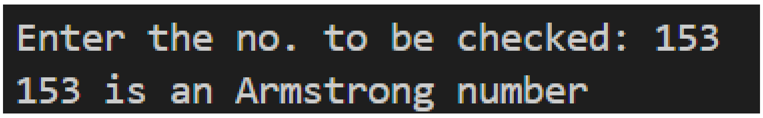
Source Code:

```
n = int(input("Enter the no. to be checked: "))
l = len(str(n))
check = n
e = 0
while check > 0:
   o = check % 10
   e = e+(o**l)
   check = check // 10
if e == n:
   print (n,"is an Armstrong number")
else:
   print (n,"is not an Armstrong number")
```

Sample Input and Output:

```
Enter the no. to be checked: 153
153 is an Armstrong number
```

Screenshot of the Output:



```
Enter the no. to be checked: 153
153 is an Armstrong number
```

Question No 2:

Write a Python program to find the factorial of a given number using a while loop

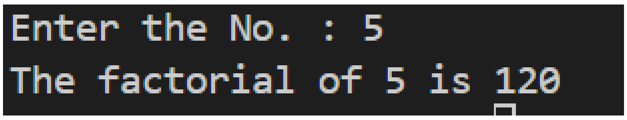
Source Code:

```
no = int(input("Enter the No. : "))
fact = 1
a = no
while a > 0:
   fact = fact*a
   a = a - 1
print ("The factorial of",no,"is",fact)
```

Sample Input and Output:

```
Enter the No. : 5
The factorial of 5 is 120
```

Screenshot of the Output:



```
Enter the No. : 5
The factorial of 5 is 120
```

Question No 3:

Write a Python program to check whether the given number is perfect number or not using for loop

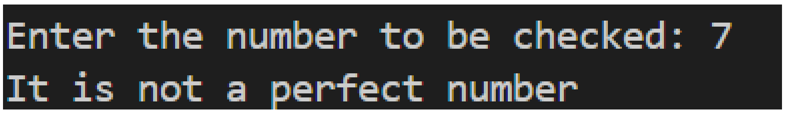
Source Code:

```
n = int(input("Enter the number to be checked: "))
pn = 0
for i in range(1,n):
   if n%i == 0:
      pn = pn + i
if pn == n:
   print("It is a perfect number")
else:
   print("It is not a perfect number")
```

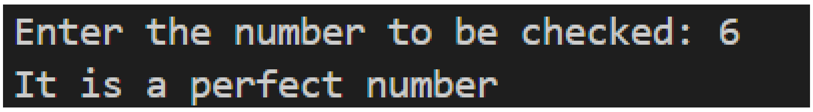
Sample Input and Output:

```
Enter the number to be checked: 7
It is not a perfect number


Enter the number to be checked: 6
It is a perfect number
```

Screenshot of the Output:



```
Enter the number to be checked: 7
It is not a perfect number
```



```
Enter the number to be checked: 6
It is a perfect number
```

Question No 4:

Write a Python program to print the Fibonacci series using for-loop

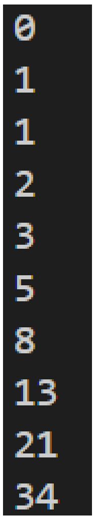
Source Code:

```
a = 0
b = 1
for i in range(10):
   print(a)
   o = a
   a = a + b
   b = o
```

Sample Input and Output:

```
0
1
1
2
3
5
8
13
21
34
```

Screenshot of the Output:



```
0
1
1
2
3
5
8
13
21
34
```

Question No 5:

Write sample Python programs to demonstrate the use of break and continue statements in while and for loop

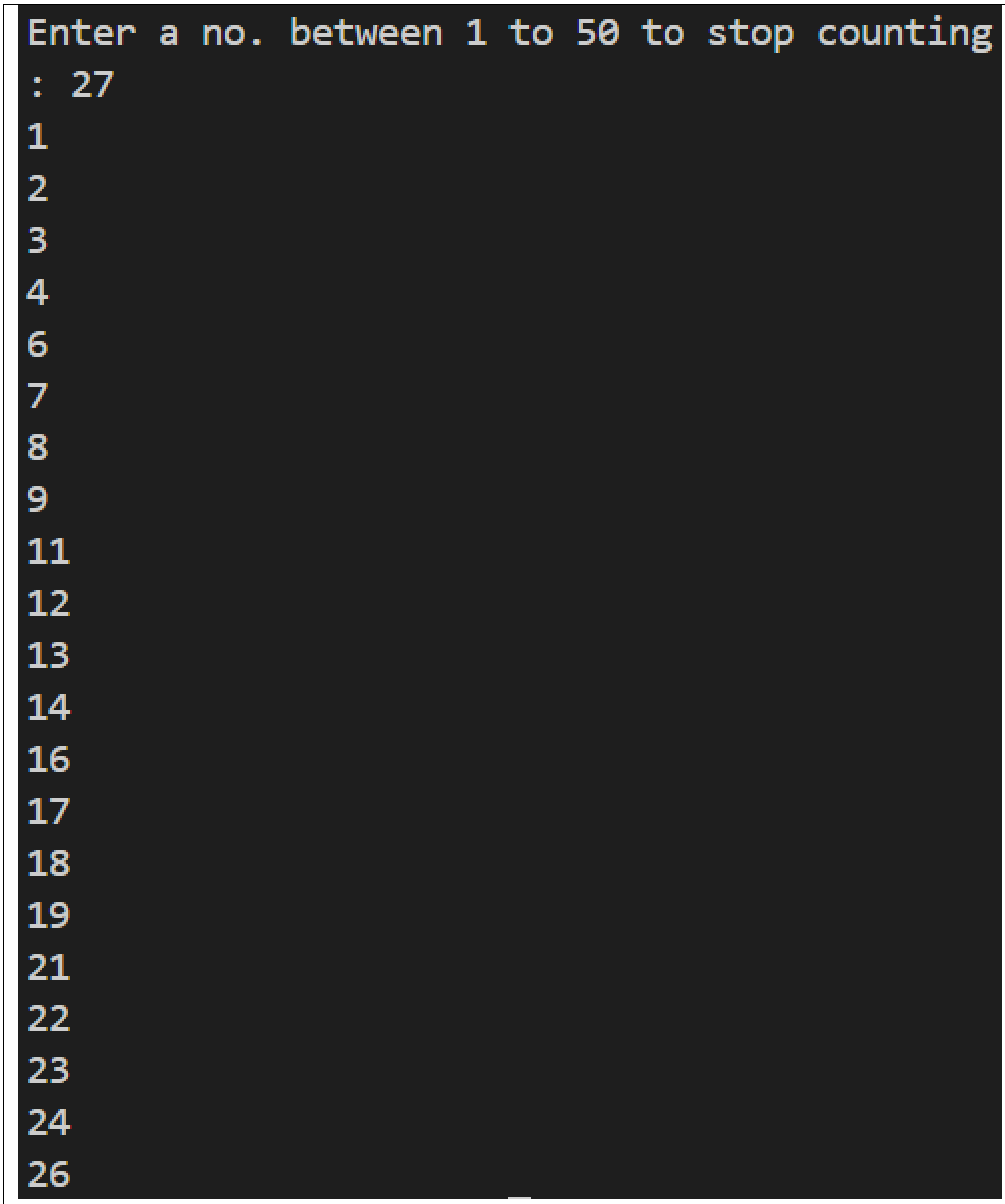
Source Code:

```
#program to print no. between 1 to 50 skipping multiples of 5 and stopping at a no. given by the user
n = int(input("Enter a no. between 1 to 50 to stop counting : "))
for i in range(50):
   if i == n:
      break
   elif i % 5 == 0:
      continue
   else:
        print (i)
```

Sample Input and Output:

```
Enter a no. between 1 to 50 to stop counting : 27
1
2
3
4
6
7
8
9
11
12
13
14
16
17
18
19
21
22
23
24
26
```

Screenshot of the Output:



```
Enter a no. between 1 to 50 to stop counting
: 27
1
2
3
4
6
7
8
9
11
12
13
14
16
17
18
19
21
22
23
24
26
```

Python Programs using Factoring Methods

Question no.1A:

Write a python program to Find the Square root of a given number Using ** Operator

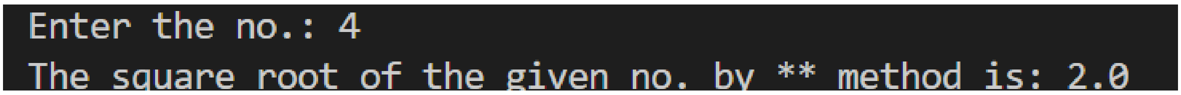
Source Code:

```
n=int(input("Enter the no.: "))
sqrt=n**0.5
print("The square root of the given no. by ** method is:",sqrt)
```

Sample Input and Output:

```
Enter the no.: 4
The square root of the given no. by ** method is: 2.0
```

Screenshot of the Execution:



```
Enter the no.: 4
The square root of the given no. by ** method is: 2.0
```

Question no.1B:

Write a python program to Find the Square root of a given complex number Using cmath

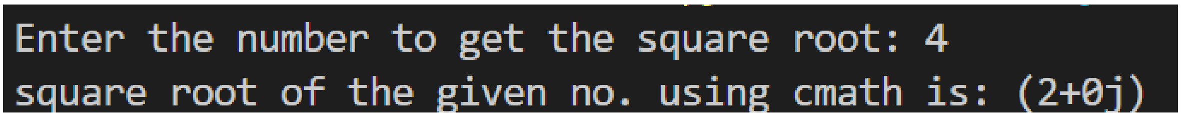
Source Code:

```
import cmath
n=int(input("Enter the number to get the square root: "))
print("square root of the given no. using cmath is:", cmath.sqrt(n))
```

Sample Input and Output:

```
Enter the number to get the square root: 4
square root of the given no. using cmath is: (2+0j)
```

Screenshot of the Execution:



```
Enter the number to get the square root: 4
square root of the given no. using cmath is: (2+0j)
```

Question no.2:

`Write a python program to find the Smallest divisor of an integer`

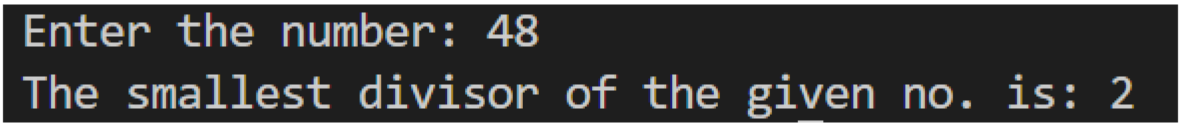
Source Code:

```
n=int(input("Enter the number: "))
divisors=[]
for i in range(2,n+1):
    if n%i == 0:
        divisors.append(i)

print("The smallest divisor of the given no. is:",min(divisors))
```

Sample Input and Output:

```
Enter the number: 48
The smallest divisor of the given no. is: 2
```

Screenshot of the Execution:



```
Enter the number: 48
The smallest divisor of the given no. is: 2
```

Question no.3:

`Write a python program to Find GCD using while loop`

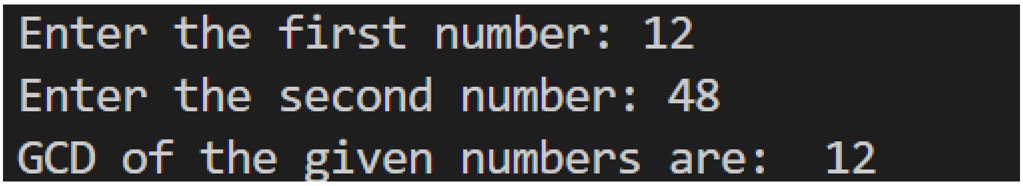
Source Code:

```
n1=int(input("Enter the first number: "))
n2=int(input("Enter the second number: "))
i=1
cd=[]
while i<= min(n1, n2):
    if min(n1, n2)%i == 0:
        if max(n1, n2)%i == 0:
            cd.append((i))
    i=i+1
print("GCD of the given numbers are: ", max(cd))
```

Sample Input and Output:

```
Enter the first number: 12
Enter the second number: 48
GCD of the given numbers are:  12
```

Screenshot of the Execution:



```
Enter the first number: 12
Enter the second number: 48
GCD of the given numbers are:  12
```

Question no.4A:

Write a python program to Generate first n prime numbers

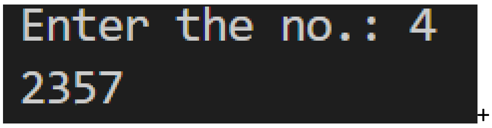
Source Code:

```
n=int(input("Enter the no.: "))
prime=[]
factors=[]
a=2
while n>len(prime):
   for i in range(1,a+1):
      if a%i==0:
         factors.append(i)
   if len(factors)==2:
      prime.append(a)
   factors=[]
   a=a+1
for i in prime:
   print(i,end="")
```

Sample Input and Output:

```
Enter the no.: 4
2357
```

Screenshot of the Execution:



```
Enter the no.: 4
2357
```

Question no.4B:

Write a python program to Find the prime factors of a number using for loop

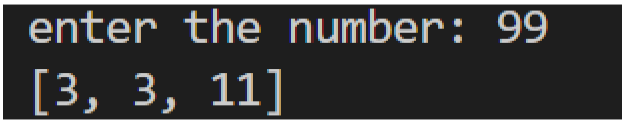
Source Code:

```
num=int(input("enter the number: "))
factors=[]
divisor=2
for i in range(num):
   for i in range(divisor, num + 1):
      if num % i == 0:
         factors.append(i)
         num //= i
         break
print(factors)
```

Sample Input and Output:

```
enter the number: 99
[3, 3, 11]
```

Screenshot of the Execution:



```
enter the number: 99
[3, 3, 11]
```

Question no.5:

Write a python program to Compute the nth Fibonacci numbers using recursion

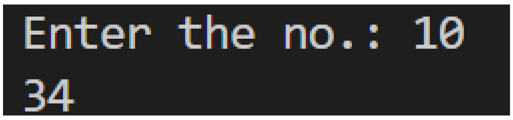
Source Code:

```
n=int(input("Enter the no.: "))
a = 0
b = 1
for i in range(n-1):
   o = a
   a = a + b
   b = o
print(a)
```

Sample Input and Output:

```
Enter the no.: 10
34
```

Screenshot of the Execution:



```
Enter the no.: 10
34
```

Python Programs using Arrays

Question no.1:

Write a Python program to add two matrices using nested loop

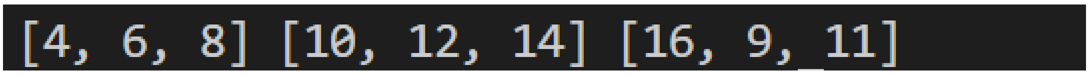
Source Code:

```
mat1=[[1,2,3],[4,5,6],[7,8,9]]
mat2=[[3,4,5],[6,7,8],[9,1,2]]
mat3=[[],[],[]]
list1=[]
for i in range(0,len(mat1)):
   for j in range(0,len(mat1[i])):
      list1.append(mat1[i][j]+mat2[i][j])
   mat3[i] = list1
   list1=[]
for i in range(0,len(mat3)):
   print(mat3[i],end=" ")
```

Sample input and output:

```
[4, 6, 8] [10, 12, 14] [16, 9, 11]
```

Screenshot of the execution



```
[4, 6, 8] [10, 12, 14] [16, 9, 11]
```

Question no.2:

Write a Python program to perform matrix transpose

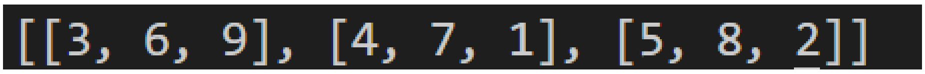
Source Code:

```
mat1=[]
mat2=[[3,4,5],[6,7,8],[9,1,2]]
for i in range(len(mat2)):
   list1=[]
   for j in range(len(mat2[i])):
      list1.append(mat2[j][i])
   mat1.append(list1)
print(mat1)
```

Sample input and output:

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

Screenshot of the execution



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

Question no.3:

```
Write a Python program to perform array pair sum
```

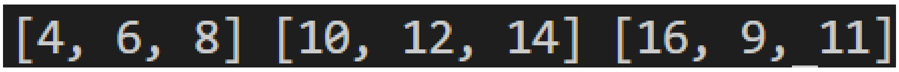
Source Code:

```
mat1=[[1,2,3],[4,5,6],[7,8,9]]
mat2=[[3,4,5],[6,7,8],[9,1,2]]
mat3=[[],[],[]]
list1=[]
for i in range(0,len(mat1)):
    for j in range(0,len(mat1[i])):
        list1.append(mat1[i][j]+mat2[i][j])
    mat3[i] = list1
    list1=[]
for i in range(0,len(mat3)):
    print(mat3[i],end=" ")
```

Sample input and output:

```
[4, 6, 8] [10, 12, 14] [16, 9, 11]
```

Screenshot of the execution



```
[4, 6, 8] [10, 12, 14] [16, 9, 11]
```

Question no.4:

```
Write a Python program to remove duplicates of the given array
```

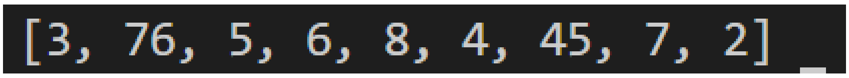
Source Code:

```
arr1=[3,76,5,6,8,4,3,45,6,7,8,5,4,3,2,2,3,4]
arr2=[]
for i in range(len(arr1)):
    if arr1[i] in arr2:
        continue
    else:
        arr2.append(arr1[i])
print(arr2)
```

Sample input and output:

```
[3, 76, 5, 6, 8, 4, 45, 7, 2]
```

Screenshot of the execution



```
[3, 76, 5, 6, 8, 4, 45, 7, 2]
```

Question no.5:

Write a python program to find the occurrence of given element in the array using function

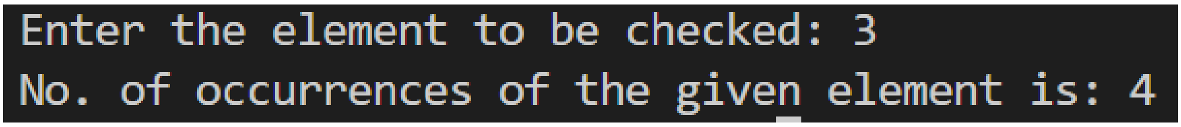
Source Code:

```
arr1=[3,76,5,6,8,4,3,45,6,7,8,5,4,3,2,2,3,4]
ele = int(input("Enter the element to be checked: "))
occurrences =0
for i in range(0, len(arr1)):
    if arr1[i]==ele:
        occurrences += 1
print("No. of occurrences of the given element is:",occurrences)
```

Sample input and output:

```
Enter the element to be checked: 3
No. of occurrences of the given element is: 4
```

Screenshot of the execution



```
Enter the element to be checked: 3
No. of occurrences of the given element is: 4
```

Python programs on List and perform various List operations

Question No 1a:

`Create a list of numbers 1 to 5`

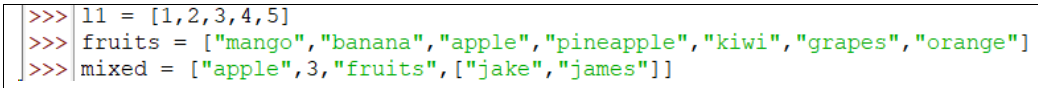
Source Code:

```
l1 = [1,2,3,4,5]
```

Sample Input and Output:

Creating a list doesn’t give any output

Screenshot of the Output:



```
>>> l1 = [1,2,3,4,5]
>>> fruits = ["mango","banana","apple","pineapple","kiwi","grapes","orange"]
>>> mixed = ["apple",3,"fruits",["jake","james"]]
```

Question No 1b:

`Create a list of strings with fruit names`

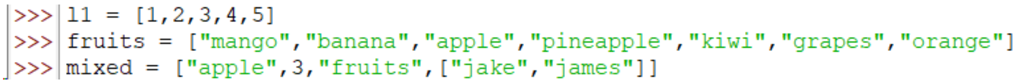
Source Code:

fruits = [“mango”,”banana”,”apple”,”pineapple”,”kiwi”,”grapes”,”orange”]

Sample Input and Output:

Creating a list doesn’t give any output

Screenshot of the Output:



```
>>> l1 = [1,2,3,4,5]
>>> fruits = ["mango","banana","apple","pineapple","kiwi","grapes","orange"]
>>> mixed = ["apple",3,"fruits",["jake","james"]]
```

Question No 1c:

`Create a mixed type of list`

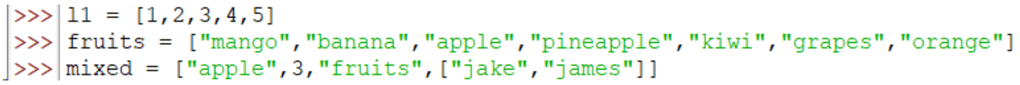
Source Code:

mixed = [“apple”,3,”fruits”,[“jake”,”james”]]

Sample Input and Output:

Creating a list doesn’t give any output

Screenshot of the Output:



```
>>> l1 = [1,2,3,4,5]
>>> fruits = ["mango","banana","apple","pineapple","kiwi","grapes","orange"]
>>> mixed = ["apple",3,"fruits",["jake","james"]]
```

Question No 1d:

Consider any of the list above and Modify the elements of list by index

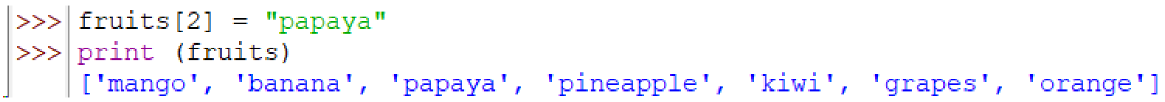
Source Code:

fruits[2] = “papaya”
print (fruits)

Sample Input and Output:

[‘mango’,’banana’,’papaya’,’pineapple’,’kiwi’,’grapes’,’orange’]

Screenshot of the Output:



```
>>> fruits[2] = "papaya"
>>> print (fruits)
    ['mango', 'banana', 'papaya', 'pineapple', 'kiwi', 'grapes', 'orange']
```

Question No 1e:

Consider any of the list above and Append an element to the end

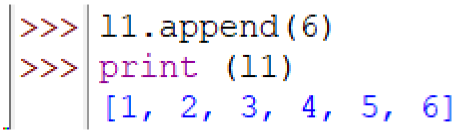
Source Code:

l1.append(6)
print (l1)

Sample Input and Output:

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

Screenshot of the Output:



```
>>> l1.append(6)
>>> print (l1)
    [1, 2, 3, 4, 5, 6]
```

Question No 1f:

Consider any of the list above and Insert an element at a specific index

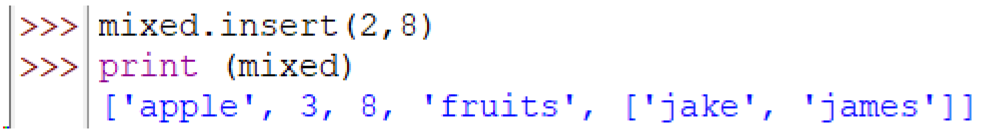
Source Code:

mixed.insert(2,8)
print (mixed)

Sample Input and Output:

[‘apple’,3,8,’fruits’,[‘jake’,’james’]]

Screenshot of the Output:



```
>>> mixed.insert(2,8)
>>> print (mixed)
    ['apple', 3, 8, 'fruits', ['jake', 'james']]
```

Question No 1g:

Consider any of the list above and Remove an element by value

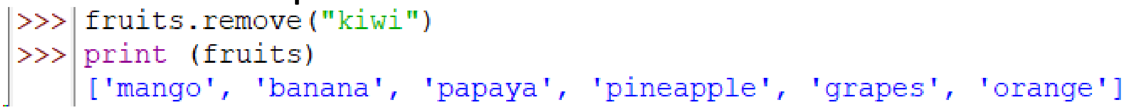
Source Code:

fruits.remove(“kiwi”)
print (fruits)

Sample Input and Output:

[‘mango’,’banana’,’papaya’,’pineapple’,’grapes’,’orange’]

Screenshot of the Output:



```
>>> fruits.remove("kiwi")
>>> print (fruits)
    ['mango', 'banana', 'papaya', 'pineapple', 'grapes', 'orange']
```

Question No 1h:

Consider any of the list above and Remove an element by index

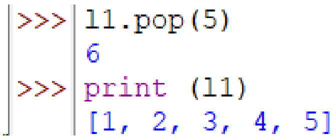
Source Code:

```
l1.pop(5)
```

Sample Input and Output:

```
6
```

Screenshot of the Output:



```
>>> l1.pop(5)
6
>>> print (l1)
[1, 2, 3, 4, 5]
```

Question No 1i:

Find the length of the list that has been created above

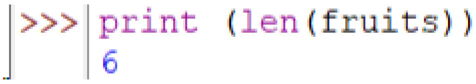
Source Code:

```
print (len(fruits))
```

Sample Input and Output:

```
6
```

Screenshot of the Output:



```
>>> print (len(fruits))
6
```

Question No 1j:

Consider any of the list above and Check if an element is in a list

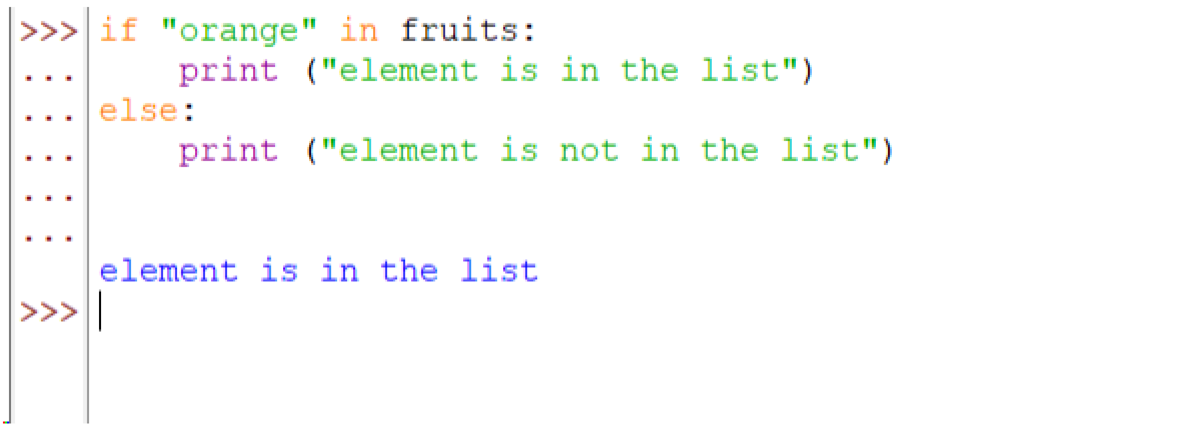
Source Code:

```
if “orange” in fruits:
   print (“element is in the list”)
else:
   print (“element is not in the list”)
```

Sample Input and Output:

```
element is in the list
```

Screenshot of the Output:



```
>>> if "orange" in fruits:
...     print ("element is in the list")
... else:
...     print ("element is not in the list")
...
...

element is in the list
>>> 
```

Question No 2:

Write a program to create two lists and concatenate them into a single list.

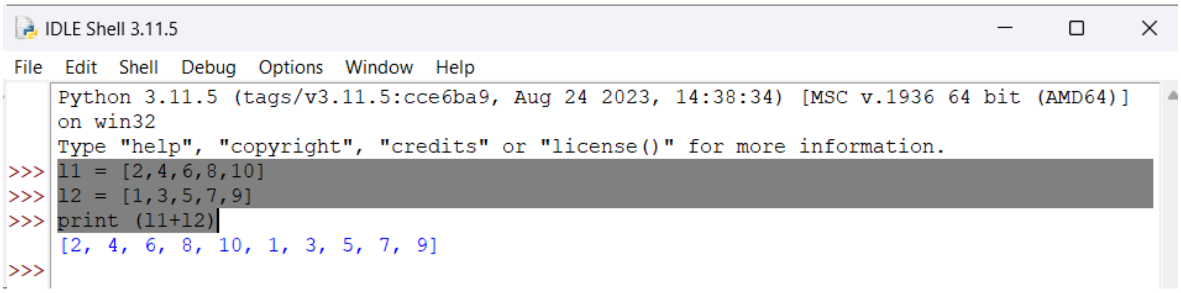
Source Code:

l1 = [2,4,6,8,10]
l2 = [1,3,5,7,9]
print (l1+l2)

Sample Input and Output:

[2, 4, 6, 8, 10, 1, 3, 5, 7, 9]

Screenshot of the Output:



```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> l1 = [2,4,6,8,10]
>>> l2 = [1,3,5,7,9]
>>> print (l1+l2)
    [2, 4, 6, 8, 10, 1, 3, 5, 7, 9]
>>>
```

Question No 3:

Write a Python program to find the maximum and minimum values in a list.

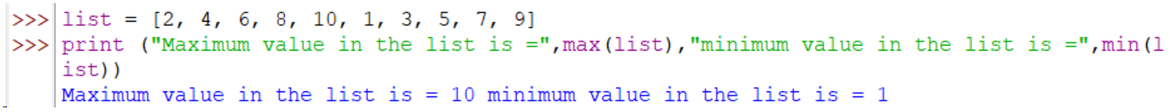
Source Code:

```
list = [2, 4, 6, 8, 10, 1, 3, 5, 7, 9]
print ("Maximum value in the list is =",max(list),"minimum value in the list is
=",min(list))
```

Sample Input and Output:

Maximum value in the list is = 10 minimum value in the list is = 1

Screenshot of the Output:



```
>>> list = [2, 4, 6, 8, 10, 1, 3, 5, 7, 9]
>>> print ("Maximum value in the list is =",max(list),"minimum value in the list is =",min(l
ist))
Maximum value in the list is = 10 minimum value in the list is = 1
```

Question No 4:

Write a Python program to remove all duplicate elements from a list and print the updated list.

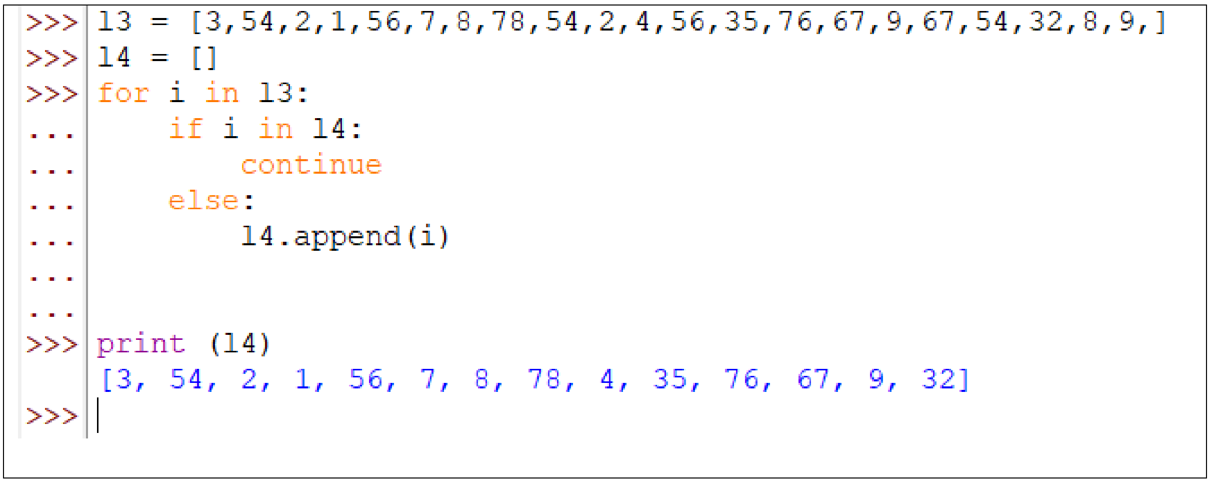
Source Code:

```
l3 = [3,54,2,1,56,7,8,78,54,2,4,56,35,76,67,9,67,54,32,8,9,]
l4 = []
for i in l3:
   if i in l4:
      continue
   else:
      l4.append(i)


print (l4)
```

Sample Input and Output:

[3, 54, 2, 1, 56, 7, 8, 78, 4, 35, 76, 67, 9, 32]

Screenshot of the Output:



```
>>> l3 = [3,54,2,1,56,7,8,78,54,2,4,56,35,76,67,9,67,54,32,8,9,]
>>> l4 = []
>>> for i in l3:
...     if i in l4:
...         continue
...     else:
...         l4.append(i)
...
...
>>> print (l4)
[3, 54, 2, 1, 56, 7, 8, 78, 4, 35, 76, 67, 9, 32]
>>> 
```

Question No 5:

`Write a program to check if a given element is present in a list.`

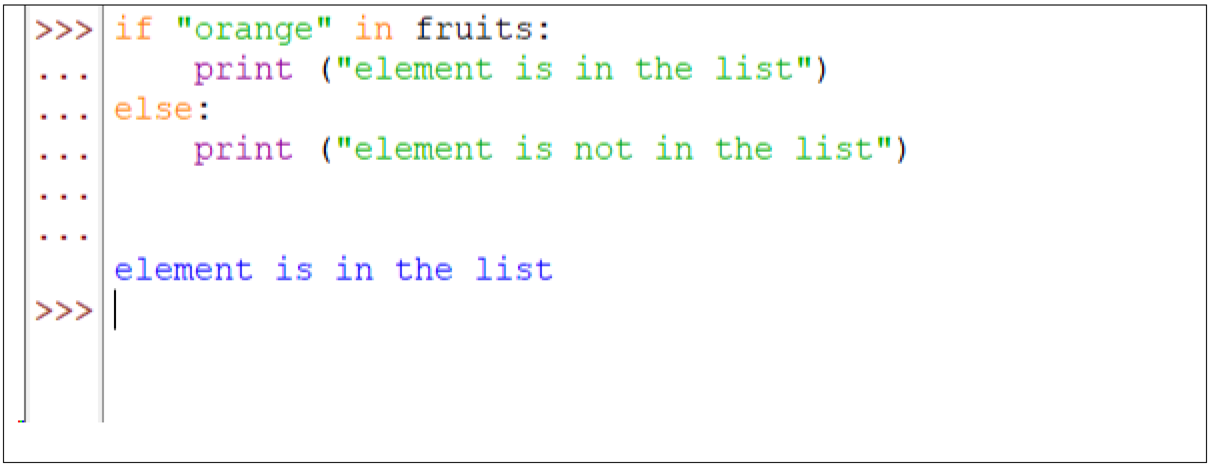
Source Code:

```
fruits = [“mango”,”banana”,”apple”,”pineapple”,”kiwi”,”grapes”,”orange”]
if “orange” in fruits:
   print (“element is in the list”)
else:
   print (“element is not in the list”)
```

Sample Input and Output:

element is in the list

Screenshot of the Output:

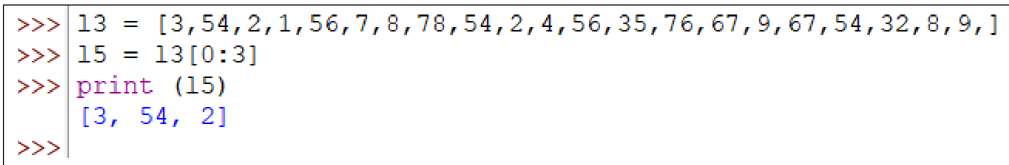


```
>>> if "orange" in fruits:
...     print ("element is in the list")
... else:
...     print ("element is not in the list")
...
...
element is in the list
>>> 
```

Question No 6: Write a program to extract the first three elements from a list using list slicing.

Source Code:
l3 = [3,54,2,1,56,7,8,78,54,2,4,56,35,76,67,9,67,54,32,8,9,]
l5 = l3[0:3]
print (l5)

Sample Input and Output: [3, 54, 2]
Screenshot of the Output:



```
>>> l3 = [3,54,2,1,56,7,8,78,54,2,4,56,35,76,67,9,67,54,32,8,9,]
>>> l5 = l3[0:3]
>>> print (l5)
 [3, 54, 2]
>>>
```

Question No 7:

`Write a program to Create a list and use list slicing to extract the elements from index 2 to index 5.`

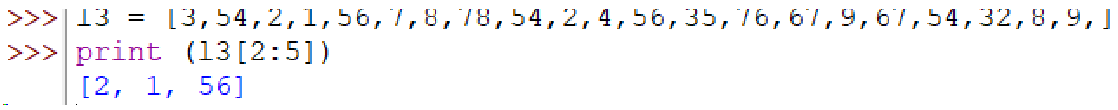
Source Code:

l3 = [3,54,2,1,56,7,8,78,54,2,4,56,35,76,67,9,67,54,32,8,9,]
print (l3[2:5])

Sample Input and Output:

[2, 1, 56]

Screenshot of the Output:



```
>>> l3 = [3,54,2,1,56,7,8,78,54,2,4,56,35,76,67,9,67,54,32,8,9,]
>>> print (l3[2:5])
 [2, 1, 56]
```

Question No 8:

`Write a program to reverse the elements of the list using list slicing.`

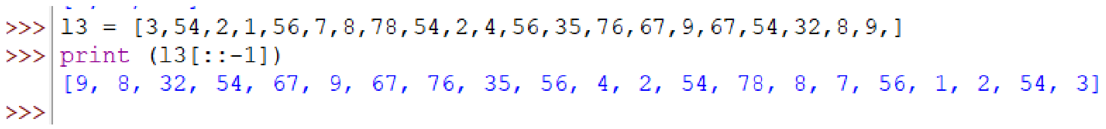
Source Code:

l3 = [3,54,2,1,56,7,8,78,54,2,4,56,35,76,67,9,67,54,32,8,9,]
print (l3[::-1])

Sample Input and Output:

[9, 8, 32, 54, 67, 9, 67, 76, 35, 56, 4, 2, 54, 78, 8, 7, 56, 1, 2, 54, 3]

Screenshot of the Output:



```
>>> l3 = [3,54,2,1,56,7,8,78,54,2,4,56,35,76,67,9,67,54,32,8,9,]
>>> print (l3[::-1])
 [9, 8, 32, 54, 67, 9, 67, 76, 35, 56, 4, 2, 54, 78, 8, 7, 56, 1, 2, 54, 3]
>>>
```

Question No 9:

Write a program to Create a list and use list slicing to extract every alternate element.

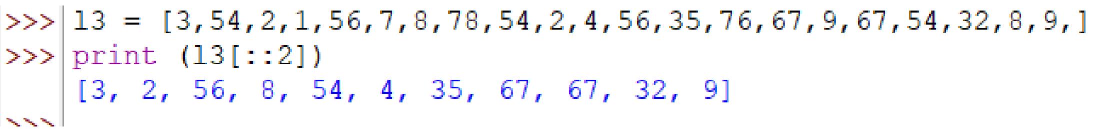
Source Code:

```
l3 = [3,54,2,1,56,7,8,78,54,2,4,56,35,76,67,9,67,54,32,8,9,]
print (l3[::2])
```

Sample Input and Output:

```
[3, 2, 56, 8, 54, 4, 35, 67, 67, 32, 9]
```

Screenshot of the Output:



```
>>> l3 = [3,54,2,1,56,7,8,78,54,2,4,56,35,76,67,9,67,54,32,8,9,]
>>> print (l3[::2])
    [3, 2, 56, 8, 54, 4, 35, 67, 67, 32, 9]
>>>
```

Question No 10:

Write a program to extract the last three elements using negative list slicing.

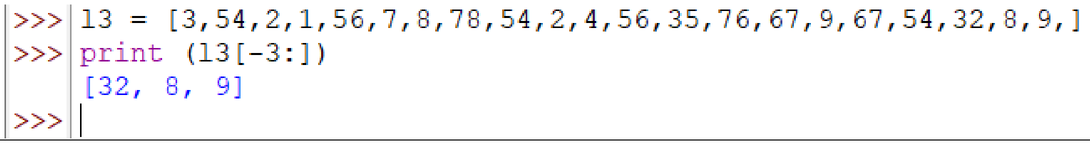
Source Code:

```
l3 = [3,54,2,1,56,7,8,78,54,2,4,56,35,76,67,9,67,54,32,8,9,]
print (l3[-3:])
```

Sample Input and Output:

```
[32, 8, 9]
```

Screenshot of the Output:



```
>>> l3 = [3,54,2,1,56,7,8,78,54,2,4,56,35,76,67,9,67,54,32,8,9,]
>>> print (l3[-3:])
    [32, 8, 9]
>>>
```

Python programs on Tuple and perform various Tuple operations

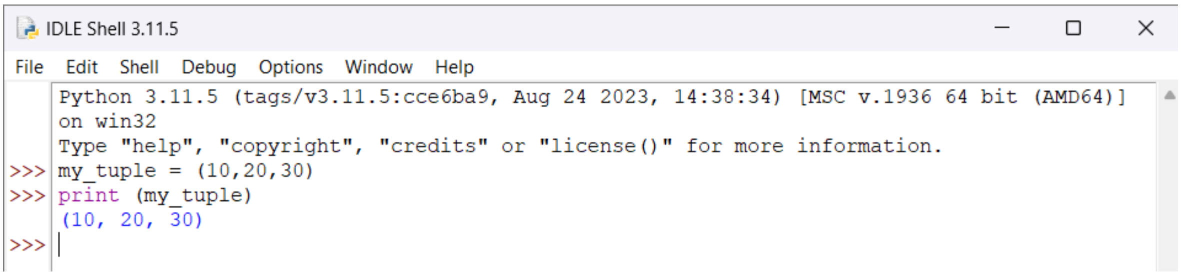
Question No 1a: `Create a tuple named my_tuple with elements 10, 20, and 30.`
Source Code:

my_tuple = (10,20,30)

Sample Input and Output:

creating a tuple does not give any output

Screenshot of the Output:



```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> my_tuple = (10,20,30)
>>> print (my_tuple)
(10, 20, 30)
>>> 
```

Question No 1b:

Access the second element of the tuple my_tuple

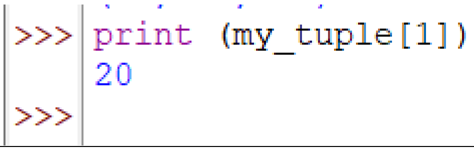
Source Code:

print (my_tuple[1])

Sample Input and Output:

20

Screenshot of the Output:



```
>>> print (my_tuple[1])
20
>>>
```

Question No 1c:

Concatenate the tuple my_tuple with a new tuple (40, 50) and

store the result in a new variable named combined_tuple.

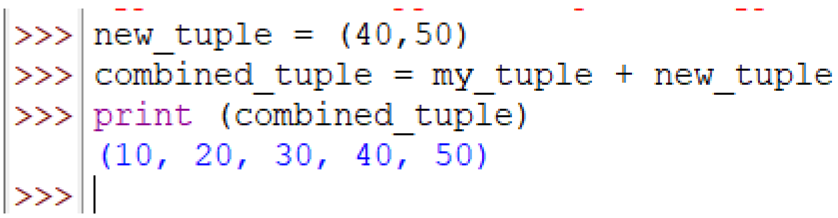
Source Code:

new_tuple = (40,50)
combined_tuple = my_tuple + new_tuple
print (combined_tuple)

Sample Input and Output:

(10, 20, 30, 40, 50)

Screenshot of the Output:



```
>>> new_tuple = (40,50)
>>> combined_tuple = my_tuple + new_tuple
>>> print (combined_tuple)
    (10, 20, 30, 40, 50)
>>> 
```

Question No 1d:

`Unpack the elements of the tuple combined_tuple into separate variables a, b, c, d, and e.`

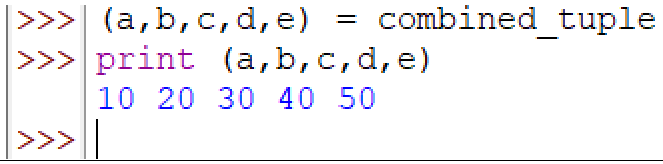
Source Code:

(a,b,c,d,e) = combined_tuple
print (a,b,c,d,e)

Sample Input and Output:

10 20 30 40 50

Screenshot of the Output:



```
>>> (a,b,c,d,e) = combined_tuple
>>> print (a,b,c,d,e)
    10 20 30 40 50
>>> 
```

Question No 1e:

`Find the index of element 30 in the tuple combined_tuple.`

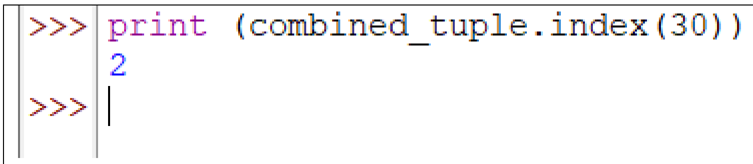
Source Code:

print (combined_tuple.index(30))

Sample Input and Output:

2

Screenshot of the Output:



```
>>> print (combined_tuple.index(30))
2
>>> |
```

Question No 1f:

Count the number of occurrences of element 20 in the tuple combined_tuple.

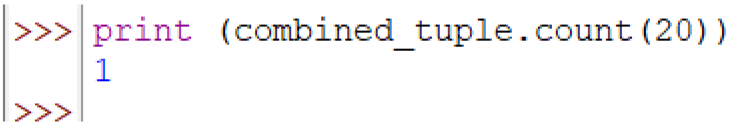
Source Code:

```
print (combined_tuple.count(20))
```

Sample Input and Output:

```
1
```

Screenshot of the Output:



```
>>> print (combined_tuple.count(20))
1
>>>
```

Question No 1g:

Convert the tuple combined_tuple to a list named my_list.

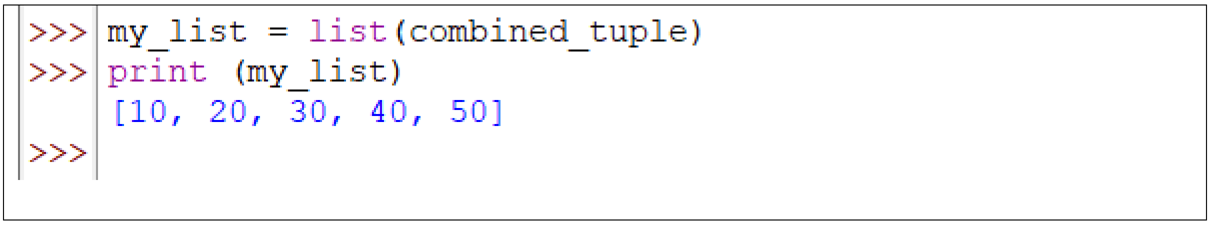
Source Code:

```
my_list = list(combined_tuple)
print (my_list)
```

Sample Input and Output:

```
[10, 20, 30, 40, 50]
```

Screenshot of the Output:



```
>>> my_list = list(combined_tuple)
>>> print (my_list)
[10, 20, 30, 40, 50]
>>>
```

Question No 1h:

`Check if element 40 is present in the tuple combined_tuple.`

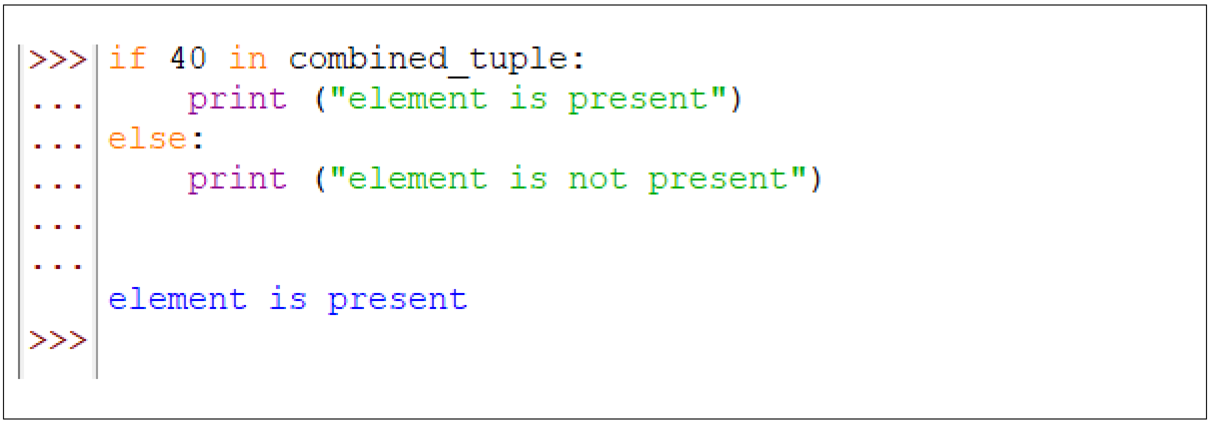
Source Code:

```
if 40 in combined_tuple:
   print ("element is present")
else:
   print ("element is not present")
```

Sample Input and Output:

```
element is present
```

Screenshot of the Output:



```
>>> if 40 in combined_tuple:
...     print ("element is present")
... else:
...     print ("element is not present")
...
...
element is present
>>>
```

Question No 1i:

`Create a new tuple named sliced_tuple containing elements`

`from index 1 to 4 (inclusive) from the combined_tuple.`

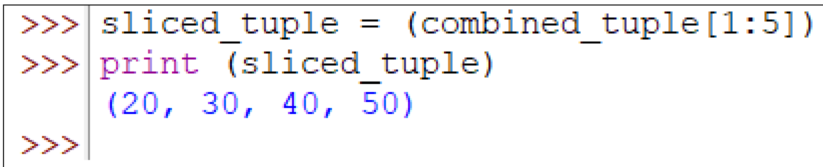
Source Code:

```
sliced_tuple = (combined_tuple[1:5])
print (sliced_tuple)
```

Sample Input and Output:

(20, 30, 40, 50)

Screenshot of the Output:



```
>>> sliced_tuple = (combined_tuple[1:5])
>>> print (sliced_tuple)
    (20, 30, 40, 50)
>>>
```

Question No 1j:

`Multiply each element of the tuple my_tuple by 3 and`

`store the result in a new tuple named multiplied_tuple.`

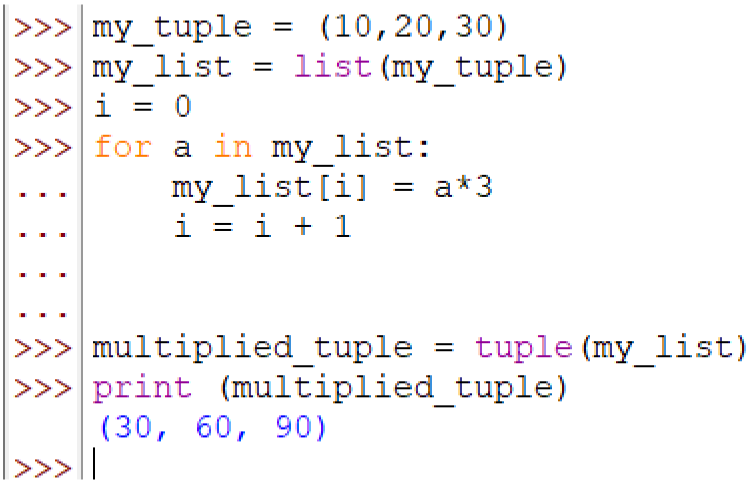
Source Code:

```
my_tuple = (10,20,30)
my_list = list(my_tuple)
i = 0
for a in my_list:
   my_list[i] = a*3
   i = i + 1


multiplied_tuple = tuple(my_list)
print (multiplied_tuple)
```

Sample Input and Output:

(30, 60, 90)

Screenshot of the Output:



```
>>> my_tuple = (10,20,30)
>>> my_list = list(my_tuple)
>>> i = 0
>>> for a in my_list:
...     my_list[i] = a*3
...     i = i + 1
...
...
>>> multiplied_tuple = tuple(my_list)
>>> print (multiplied_tuple)
    (30, 60, 90)
>>> 
```

Question No 2:

`Write a program that concatenates two tuples and prints the result.`

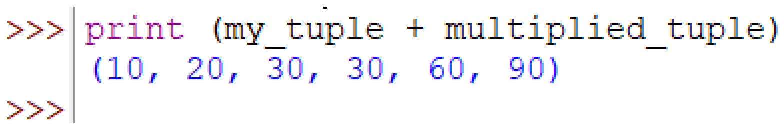
Source Code:

```
print (my_tuple + multiplied_tuple)
```

Sample Input and Output:

```
(10, 20, 30, 30, 60, 90)
```

Screenshot of the Output:



```
>>> print (my_tuple + multiplied_tuple)
    (10, 20, 30, 30, 60, 90)
>>>
```

Question No 3:

`Write a program that demonstrates tuple packing and unpacking.`

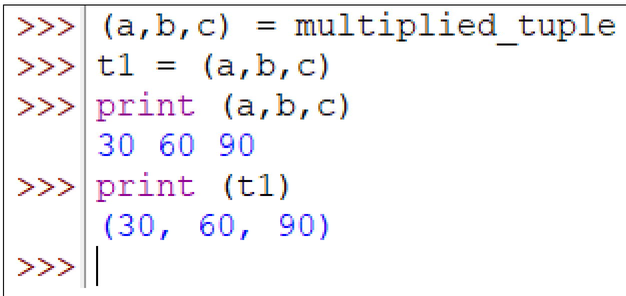
Source Code:

```
(a,b,c) = multiplied_tuple
t1 = (a,b,c)
print (a,b,c)
print (t1)
```

Sample Input and Output:

```
30 60 90
(30, 60, 90)
```

Screenshot of the Output:



```
>>> (a,b,c) = multiplied_tuple
>>> t1 = (a,b,c)
>>> print (a,b,c)
30 60 90
>>> print (t1)
(30, 60, 90)
>>> 
```

Question No 4:

```
Write a program that replicates elements in a tuple.
```

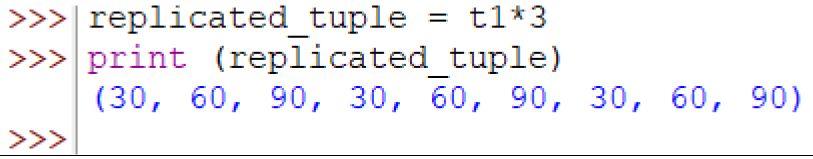
Source Code:

replicated_tuple = t1*3
print (replicated_tuple)

Sample Input and Output:

(30, 60, 90, 30, 60, 90, 30, 60, 90)

Screenshot of the Output:



```
>>> replicated_tuple = t1*3
>>> print (replicated_tuple)
(30, 60, 90, 30, 60, 90, 30, 60, 90)
>>> 
```

Question No 5:

```
Write a program that slices a tuple to get a subset of elements.
```

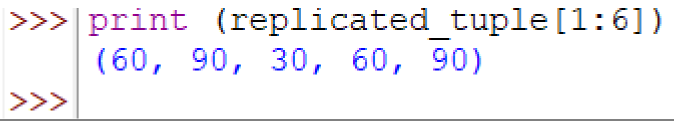
Source Code:

print (replicated_tuple[1:6])

Sample Input and Output:

(60, 90, 30, 60, 90)

Screenshot of the Output:



```
>>> print (replicated_tuple[1:6])
(60, 90, 30, 60, 90)
>>> 
```

Question No 6:

`Write a program that compares two tuples.`

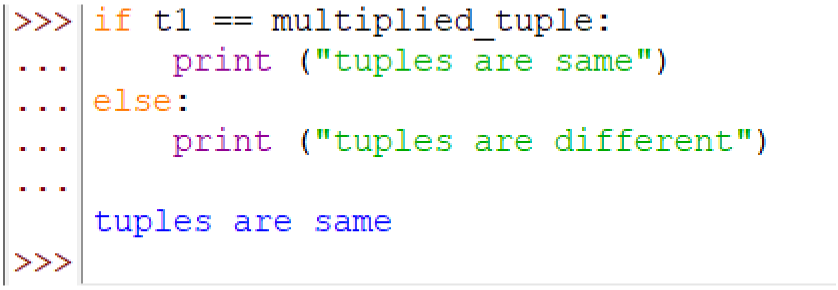
Source Code:

```
if t1 == multiplied_tuple:
   print ("tuples are same")
else:
   print ("tuples are different")
```

Sample Input and Output:

```
tuples are same
```

Screenshot of the Output:



```
>>> if t1 == multiplied_tuple:
...     print ("tuples are same")
... else:
...     print ("tuples are different")
...
tuples are same
>>>
```

Question No 7:

`Write a program that checks if an element is present in a tuple.`

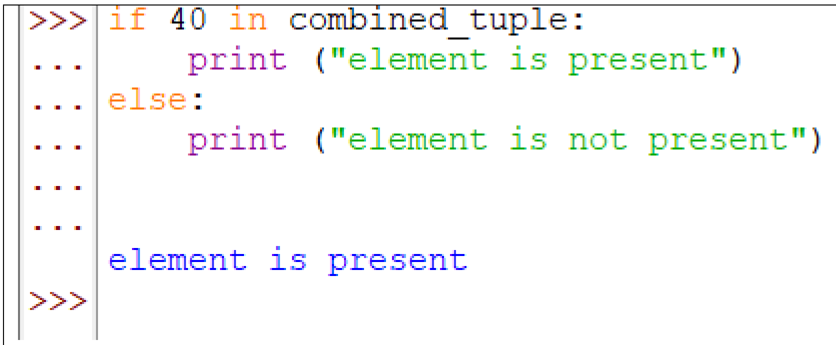
Source Code:

```
if 40 in combined_tuple:
   print ("element is present")
else:
   print ("element is not present")
```

Sample Input and Output:

```
element is present
```

Screenshot of the Output:



```
>>> if 40 in combined_tuple:
...     print ("element is present")
... else:
...     print ("element is not present")
...
...
element is present
>>>
```

Question No 8:

Write a program that calculates the length of a tuple.

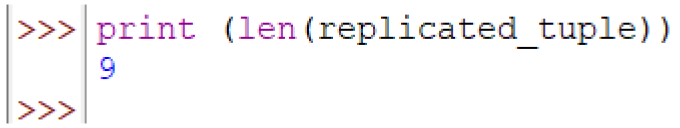
Source Code:

print (len(replicated_tuple))

Sample Input and Output:

9

Screenshot of the Output:



```
>>> print (len(replicated_tuple))
9
>>>
```

Question No 9:

Write a program that unpacks elements from a tuple.

Source Code:

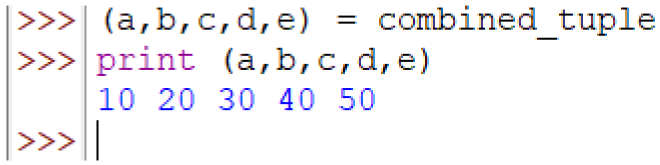
(a,b,c,d,e) = combined_tuple

print (a,b,c,d,e)

Sample Input and Output:

10 20 30 40 50

Screenshot of the Output:



```
>>> (a,b,c,d,e) = combined_tuple
>>> print (a,b,c,d,e)
10 20 30 40 50
>>>
```

Question No 10:

Write a program that uses nested tuples.

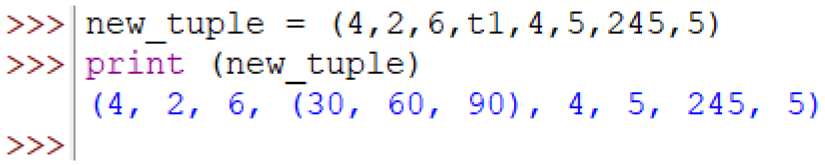
Source Code:

```
new_tuple = (4,2,6,t1,4,5,245,5)
print (new_tuple)
```

Sample Input and Output:

```
(4, 2, 6, (30, 60, 90), 4, 5, 245, 5)
```

Screenshot of the Output:



```
>>> new_tuple = (4,2,6,t1,4,5,245,5)
>>> print (new_tuple)
 (4, 2, 6, (30, 60, 90), 4, 5, 245, 5)
>>>
```

Python program for creating Set and perform various Set operations

Question no. 1:

```
Write a Python Program to create and perform various set operations .
```

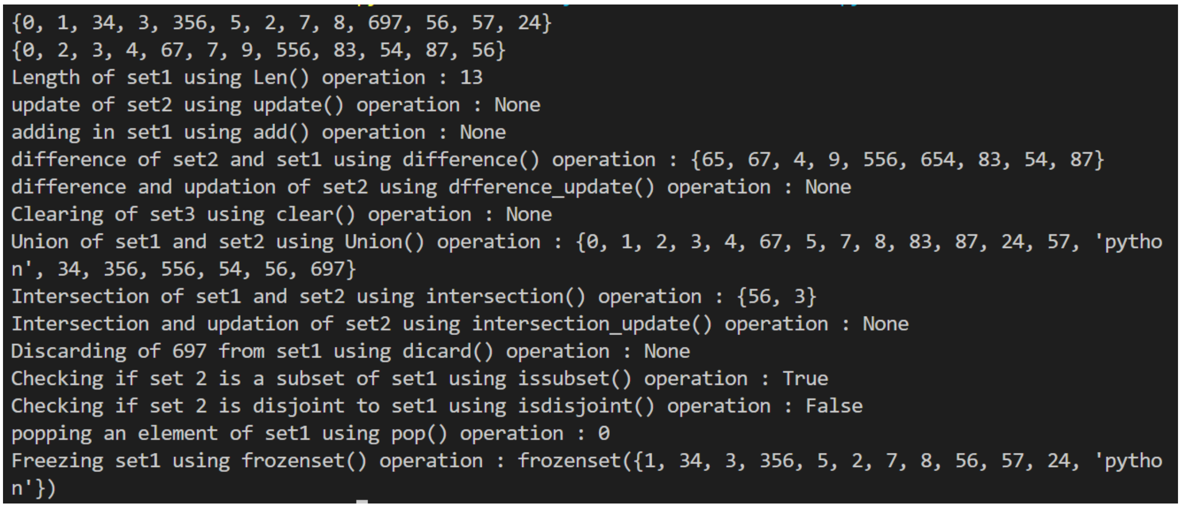
Source Code:

```
list1=[1,5,34,56,3,7,57,356,7,24,8,697,0,2]
list2=[2,3,54,2,56,87,54,2,7,9,0,556,83,4,67,4]
set1=set(list1)
set2=set(list2)
print(set1)
print(set2)
print("Length of set1 using Len() operation :",len(set1))
set3={7,8,65,654,34,0,9,2}
print("update of set2 using update() operation :",set2.update(set3))
print("adding in set1 using add() operation :",set1.add("python"))
print("difference of set2 and set1 using difference() operation :",set2.difference(set1))
print("difference and updation of set2 using dfference_update() operation
:",set2.difference_update(set3))
print("Clearing of set3 using clear() operation :",set3.clear())
print("Union of set1 and set2 using Union() operation :",set2.union(set1))
print("Intersection of set1 and set2 using intersection() operation :",set2.intersection(set1))
print("Intersection and updation of set2 using intersection_update() operation
:",set2.intersection_update(set1))
print("Discarding of 697 from set1 using dicard() operation :",set1.discard(697))
print("Checking if set 2 is a subset of set1 using issubset() operation :",set2.issubset(set1))
print("Checking if set 2 is disjoint to set1 using isdisjoint() operation :",set2.isdisjoint(set1))
print("popping an element of set1 using pop() operation :",set1.pop())
print("Freezing set1 using frozenset() operation :",frozenset(set1))
```

Sample Input and Output:

```
{0, 1, 34, 3, 356, 5, 2, 7, 8, 697, 56, 57, 24}
{0, 2, 3, 4, 67, 7, 9, 556, 83, 54, 87, 56}
Length of set1 using Len() operation : 13
update of set2 using update() operation : None
adding in set1 using add() operation : None
difference of set2 and set1 using difference() operation : {65, 67, 4, 9, 556, 654, 83, 54, 87}
difference and updation of set2 using dfference_update() operation : None
Clearing of set3 using clear() operation : None
Union of set1 and set2 using Union() operation : {0, 1, 2, 3, 4, 67, 5, 7, 8, 83, 87, 24, 57, 'python', 34, 356,
556, 54, 56, 697}
Intersection of set1 and set2 using intersection() operation : {56, 3}
Intersection and updation of set2 using intersection_update() operation : None
Discarding of 697 from set1 using dicard() operation : None
Checking if set 2 is a subset of set1 using issubset() operation : True
Checking if set 2 is disjoint to set1 using isdisjoint() operation : False
popping an element of set1 using pop() operation : 0
Freezing set1 using frozenset() operation : frozenset({1, 34, 3, 356, 5, 2, 7, 8, 56, 57, 24, 'python'})
```

Screenshot of the Execution:



```
{0, 1, 34, 3, 356, 5, 2, 7, 8, 697, 56, 57, 24}
{0, 2, 3, 4, 67, 7, 9, 556, 83, 54, 87, 56}
Length of set1 using Len() operation : 13
update of set2 using update() operation : None
adding in set1 using add() operation : None
difference of set2 and set1 using difference() operation : {65, 67, 4, 9, 556, 654, 83, 54, 87}
difference and updation of set2 using dfference_update() operation : None
Clearing of set3 using clear() operation : None
Union of set1 and set2 using Union() operation : {0, 1, 2, 3, 4, 67, 5, 7, 8, 83, 87, 24, 57, 'pytho
n', 34, 356, 556, 54, 56, 697}
Intersection of set1 and set2 using intersection() operation : {56, 3}
Intersection and updation of set2 using intersection_update() operation : None
Discarding of 697 from set1 using dicard() operation : None
Checking if set 2 is a subset of set1 using issubset() operation : True
Checking if set 2 is disjoint to set1 using isdisjoint() operation : False
popping an element of set1 using pop() operation : 0
Freezing set1 using frozenset() operation : frozenset({1, 34, 3, 356, 5, 2, 7, 8, 56, 57, 24, 'pytho
n'})
```

Python program for creating Dictionary and perform various Dictionary operations

Question no. 1:

`Write a Python Program to create and perform various dictionary`

```
operations .
```

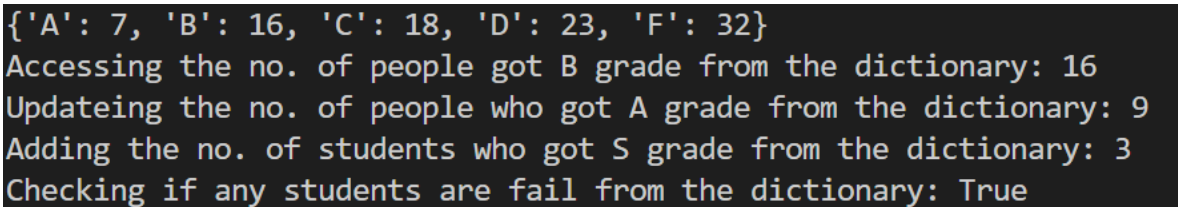
Source Code:

```
dict1={"A":7,"B":16,"C":18,"D":23,"F":32} #grade and no. of people got it
print(dict1)
print("Accessing the no. of people got B grade from the dictionary:",dict1["B"])
dict1["A"] = 9
print("Updateing the no. of people who got A grade from the dictionary:",dict1["A"])
dict1["S"] =  3
print("Adding the no. of students who got S grade from the dictionary:",dict1["S"])
print("Checking if any students are fail from the dictionary:", "F" in dict1)
```

Sample Input and Output:

```
{'A': 7, 'B': 16, 'C': 18, 'D': 23, 'F': 32}
Accessing the no. of people got B grade from the dictionary: 16
Updateing the no. of people who got A grade from the dictionary: 9
Adding the no. of students who got S grade from the dictionary: 3
Checking if any students are fail from the dictionary: True
```

Screenshot of the Execution:



```
{'A': 7, 'B': 16, 'C': 18, 'D': 23, 'F': 32}
Accessing the no. of people got B grade from the dictionary: 16
Updateing the no. of people who got A grade from the dictionary: 9
Adding the no. of students who got S grade from the dictionary: 3
Checking if any students are fail from the dictionary: True
```