

LET US REVISE

- ☞ Statements are the instructions given to the computer to perform any kind of action.
- ☞ Python statements can be on one of these types : empty statement, single statement and compound statement.
- ☞ An empty statement is the statement that does nothing. Python offers **pass** statement as empty statement.
- ☞ Single executable statement forms a simple statement.
- ☞ A compound statement represents a group of statements executed as a unit.
- ☞ Every compound statement of Python has a header and an indented body below the header.
- ☞ Some examples of compound statements are : if statement, while statement etc.
- ☞ The flow of control in a program can be in three ways : sequentially (the sequence construct), selectively (the selection construct), and iteratively (the iteration construct).
- ☞ The sequence construct means statements get executed sequentially.
- ☞ The selection construct means the execution of statement(s) depending upon a condition-test.
- ☞ The iteration (looping) constructs mean repetition of a set-of-statements depending upon a condition-test.
- ☞ Python provides one selection statement if in many forms – if..else and if..elif..else.
- ☞ The if..else statement tests an expression and depending upon its truth value one of the two sets-of-action is executed.
- ☞ The if-else statement can be nested also i.e., an if statement can have another if statement.
- ☞ A sequence is a succession of values bound together by a single name. Some Python sequences are strings, lists and tuples.
- ☞ The range() function generates a sequence of list type.
- ☞ The statements that allow a set of instructions to be performed repeatedly are iteration statements.
- ☞ Python provides two looping constructs – **for** and **while**. The **for** is a counting loop and **while** is a conditional loop.
- ☞ The while loop is an entry-controlled loop as it has a control over entry in the loop in the form of test condition.
- ☞ Loops in Python can have **else** clause too. The **else** clause of a loop is executed in the end of the loop only when loop terminates normally.
- ☞ The **break** statement can terminate a loop immediately and the control passes over to the statement following the statement containing break.
- ☞ In nested loops, a **break** terminates the very loop it appears in.
- ☞ The **continue** statement abandons the current iteration of the loop by skipping over the rest of the statements in the loop-body. It immediately transfers control to the beginning of the next iteration of the loop.

Solved Problems

1. What is a statement ? What is the significance of an empty statement ?

Solution. A statement is an instruction given to the computer to perform any kind of action.

An empty statement is useful in situations where the code requires a statement but logic does not. To fill these two requirements simultaneously, empty statement is used.

Python offers **pass** statement as an empty statement.

2. If you are asked to label the Python loops as determinable or non-determinable, which label would you give to which loop ? Justify your answer.

Solution. The 'for loop' can be labelled as *determinable loop*, as number of its iterations can be determined before-hand as the size of the sequence, it is operating upon.

The 'while loop' can be labelled as *non-determinable loop*, as its number of iterations cannot be determined before-hand. Its iterations depend upon the result of a test-condition, which cannot be determined before-hand.

3. *There are two types of else clauses in Python. What are these two types of else clause ?*

Solution. The two types of Python else clauses are : (a) else in an if statement (b) else in a loop statement
The else clause of an if statement is executed when the condition of the if statement results into *false*.

The else clause of a loop is executed when the loop is terminating normally i.e., when its test-condition has gone *false* for a while loop or when the for loop has executed for the last value in sequence.

4. *Use the Python range() function to create the following list : [7, 3, -1, -5]*

Solution. `range (7, -6, -4)`

5. *Identify and correct the problem with following code :*

```
countdown = 10           # Count from 10 down to 1
while countdown > 0 :
    print (countdown, end = ' ')
    countdown - 1
print ("Finally.")
```

Solution. *It is an infinite loop.* The reason being, in the above while loop, there is an expression `countdown - 1`, which subtracts 1 from value of `countdown` BUT this statement is not updating the loop variable `countdown`, hence loop variable `countdown` always remains unchanged and hence it is an **infinite loop**.

The solution to above problem will be replacement of `countdown - 1` with following statement in the body of while loop :

```
countdown = countdown - 1
```

Now the while loop's variable will be updated in every iteration and hence loop will be a finite loop.

6. *Following code is meant to be an interactive grade calculation script for an entrance test that converts from a percentage into a letter grade :*

```
90 and above is A+,
80-90 is A,
60-80 is A-,
and everything below is fail (U know high standards).
```

The program should prompt the user 100 times for grades. Unfortunately, the program was written by a terrible coder (me :/), so there are numerous bugs in the code. Find all the bugs and fix them.

```
For i : range(1 .. 100)
    grade == float (input( "What\'s the grade percentage ?" ))
    if grade > 90
        print ("That's an A + !")
    if 80 > grade > 90:
        print (" " "An A is really good!" " ")
    elsif 60 < grade :
        print (You got an A-!)
    else grade < 60 :
        print ("Sorry, you need an A- to qualify!")
```


Solution. The corrected code is given below, with corrections marked in colour.

```
for i in range (0, 100) :
    grade = float (input( "What\'s the grade percentage ?" ))
    if grade >= 90 :
        print ("That's an A + ! ")
    if 80 <= grade < 90:
        print ("An A is really good!")          # although triple-quoted string is also OK
    elif 60 <= grade :
        print ("You got an A-!")
    else grade < 60:
        print ("Sorry, you need an A- to qualify!")
```

7. What is the output of following code ?

```
if (4 + 5 == 10) :
    print ("TRUE")
else :
    print ("FALSE")
print ("TRUE")
```

Solution. FALSE
TRUE

8. Following code contains an endless loop. Could you find out why ? Suggest a solution.

```
n = 10
answer = 1
while ( n > 0 ) :
    answer = answer + n ** 2
    n = n + 1
print (answer)
```

Solution. For a while loop to terminate, it is necessary that its condition becomes false at one point. But in the above code, the test-condition of *while* loop will never become *false*, because the condition is $n > 0$ and n is always incrementing ; thus n will always remain > 0 . Hence it will repeat endlessly.

There are *two* possible solutions :

(i) change the condition to some reachable limit (as per update expression) e.g.,
while (n < 100) :

(ii) change the updation equation of loop variable so that it makes the condition *false* at some point of time, e.g.,

```
while (n > 0) :
    :
    n = n - 1
```

9. Consider below given two code fragments. What will be their outputs if the inputs are entered in the given order are (i) 'Jacob' (ii) '12345' (iii) 'END' (iv) 'end' ?

```
(a) name = " "
while True :
    name = input( "Enter name ('end' to exit):")
    if name == "end" :
        break
    print ("Hello", name)
else :
    print ("Wasn't it fun ?")
```

```
(b) name = " "
while name != "end" :
    name = input( "Enter name ('end' to exit):" )
    if name == "end" :
        pass
    print ("Hello", name)
else :
    print ("Wasn't it fun ?")
```

Solution. Code (a) will print

```
for input (i)
    Hello Jacob
for input (ii)
    Hello 12345
for input (iii)
    Hello END
for input (iv)
    no output will be printed
```

Code (b) will print

```
for input (i)
    Hello Jacob
for input (ii)
    Hello 12345
for input (iii)
    Hello END
for input (iv)
    Hello end
    Wasn't it fun ?
```

10. Write Python code to add the odd numbers up to (and including) a given value N and print the result.

Solution.

```
N = int (input ('Enter number'))
sum = 0
i = 1
while i <= N :
    sum = sum + i
    i = i + 2
print (sum)
```

11. Consider the following Python program :

```
N = int(input ( "Enter N : " ))
i = 1
sum = 0
while i < N :
    if i % 2 == 0 :
        sum = sum + i
    i = i + 1
print (sum)
```

- (a) What is the output when the input value is 5 ? (b) What is the output when the input value is 0 ?

Solution. (a) 6 (b) 0

12. Consider the following Python program, intended to calculate sums :

```
number = int(input ("Enter number"))
n, result = number, 1
```

```

while True or n :
    result = result + n
    n = n - 1
print ("sums =", result)

```

- (a) What happens when the user enters a value of 5 ? (b) How would you fix this program ?

Solution. The problem is that the program will repeat infinitely.

The problem lies with the condition of while loop this condition will never go false.

Correct condition will be any of these :

```

while n :
    :
or while n > 0 :
    :

```

13. Write a program to find lowest and second lowest number from the 10 numbers input.

Solution.

```

# to find lowest and second lowest integer from 10 integers
small = smaller = 0
for i in range(10):
    n = int(input("Enter number :"))
    if i == 0 :                               # first number read
        small = n
    elif i == 1 :                             # second number read
        if n <= small :
            smaller = n
        else:
            smaller = small
            small = n
    else:                                     # for every integer read 3rd integer onwards
        if n < smaller :
            small = smaller
            smaller = n
        elif n < small :
            small = n

print("The lowest number is : ", smaller)
print("The second lowest number is : ", small)

```

Sample Run

```

The lowest number is : -6
The second lowest number is : -4

```

The sample run of above program (from the ten input numbers as :

-6, 13, 20, -3, 15, 18, 99, -4, 11, 23) has been given above.

14. Number Guessing Game – This program generates a random number between 1 and 100 and continuously asks the user to guess the number, giving hints along the way.

Solution.

```

import random
secretNum = random.randint(1,100)           #generate a random integer between [1 100]
guessNumString = int(input ("Guess a number between 1 and 100 inclusive:"))

```

```

while guessNum != secretNum :
    if guessNum < secretNum :
        print ("Your guess is too low.")
    else:
        print ("Your guess is too high.")
    guessNum = int(input ( "Guess a number between 1 and 100 inclusive:" ))
print ("Congratulations! You guessed the number correctly.")

```

15. Write a Python script to print Fibonacci series' first 20 elements. Some initial elements of a Fibonacci series are :

0 1 1 2 3 5 8...

Solution.

```

first = 0
second = 1
print (first)
print (second)
for a in range(1, 19) :
    third = first + second
    print (third)
    first, second = second, third

```

16. Write a Python script to read an integer > 1000 and reverse the number.

Solution.

```

num = int(input( "Enter an integer (>1000) :"))
tnum = num
reverse = 0
while tnum :
    digit = tnum % 10
    tnum = tnum // 10
    reverse = reverse * 10 + digit
print ("Reverse of", num , "is", reverse)

```

17. Write a Python script to generate divisors of a number.

Solution.

```

num = int(input("Enter an integer :" ))
mid = num /2
print ("The divisors of" num, "are :")
for a in range (2, mid + 1) :
    if num % a == 0 :
        print (a, end = ' ')
    else :
        print ("-End-")

```

18. Input three angles and determine if they form a triangle or not.

Solution.

```

#check triangle from angles
angle1 = angle2 = angle3 = 0
angle1 = float(input("Enter angle 1 : "))

```

```

angle2 = float(input("Enter angle 2 : "))
angle3 = float(input("Enter angle 3 : "))
if (angle1 + angle2 + angle3) == 180 :
    print("These angles form a triangle.")
else:
    print("These angles do not form a triangle.")

```

19. Numbers in the form $2^n - 1$ are called Mersenne Numbers, e.g., $2^1 - 1 = 1, 2^2 - 1 = 3, 2^3 - 1 = 7$. Write a Python script that displays first ten Mersenne numbers.

Solution.

```

#program to print mersenne numbers
print("First 10 Mersenne numbers are:")
for a in range(1, 11):
    mersnum = 2 ** a - 1
    print(mersnum, end = " ")
print()

```

Sample Run

First 10 Mersenne numbers are:
1 3 7 15 31 63 127 255 511 1023

20. Write a program to calculate BMI of a person after inputting its weight in kgs and height in meters and then print the **Nutritional Status** as per following table :

Nutritional Status	WHO criteria BMI cut-off
Underweight	< 18.5
Normal	18.5 – 24.9
Overweight	25 – 29.9
Obese	≥ 30

Solution.

```

weight_in_kg = float(input("Enter weight in kg :"))
height_in_meter = float(input("Enter height in meters :"))
bmi = weight_in_kg / ( height_in_meter * height_in_meter)
print("BMI is : ", bmi, end = " ")
if bmi < 18.5 :
    print("... Underweight")
elif bmi < 25 :
    print("... Normal")
elif bmi < 30 :
    print("... Overweight")
else:
    print("... Obese")

```

21. To calculate the personal loan EMI, the following formula is used :

$$EMI = \frac{[P \times R \times (1 + R)^N]}{[(1 + R)^N - 1]}$$

where

- ◆ EMI is the equated monthly installment
- ◆ P is the principal or the amount that is borrowed as a loan

- ❖ **R** is the rate of interest that is levied on the loan amount (the interest rate should be a monthly rate)
- ❖ **N** is the tenure of repayment of the loan or the number of monthly installments that you will pay (tenure should be in months)

Using the above formula, write a Python program that calculates EMI for a given loan, rate-of-interest, number-of-installments.

Solution.

```
P = float(input("Enter principal amount:"))
R = float(input("Enter rate of interest per annum : " ))
MR = R / (12 * 100)          # monthly rate of interest
N = int(input("Enter the number of installments :"))
EMI = (P * MR * (1 + MR) ** N) / ( ((1 + MR) ** N) - 1 )

print("\tEMI Calculation")
print("Principal Amount :", P)
print("Rate of Interest (per annum) :", R)
print("Number of installments :", N)
print("EMI amount :", EMI)
```

```
Enter principal amount: 165000
Enter rate of interest per annum :16
Enter the number of installments : 20

EMI Calculation
Principal Amount : 165000.0
Rate of Interest (per annum) : 16.0
Number of installments : 20
EMI amount : 9453.387710620096
```

GLOSSARY

Block	A group of consecutive statements having same indentation level.
Body	The block of statements in a compound statement that follows the header.
Empty statement	A statement that appears in the code but does nothing.
Infinite Loop	A loop that never ends. <i>Endless loop.</i>
Iteration Statement	Statement that allows a set of instructions to be performed repeatedly.
Jump Statement	Statement that unconditionally transfers program control within a function.
Looping Statement	Iteration statement. Also called a loop.
Nesting	Same program-construct within another of same type of construct.
Nested Loop	A loop that contains another loop inside its body.
Suite	Block.

Assignments

Type A : Short Answer Questions/Conceptual Questions

1. What is the common structure of Python compound statements ?
2. What is the importance of the three programming constructs ?
3. What is empty statement ? What is its need ?
4. Which Python statement can be termed as empty statement ?

For
Solutions for
Selected Questions



Scan
QR Code

5. What is an algorithm ?
6. What is a flowchart ? How is it useful ?
7. Draw flowchart for displaying first 10 odd numbers.
8. What is entry-controlled loop ?
9. What are the four elements of a while loop in Python ?
10. What is the difference between else clause of if-else and else clause of Python loops ?
11. In which cases, the else clause of a loop does not get executed ?
12. What are jump statements ? Name them.
13. How and when are named conditions useful ?
14. What are endless loops ? Why do such loops occur ?

Type B : Application Based Questions

1. Rewrite the following code fragment that saves on the number of comparisons :

```
if (a == 0) :
    print ("Zero")
if (a == 1) :
    print ("One")
if (a == 2) :
    print ("Two")
if (a == 3) :
    print ("Three")
```
2. Under what conditions will this code fragment print "water" ?

```
if temp < 32 :
    print ("ice")
elif temp < 212:
    print ("water")
else :
    print ("steam")
```
3. What is the output produced by the following code ?

```
x = 1
if x > 3 :
    if x > 4 :
        print ("A", end = ' ')
    else :
        print ("B", end = ' ')
elif x < 2:
    if (x != 0):
        print ("C", end = ' ')
print ("D")
```
4. What is the error in following code ? Correct the code :

```
weather = 'raining'
if weather = 'sunny' :
    print ("wear sunblock")
elif weather = "snow":
    print ("going skiing")
else :
    print (weather)
```

5. What is the output of the following lines of code ?

```
if int('zero') == 0 :
    print ("zero" )
elif str(0) == 'zero' :
    print (0)
elif str(0) == '0 ' :
    print (str(0))
else:
    print ("none of the above")
```

6. Find the errors in the code given below and correct the code :

```
if n == 0
    print ("zero")
elif : n == 1
    print ("one")
elif
    n == 2:
    print ("two")
else n == 3:
    print ("three")
```

7. What is following code doing ? What would it print for input as 3 ?

```
n = int (input( "Enter an integer:" ))
if n < 1 :
    print ("invalid value")
else :
    for i in range(1, n + 1):
        print (i * i)
```

8. How are following two code fragments different from one another ? Also, predict the output of the following code fragments :

```
(a) n = int(input( "Enter an integer:" ))
    if n > 0 :
        for a in range(1, n + n ) :
            print (a / (n/2))
    else :
        print ("Now quitting")
```

```
(b) n = int (input( "Enter an integer:"))
    if n > 0 :
        for a in range(1, n + n) :
            print (a / (n/2))
    else :
        print ("Now quitting")
```

9. Rewrite the following code fragments using for loop :

```
(a) i = 100
    while (i > 0) :
        print (i)
        i -= 3
```

```
(b) while num > 0 :
        print (num % 10)
        num = num/10
```

```
(c) while num > 0 :
        count += 1
        sum += num
        num -= 2
        if count == 10 :
            print (sum/float(count))
            break
```

10. Rewrite following code fragments using while loops :

```
(a) min = 0
    max = num
    if num < 0 :
        min = num
        max = 0          # compute sum of integers from min to max
    for i in range(min, max + 1):
        sum += i
```

```
(b) for i in range(1, 16) :
    if i % 3 == 0 :
        print (i)
(c) for i in range(4) :
    for j in range(5):
        if i + 1 == j or j + i == 4 :
            print ("+", end = ' ')
        else :
            print ("o", end = ' ')
    print ()
```

11. Predict the output of the following code fragments :

```
(a) count = 0
while count < 10:
    print ("Hello")
    count += 1
```

```
(c) keepgoing = True
x = 100
while keepgoing :
    print (x)
    x = x - 10
    if x < 50 :
        keepgoing = False
```

```
(f) for x in range(5):
    print (x)
```

```
(i) for z in range(-500, 500, 100):
    print (z)
```

```
(k) x = 10
y = 5
for i in range(x-y * 2):
    print ("% ", i)
```

```
(m) for x in range(3):
    for y in range(4):
        print (x, y, x + y)
```

```
(b) x = 10
y = 0
while x > y:
    print (x, y)
    x = x - 1
    y = y + 1
```

```
(d) x = 45
while x < 50 :
    print (x)
```

```
(e) for x in [1,2,3,4,5]:
    print (x)
```

```
(g) for p in range(1,10):
    print (p)
```

```
(h) for q in range(100,50,-10):
    print (q)
```

```
(j) for y in range(500,100,100):
    print (" * ", y)
```

```
(l) for x in [1,2,3]:
    for y in [4, 5, 6]:
        print (x, y)
```

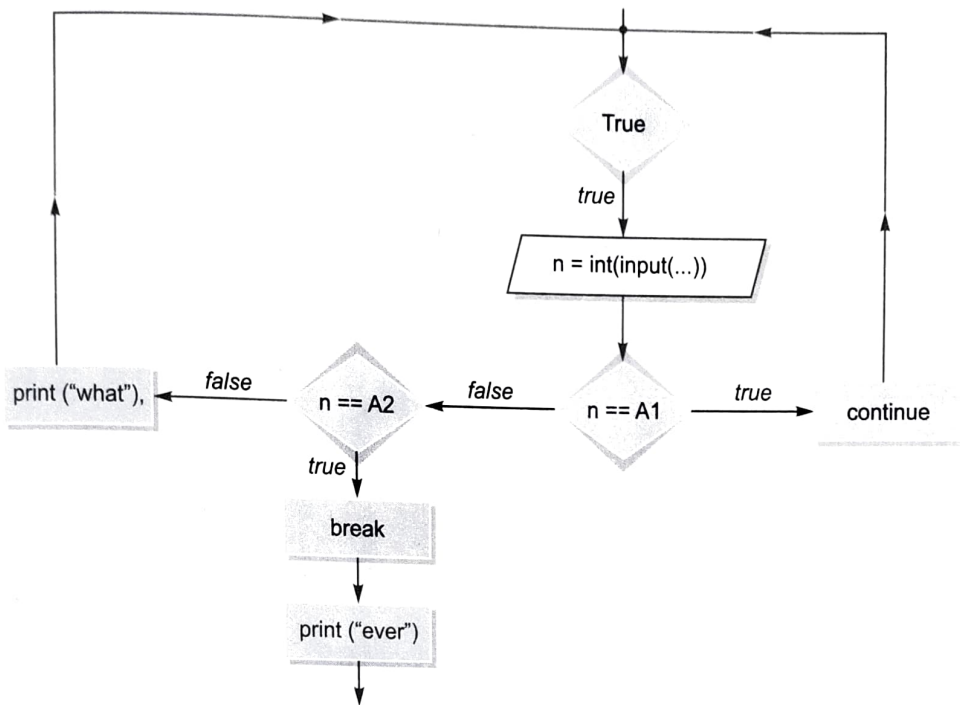
```
(n) c = 0
for x in range(10):
    for y in range(5):
        c += 1
    print (c)
```

12. What is the output of the following code ?

```
for i in range(4):
    for j in range(5):
        if i + 1 == j or j + i == 4:
            print ("+", end = ' ')
        else:
            print ("o", end = ' ')
    print()
```

13. In the nested for loop code above (Q. 12), how many times is the condition of the if clause evaluated

14. Which of the following Python programs implement the control flow graph shown ?



(a) `while True :`
 `n = int(input("Enter an int:"))`
 `if n == A1 :`
 `continue`
 `elif n == A2 :`
 `break`
 `else :`
 `print ("what")`
 `else :`
 `print ("ever")`

(b) `while True :`
 `n = int(input("Enter an int: "))`
 `if n == A1 :`
 `continue`
 `elif n == A2 :`
 `break`
 `else :`
 `print ("what")`
 `print ("ever")`

(c) `while True :`
 `n = int(input("Enter an int: "))`
 `if n == A1 :`
 `continue`
 `elif n == A2 :`
 `break`
 `print ("what")`
 `print ("ever")`

Type C : Programming Practice/Knowledge based Questions

- Write a Python script that asks the user to enter a length in centimetres. If the user enters a negative length, the program should tell the user that the entry is invalid. Otherwise, the program should convert the length to inches and print out the result. There are 2.54 centimetres in an inch.
- A store charges ₹ 120 per item if you buy less than 10 items. If you buy between 10 and 99 items, the cost is ₹ 100 per item. If you buy 100 or more items, the cost is ₹ 70 per item. Write a program that asks the user how many items they are buying and prints the total cost.
- Write a program that asks the user for two numbers and prints **Close** if the numbers are within .001 of each other and **Not close** otherwise.

4. A year is a leap year if it is divisible by 4, except that years divisible by 100 are not leap years unless they are also divisible by 400. Write a program that asks the user for a year and prints out whether it is a leap year or not.
5. Write a program to input length of three sides of a triangle. Then check if these sides will form a triangle or not. (Rule is : $a + b > c$; $b + c > a$; $c + a > b$)
6. Write a short program to input a digit and print it in words.
7. Write a short program to print first n odd numbers in descending order.
8. Write a short program to print the following series :
 - (i) 1 4 7 10 40.
 - (ii) 1 - 4 7 - 10 - 40
9. Write a short program to find average of list of numbers entered through keyboard.
10. Write a program to take an integer a as an input and check whether it ends with 4 or 8. If it ends with 4, print "ends with 4", if it ends with 8, print "ends with 8", otherwise print "ends with neither".
11. Write a program to take N ($N > 20$) as an input from the user. Print numbers from 11 to N . When the number is a multiple of 3, print "Topsy", when it is a multiple of 7, print "Topsy". When it is a multiple of both, print "TopsyTopsy".
12. Write a short program to find largest number of a list of numbers entered through keyboard.
13. Write a program to input N numbers and then print the second largest number.
14. Given a list of integers, write a program to find those which are palindromes. For example, the number 4321234 is a palindrome as it reads the same from left to right and from right to left.
15. Write a Python program to print every integer between 1 and n divisible by m . Also report whether the number that is divisible by m is even or odd.
16. Write a program to accept the age of n employees and count the number of persons in the following age group : (i) 26 - 35 (ii) 36 - 45 (iii) 46 - 55.
17. Write a program using nested loops to produce a rectangle of *s with 6 rows and 20 *s per row.
18. Given three numbers A , B and C , write a program to write their values in an ascending order. For example, if $A = 12$, $B = 10$, and $C = 15$, your program should print out :

Smallest number = 10
Next higher number = 12
Highest number = 15
19. Ask the user to enter a temperature in Celsius. The program should print a message based on the temperature :
 - ▲ If the temperature is less than -273.15 , print that the temperature is invalid because it is below absolute zero.
 - ▲ If it is exactly -273.15 , print that the temperature is absolute 0.
 - ▲ If the temperature is between -273.15 and 0, print that the temperature is below freezing.
 - ▲ If it is 0, print that the temperature is at the freezing point.
 - ▲ If it is between 0 and 100, print that the temperature is in the normal range.
 - ▲ If it is 100, print that the temperature is at the boiling point.
 - ▲ If it is above 100, print that the temperature is above the boiling point.
20. Write a program that inputs cost price and selling price for 10 items. The program then prints if the overall profit occurred or loss occurred.
21. Modify previous question so that the program now also prints profit/loss for individual items along with overall profit/loss.