UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 21/12/2020
S03018

Ex. No.: 2

Sum of two numbers

Aim

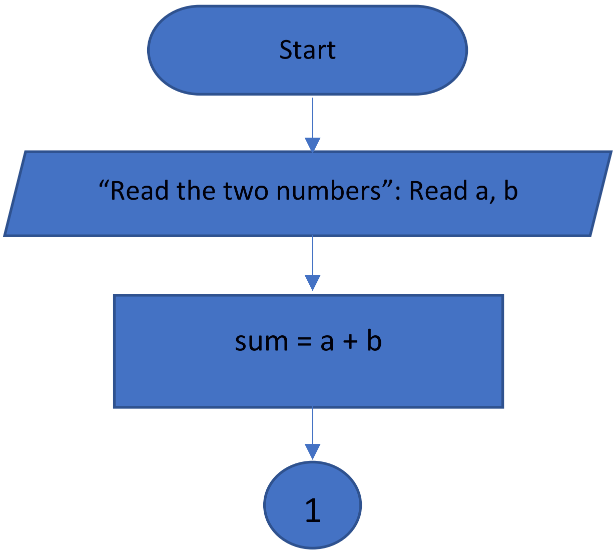
To find the sum of two numbers.

Algorithm

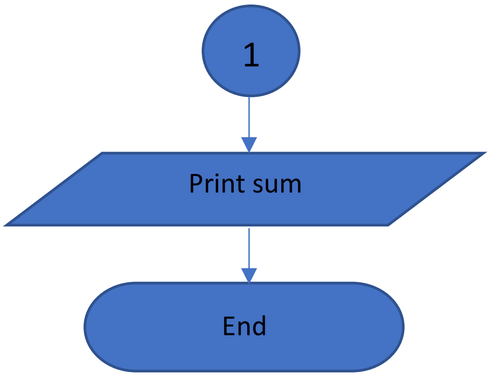
```
Input: Any two numbers
Output: Sum of the two numbers

Step 1: Start
Step 2: Read any two numbers.
Step 3: Add the two numbers.
Step 4: Store the sum in another variable.
Step 5: Print the sum.
Step 6: End
```

Flowchart



Ex. No.: 2



Program

```
# Python program to find the sum of two numbers provided by the user.

# To take integer input from the user
a=int(input("Enter the first number: "))
b=int(input("Enter the second number: "))
#To add the two numbers and store the sum
c=a+b
#To print the sum
print("Sum:",c)
```

Output

```
Input:

43, 52

Output:

95
```

```
Enter the first number: 43
Enter the second number: 52
Sum: 95
```

Results / Inferences

Program for finding the sum of two given numbers is written and executed.

Area of a triangle

Aim

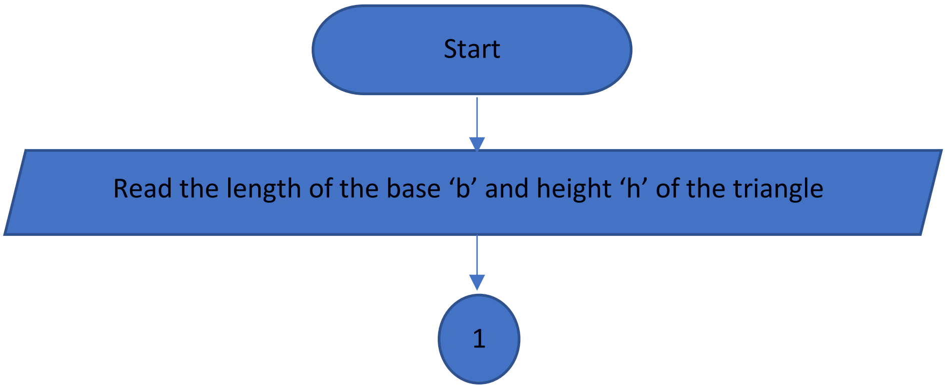
To find the area of a triangle.

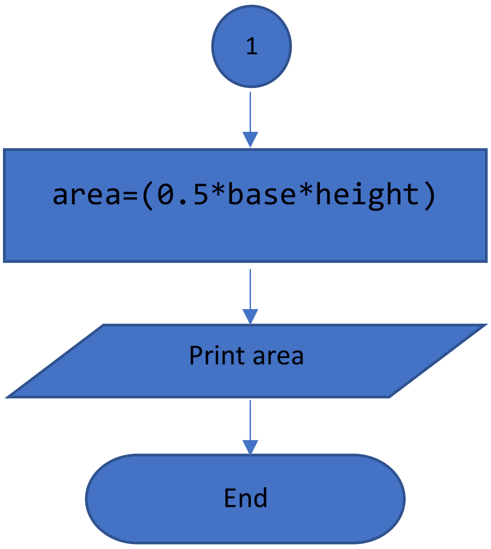
Algorithm

```
Input: Chosen base and height of the triangle
Output: Area of the triangle

Step 1: Start
Step 2: Read the length of the base and altitude of the triangle.
Step 3: Calculate the area with the formula area=(0.5*base*height).
Step 4: Store the value as the area of the triangle.
Step 5: Print the area.
Step 6: End
```

Flowchart


Start
Read the length of the base ‘b’ and height ‘h’ of the triangle
1



Program

```
# Python program to find the area of a given triangle of known base
and height.

# To take base and height as input from the user
print("Enter lengths of the base and the corresponding altitude:")
b=int(input("b="))
h=int(input("h="))
area=(b*h)/2
#To print the area
print("Area=",area)
```

Output

```
Input:

4, 3
```

Output:

6

```
Enter lengths of the base and the corresponding altitude:
b=4
h=3
Area= 6.0
```

Results / Inferences

Program for finding the area of a triangle is written and executed.

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 21/12/2020
S03018

Ex. No.: 2

Square root

Aim

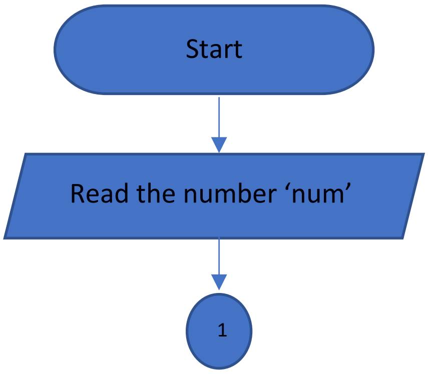
To find the square root of a number.

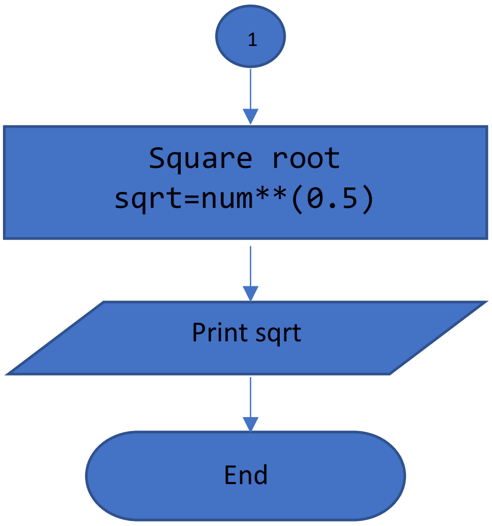
Algorithm

```
Input: Any number
Output: The square root of the number

Step 1: Start
Step 2: Read the number for which the square root is to be found.
Step 3: Calculate the value of the number to the power of half.
Step 4: Store the value as the square root.
Step 5: Print the square root of the number.
Step 6: End
```

Flowchart





Program

```
# Python program to find the square root of a given number.

# To take the number as input from the user
num=int(input("Enter the number: "))
nsqrt=num**0.5
# To print the square root of the number
print("Square root: ",nsqrt)
```

Output

```
Input:

4

Output:
```

```
2.0
```

```
Enter the number: 4
Square root:  2.0
```

Results / Inferences

Program for finding the square root of a number is written and executed.

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 21/12/2020
S03018

Ex. No.: 2

Roots of a Quadratic Equation

Aim

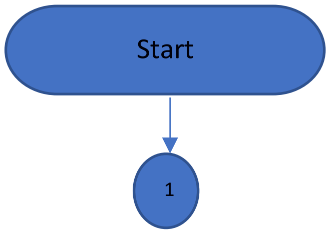
To find the solution of a quadratic equation.

Algorithm

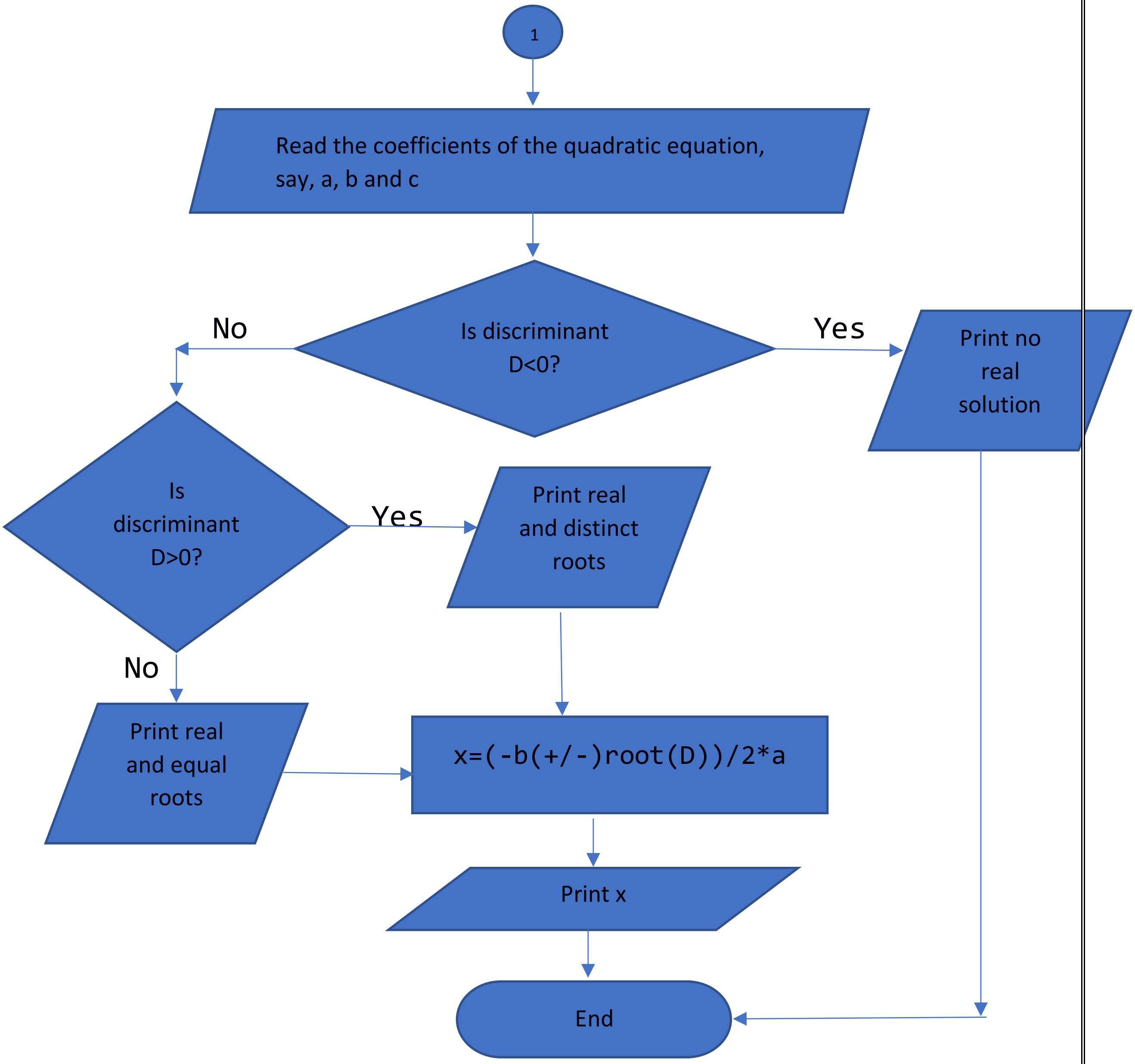
```
Input: Coefficients of the quadratic equation.
Output: The solution of the quadratic equation

Step 1: Start
Step 2: Read the coefficients of the quadratic equation.
Step 3: Using the coefficients, find the discriminant (D).
Step 4: If D<0, print that the quadratic equation does not have real solutions.
Step 5: If D=0, print that the quadratic equation has real and equal solutions.
Step 6: If D>0, print that the quadratic equation has real and distinct solutions.
Step 7: Use the quadratic equation formula to solve for x. (x=(-b(+/-)root(D))/2*a)
Step 8: Store the values of x and print them.
Step 9: End
```

Flowchart



Ex. No.: 2


1
Read the coefficients of the quadratic equation, say, a, b and c
Is discriminant D<0?
No
Yes
Print no real solution
Is discriminant D>0?
Yes
Print real and distinct roots
No
Print real and equal roots
x=(-b(+/-)root(D))/2*a
Print x
End

Program

```
# Python program to find the solutions of a given quadratic equation.
# To define a square root function
def sqrt(m): return m**0.5
# To take coefficients as input from the user
a=int(input("Coefficient of x^2: "))
b=int(input("Coefficient of x: "))
c=int(input("Constant: "))
# To calculate the discriminant
D=(b**2)-(4*a*c)
# To check if the discriminant value is lesser than, greater than or
equal to zero and solve accordingly
if D>0:
  soln1=((-b+sqrt(D))/(2*a))
  soln2=((-b-sqrt(D))/(2*a))
  print("The roots are real and distinct. The roots are ",soln1," and
",soln2,".")
elif D==0:
  soln=(-b/(2*a))
  print("The roots are real and equal. The roots are ",soln," and
",soln,".")
else:
  print("The discriminant is negative. The quadratic equation does
not have real solutions.")
```

Output

Input:

1, -5, 6

Output:

3, 2

```
Coefficient of x^2: 1
Coefficient of x: -5
```

```
Constant: 6
The roots are real and distinct. The roots are 3.0 and
2.0.
```

Input:

1, 2, 1

Output:

-1, -1

```
Coefficient of x^2: 1
Coefficient of x: 2
Constant: 1
The roots are real and equal. The roots are -1.0 and
-1.0.
```

Input:

1, 1, 1

Output:

```
Coefficient of x^2: 1
Coefficient of x: 1
Constant: 1
The discriminant is negative. The quadratic equation does
not have real solutions.
```

Results / Inferences

Program for finding the solutions of a quadratic equation is written and executed.

Krithika Swaminathan
Date: 21/12/2020
S03018

Ex. No.: 2

Temperature conversion

Aim

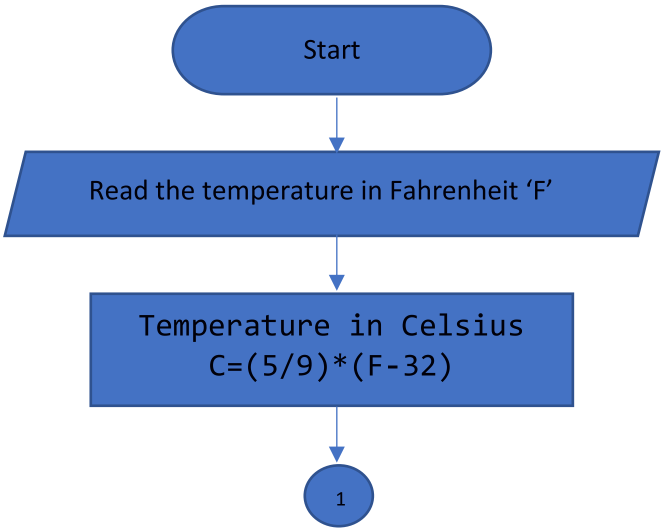
To convert the temperature from Fahrenheit to Celsius.

Algorithm

Input: Temperature in Fahrenheit
Output: Temperature in Celsius

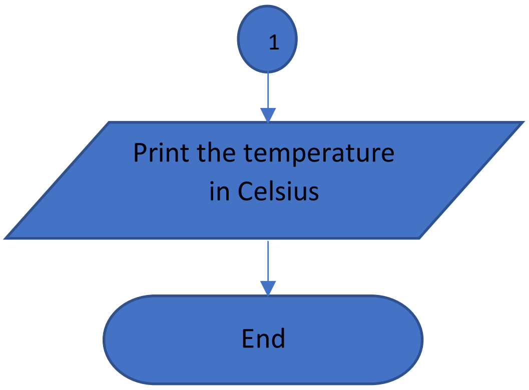
```
Step 1: Start
Step 2: Read the temperature in Fahrenheit.
Step 3: Use the formula C=(5/9)*(F-32) to calculate the temperature in Celsius.
Step 4: Print the temperature in Celsius.
Step 5: End
```

Flowchart



Krithika Swaminathan
Date: 21/12/2020
S03018

Ex. No.: 2



Program

```
# Python program to convert the temperature from Fahrenheit to
Celsius.

# To take temperature in Fahrenheit as input from the user
f=float(input("Enter temperature in Fahrenheit: "))
c=(5/9)*(f-32)
#To print the temperature in Celsius
print("Temperature is ",c," degree Celsius.")
```

Output

Input:

95

Output:

35

```
Enter temperature in Fahrenheit: 95
Temperature is 35.0 degree Celsius.
```

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 21/12/2020
S03018

Ex. No.: 2

Results / Inferences

Program for converting the given temperature from Fahrenheit to Celsius is written and executed.

Krithika Swaminathan
Date: 21/12/2020
S03018

Ex. No.: 2

Division

Aim

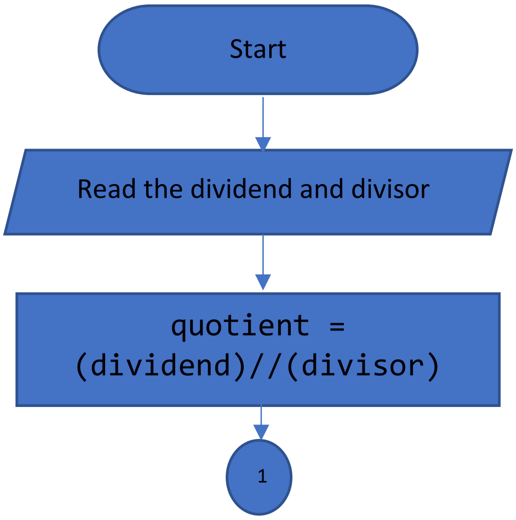
To find the quotient and remainder after a division.

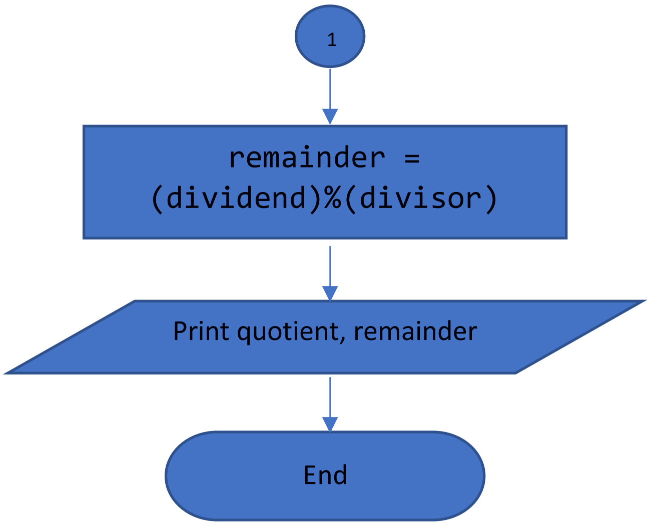
Algorithm

```
Input: Dividend and divisor
Output: Quotient and remainder

Step 1: Start
Step 2: Read the dividend and divisor.
Step 3: Divide the dividend by the divisor and store the floor of the result as the quotient.
Step 4: Store the modulus of the division as the remainder.
Step 5: Print the quotient and the remainder.
Step 6: End
```

Flowchart


Start
Read the dividend and divisor
quotient = (dividend)//(divisor)
1



Program

```
# Python program to find the quotient and remainder after a division.

# To take dividend and divisor as input from the user
dd=int(input("Dividend: "))
ds=int(input("Divisor: "))
#To get the quotient
q=dd//ds
#To get the remainder
r=dd%ds
#To print the quotient and remainder
print("Quotient: ",q,"\tRemainder: ",r)
```

Output

```
Input:

43, 5
```

Output:

```
8, 3
```

```
Dividend: 43
Divisor: 5
Quotient:  8    Remainder:  3
```

Results / Inferences

Program for finding the quotient and remainder of a division is written and executed.

Swapping two numbers

Aim

To swap two numbers using tuple assignment method and temporary variable method.

Algorithm

Input: Two numbers in a specific order
Output: The same numbers, but with the order swapped once, then swapped again to the original order.

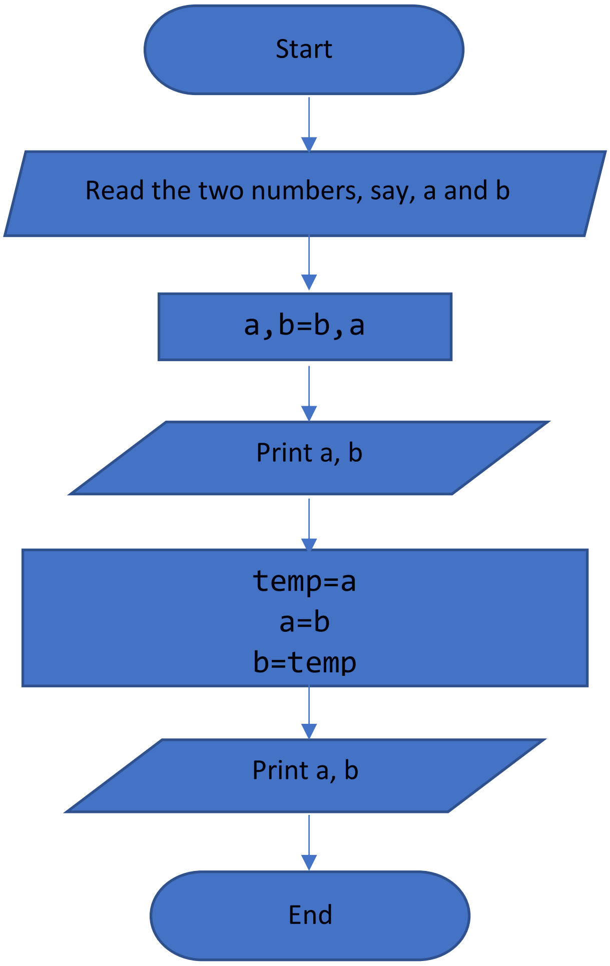
```
Step 1: Start
Step 2: Read two numbers and store them in two variables.
Step 3: To proceed with tuple assignment, interchange the order of the variables and assign the same variables to them in the original order. The numbers are now swapped.
Step 4: Print the numbers.
Step 5: To proceed with the third variable method, create a temporary variable.
Step 6: Store one of the numbers, say, from variable1, in the temporary variable.
Step 7: Store the number from variable2 in variable1.
Step 8: Restore the number from the temporary variable to variable2. Now, the numbers have been swapped again to the original order.
Step 9: Print the numbers.
Step 10: End
```

Flowchart

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 21/12/2020
S03018

Ex. No.: 2



Program

```
# Python program to swap two numbers using tuple assignment method
and temporary variable method.

# To take two numbers as input from the user
a=int(input("Enter first number: "))
```

```
b=int(input("Enter second number: "))
#Tuple assignment method
print("Swapping by tuple assignment method...")
a,b=b,a
#To print the swapped numbers
print("First number: ",a)
print("Second number: ",b)
#Temporary variable method
print("Swapping again by temp variable method...")
temp=a
a=b
b=temp
#To print the numbers swapped again to the original order
print("First number: ",a)
print("Second number: ",b)
```

Output

```
Input:

5, 9

Output:

9, 5 and 5, 9
```

```
Enter first number: 5
Enter second number: 9
Swapping by tuple assignment method...
First number:  9
Second number:  5
Swapping again by temp variable method...
First number:  5
Second number:  9
```

Results / Inferences

Program for swapping two numbers is written and executed.

Krithika Swaminathan
Date: 21/12/2020
S03018

Ex. No.: 2

Hypotenuse of a triangle

Aim

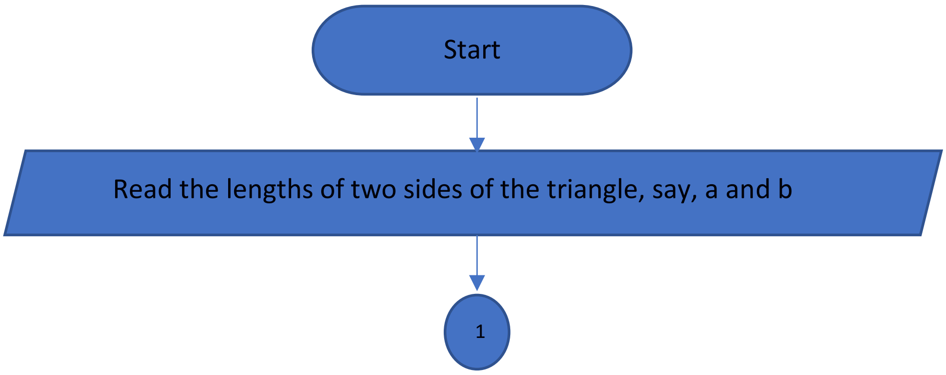
To find the length of the hypotenuse of a right-angled triangle.

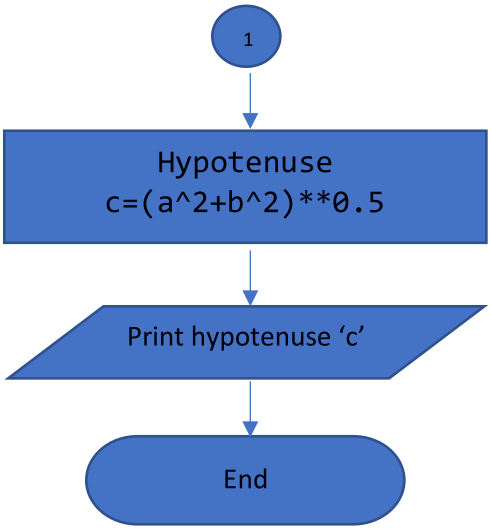
Algorithm

Input: Lengths of sides adjacent to the right angle
Output: Length of the hypotenuse

```
Step 1: Start
Step 2: Read the lengths of the two sides adjacent to the right angle of the triangle.
Step 3: Using the Pythagorean theorem, calculate the square of the length of the hypotenuse. (Hypotenuse^2 = Base^2 + Height^2)
Step 4: Calculate the hypotenuse from its square by raising it to the power of half.
Step 5: Print the length of the hypotenuse.
Step 6: End
```

Flowchart





Program

```
# Python program to find the length of the hypotenuse of a given
triangle of known lengths of other two sides.

# To take lengths of sides adjacent to the right angle as input from
the user
a=int(input("Enter the length of side a of the triangle:"))
b=int(input("Enter the length of side b of the triangle:"))
cs=(a*a)+(b*b)
c=cs**0.5
#To print the length of the hypotenuse
print("The length of the hypotenuse is: ",c)
```

Output

Input:

3, 4

Output:

5

```
Enter the length of side a of the triangle:3
Enter the length of side b of the triangle:4
The length of the hypotenuse is:  5.0
```

Results / Inferences

Program for finding the length of the hypotenuse of a right-angled triangle is written and executed.

Average of three marks

Aim

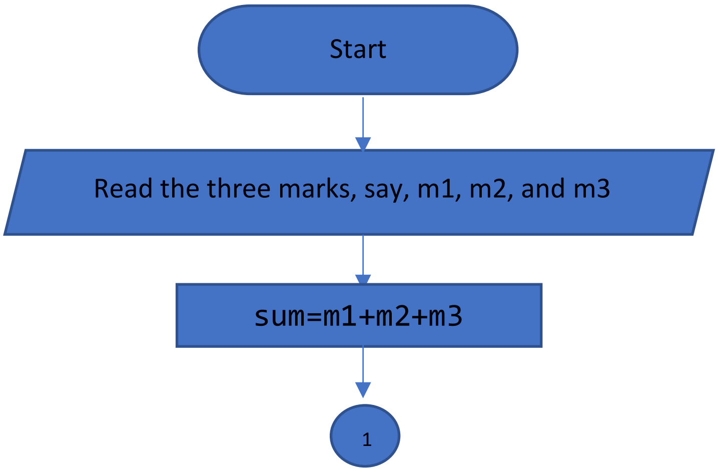
To find the average of three given marks.

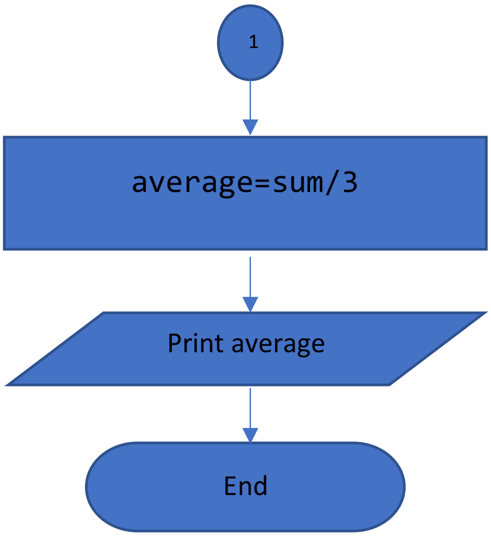
Algorithm

```
Input: The three marks
Output: The average of the three marks

Step 1: Start
Step 2: Read the three marks.
Step 3: Calculate the sum of the three marks.
Step 4: Divide the sum by 3 and store the result as the average.
Step 5: Print the average.
Step 6: End
```

Flowchart





Program

```
# Python program to find the average of three given marks.

# To take the marks as input from the user
print("Enter the three marks for calculation.")
m1=float(input("Mark1: "))
m2=float(input("Mark2: "))
m3=float(input("Mark3: "))
avg=round(((m1+m2+m3)/3),2)
#To print the average
print("Average: ",avg)
```

Output

```
Input:

94, 98, 96
```

Output:

```
96.0
```

```
Enter the three marks for calculation.
Mark1: 94
Mark2: 98
Mark3: 96
Average:  96.0
```

Results / Inferences

Program for finding the average of three given marks is written and executed.

Weight Conversion

Aim

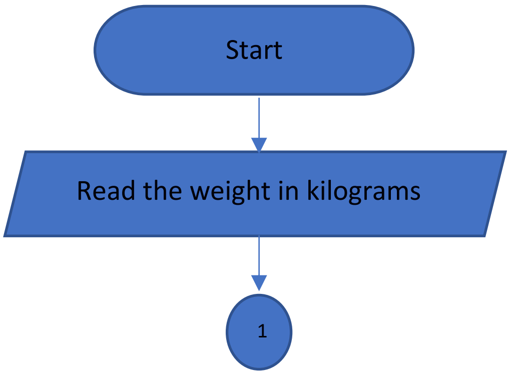
To convert the given weight from kilograms to pounds and tonnes.

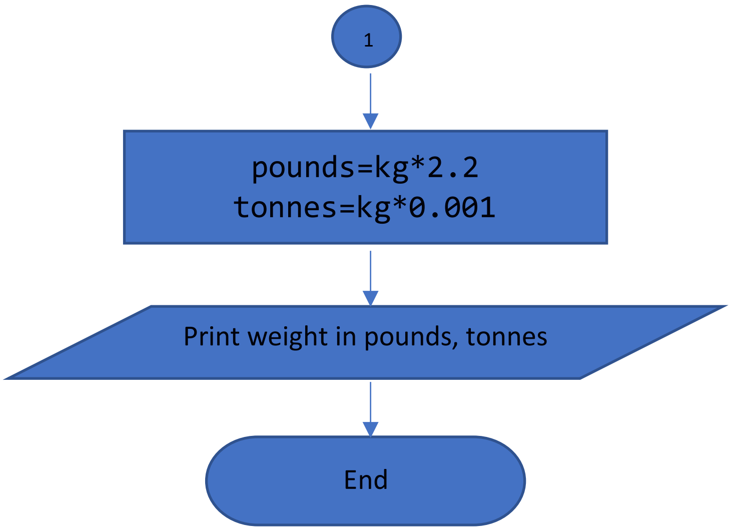
Algorithm

Input: The weight of the object in kilograms
Output: The weight of the object in pounds and tonnes

```
Step 1: Start
Step 2: Read the weight of the object in kilograms.
Step 3: Convert the weight to pounds using the formula that 1kg=2.20 pound.
Step 4: Print the weight in pounds.
Step 5: Convert the same weight to tonnes using the formula that 1kg=0.001 tonne.
Step 6: Print the weight in tonnes.
Step 7: End
```

Flowchart





Program

```
# Python program to convert the given weight from kilograms to pounds
and tonnes.

# To take weight in kilograms as input from the user
wk=float(input("Enter the weight in kg:"))
wp=wk*2.2
wt=wk*0.001
#To print the weight in pounds
print("Weight in pounds: ",wp)
#To print the weight in tonnes
print("Weight in tonnes: ",wt)
```

Output

Input:

2

Output:

4.4, 0.002

```
Enter the weight in kg:2
Weight in pounds:  4.4
Weight in tonnes:  0.002
```

Results / Inferences

Program for converting the given weight from kilograms to tonnes is written and executed.

Krithika Swaminathan
Date: 21/12/2020
S03018

Ex. No.: 2

Simple Interest

Aim

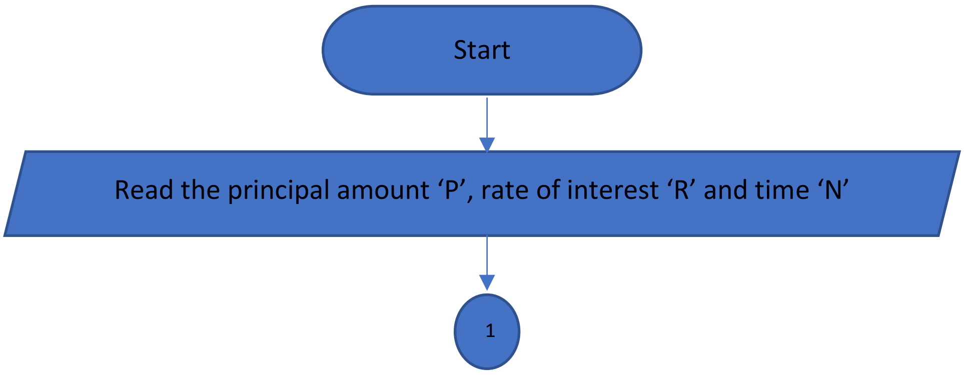
To calculate the simple interest.

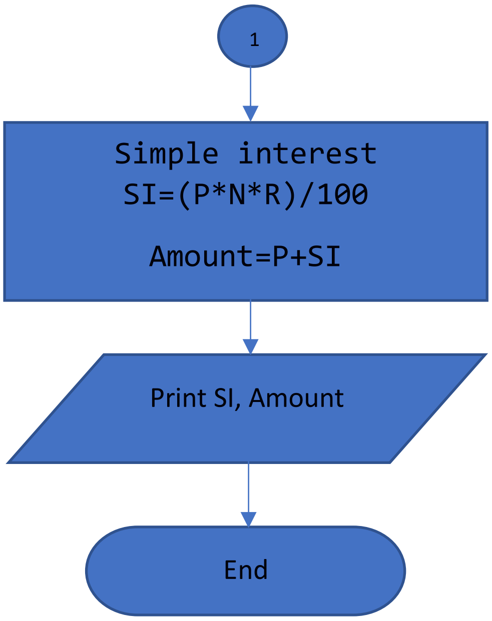
Algorithm

Input: The principal, rate of interest and the time period
Output: The simple interest and the final amount

```
Step 1: Start
Step 2: Read the principal amount (P), rate of interest (R) in
ppa and time period (T) to calculate the simple interest.
Step 3: Multiply P, R and T and divide the product by 100.
Store this value as the simple interest. (SI= PNR/100)
Step 4: Print the simple interest.
Step 5: If required, add the simple interest to the principal
and store as the final amount. Print the final amount.
Step 6: End
```

Flowchart


Start
Read the principal amount ‘P’, rate of interest ‘R’ and time ‘N’
1



Program

```
# Python program to calculate the simple interest.

# To take principal, rate and time as input from the user
print("Enter principal amount (in rupees), rate of interest (in % p.a.) and time (in years).")
p=float(input("Principal: "))
r=float(input("Rate: "))
n=float(input("Time: "))

# To calculate the simple interest
SI=(p*n*r)/100
# To calculate the total amount
A=p+SI
# To print the simple interest
print("Simple interest: Rs.",SI)
print("Final amount: Rs.",A)
```

Output

```
Input:

1000, 5, 2

Output:

100, 1100
```

```
Enter principal amount (in rupees), rate of interest (in
% p.a.) and time (in years).
Principal: 1000
Rate: 5
Time: 2
Simple interest: Rs. 100.0
Final amount: Rs. 1100.0
```

Results / Inferences

Program for calculating the simple interest is written and executed.

Krithika Swaminathan
Date: 21/12/2020
S03018

Ex. No.: 2

Gross Salary

Aim

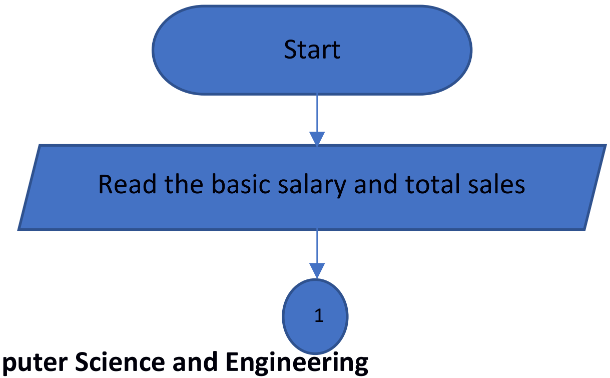
To calculate the gross salary of a medical representative.

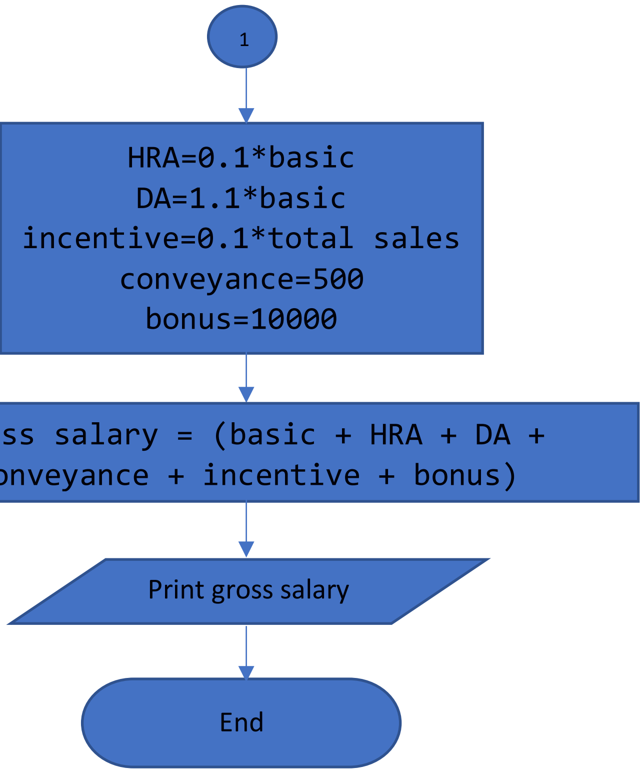
Algorithm

Input: The basic pay and total sales
Output: The gross salary

```
Step 1: Start
Step 2: Read the values of basic salary, total sales, conveyance and bonus, ensuring that conveyance=(fixed) Rs.500 and bonus=(fixed) Rs.1000.
Step 3: Calculate the HRA, DA and incentive based on the formulae HRA=(10% of basic), DA=(110% basic) and incentive=(10% total sales).
Step 4: Calculate the gross salary based on the formula that Gross salary = (basic + HRA + DA + Conveyance + incentive + bonus).
Step 5: Print the gross salary.
Step 6: End
```

Flowchart




1
HRA=0.1*basic
DA=1.1*basic
incentive=0.1*total sales
conveyance=500
bonus=10000
gross salary = (basic + HRA + DA + Conveyance + incentive + bonus)
Print gross salary
End

Program

```
# Python program to calculate the gross salary of a medical representative.

# To take basic pay and total sales in rupees as input from the user
bp=float(input("Basic pay: "))
ts=float(input("Total sales: "))
hra=0.1*bp
da=1.1*bp
con=500
inc=0.1*ts
bonus=1000
```

```
gp=bp+hra+da+con+inc+bonus
#To print the gross salary
print("Gross salary: Rs.",gp)
```

Output

```
Input:

10000, 2000

Output:

23700
```

```
Basic pay: 10000
Total sales: 2000
Gross salary: Rs. 23700.0
```

Results / Inferences

Program for calculating the gross salary of a medical representative is written and executed.

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 21/12/2020
S03018

Ex. No.: 2

Reversing a 2-digit number

Aim

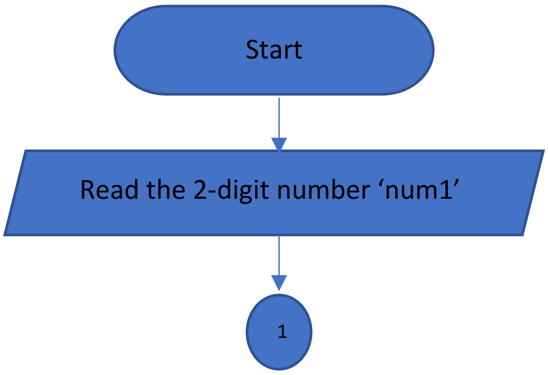
To reverse a 2-digit number and find the sum of the original and the reversed number.

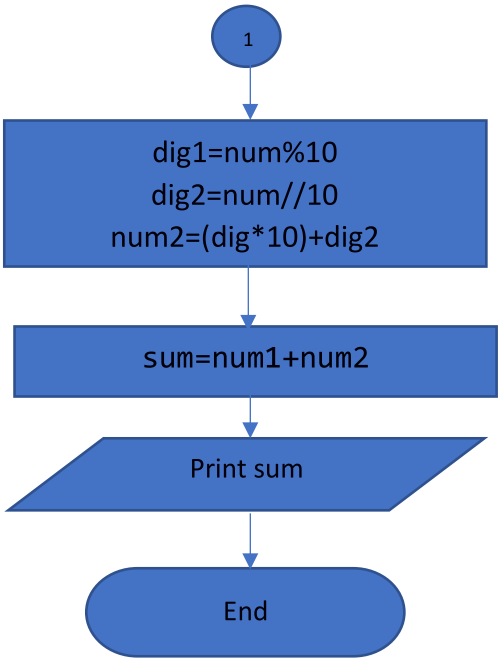
Algorithm

Input: A two-digit number
Output: The number with its digits reversed and the sum of the original number with the reversed number

```
Step 1: Start
Step 2: Read the 2-digit number.
Step 3: Split the number into its digits (by taking the remainder or using string slicing).
Step 4: Multiply the unit’s digit by 10 and add the result to the ten’s digit. Store the result as the reversed number.
Step 5: Add the original number to the reversed number and print the sum.
Step 6: End
```

Flowchart





Program

```
# Python program to reverse a 2-digit number and find the sum of the
original and the reversed number.

# To take a 2-digit number as input from the user
num=int(input("Enter a 2-digit number:"))
dig1=num%10
dig2=num//10
num2=(dig1*10)+dig2
#To print the reversed number
print("Reversed number: ",num2)
sum=num+num2
#To print the sum of the two numbers
print("Sum: ",sum)
```

Output

```
Input:

27

Output:

72, 99
```

```
Enter a 2-digit number:27
Reversed number:  72
Sum:  99
```

Results / Inferences

Program for reversing a 2-digit number and finding the sum of the original number and the reversed number is written and executed.