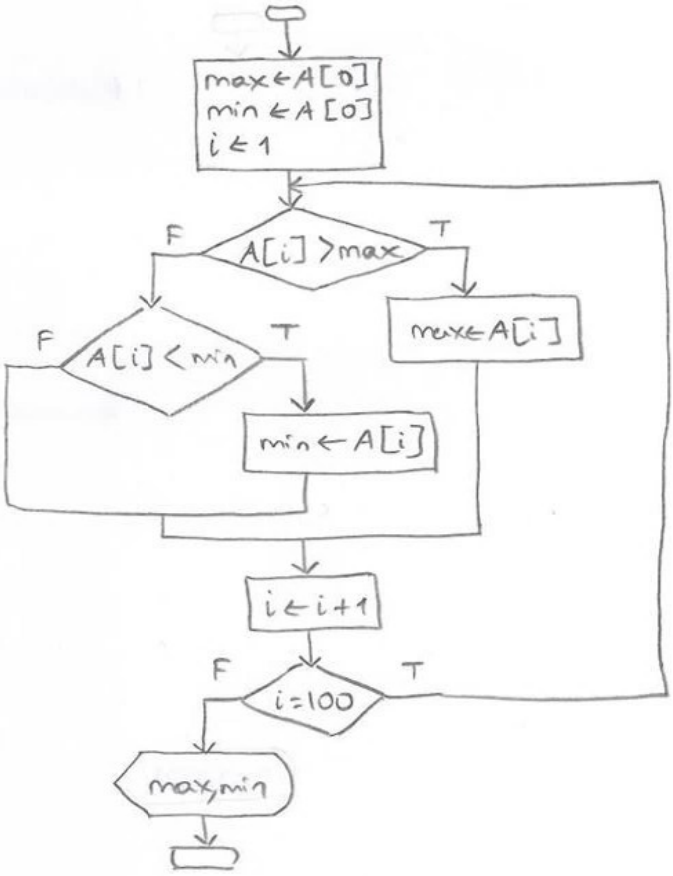
(11) Find the maximum and minimum of 100 numbers (A[0..99])

```
max ← A[0]
min ← A[0]
i ← 1
repeat
   ( if ( A[i] > max ) then
        ( max ← A[i] )
     else (if (A[i] < min) then
             (min ← A[i])
          )
   i ← i + 1
   ) until ( i = 100)
Print max, min
```

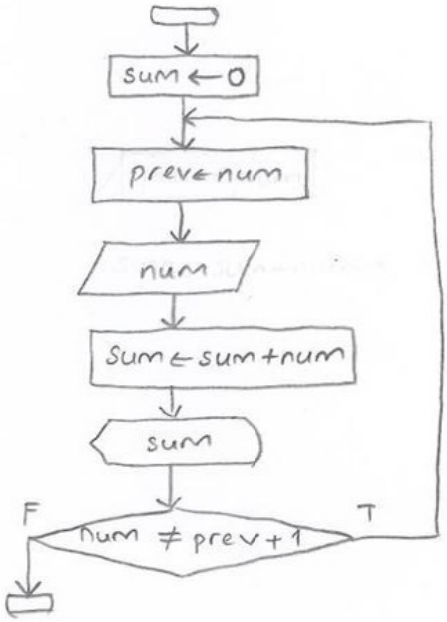


(12) Print the sum of the number that the user enters. Exit the program, if two consecutive numbers are entered after printing the sum.

sum

num

prev



ALGORITHMS and FLOWCHARTS

The study of algorithms is the cornerstone of computer science It is fundamentally important.

An algorithm, is an ordered set of unambiguous and well-defined instructions that performs some task and halts in finite time.

- an ordered set : you can number the steps.
- unambiguous : each instruction is clear, do-able and can be done without difficulty. (does not require creative skills)
- performs some task
- halts in finite time : algorithms terminate!

- Adding Two Numbers

Java :

```
public static void main (String args[]) {
    System.out.println("Enter the first number:");
    int num1 = System.in.read();
    System.out.println("Enter the second number:");
    int num2 = System.in.read();
    int sum = Integer.parseInt(a) +
              Integer.parseInt(b);
    System.out.println("The sum is " + sum);
}
```

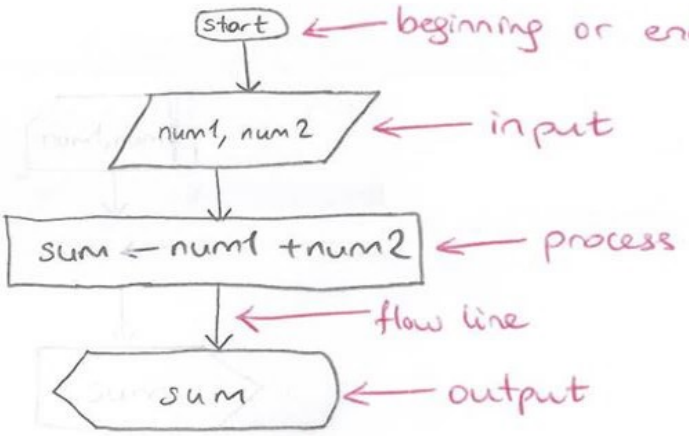
Natural Language :

1. Read the first number.
2. Read the second number.
3. Add the two numbers.
4. Print the result.

Pseudocode :

```
Read num1.
Read num2.
sum ← num1 + num2
Print sum.
```

Flowchart :



A pseudocode is a notational system in which ideas can be expressed informally during the algorithm development process.

1. saving a computed value - assignment statements

name ← expression

(assign name, the value of expression)

c ← a + b

2. conditional operations

if (condition) then (activity) else (activity)

if (condition) then (activity)

if (sales have decreased) then (lower the price by 5%)

if (a > 0) then (x ← x + 1) else (x ← x - 1)

3. Iterative operations

while (condition) do (activity)

repeat (activity) until (condition)

Indentation :

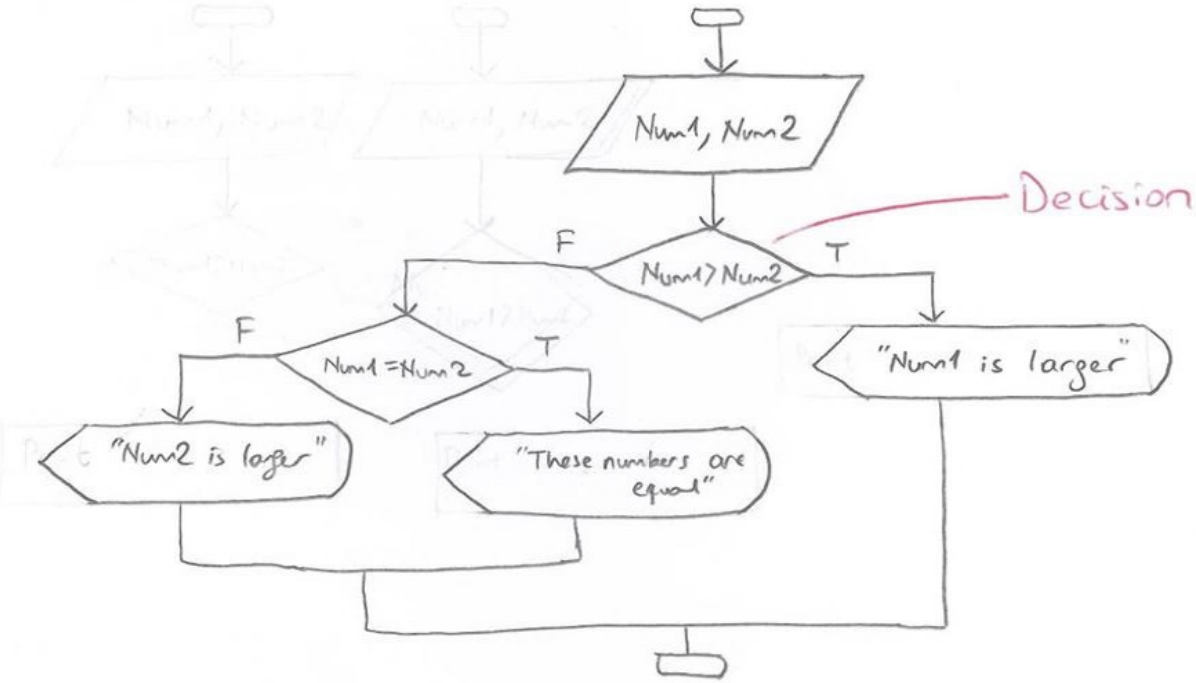
if (not raining) then (if (temperature = hot) then (go swimming) else (play golf)) else (watch television)

```
if (not raining)
   then ( if (temperature = hot)
             then (go swimming)
             else (play golf)
        )
   else (watch television)
```

Examples

① Read two numbers and print the larger. If the numbers are equal, print: "These numbers are equal".

```
Read Num1 and Num2
if (Num1 > Num2)
    then ( Print "Num1 is larger")
    else ( if (Num1 = Num2)
              then (Print "These numbers are equal")
              else (Print "Num2 is larger")
         )endif
```

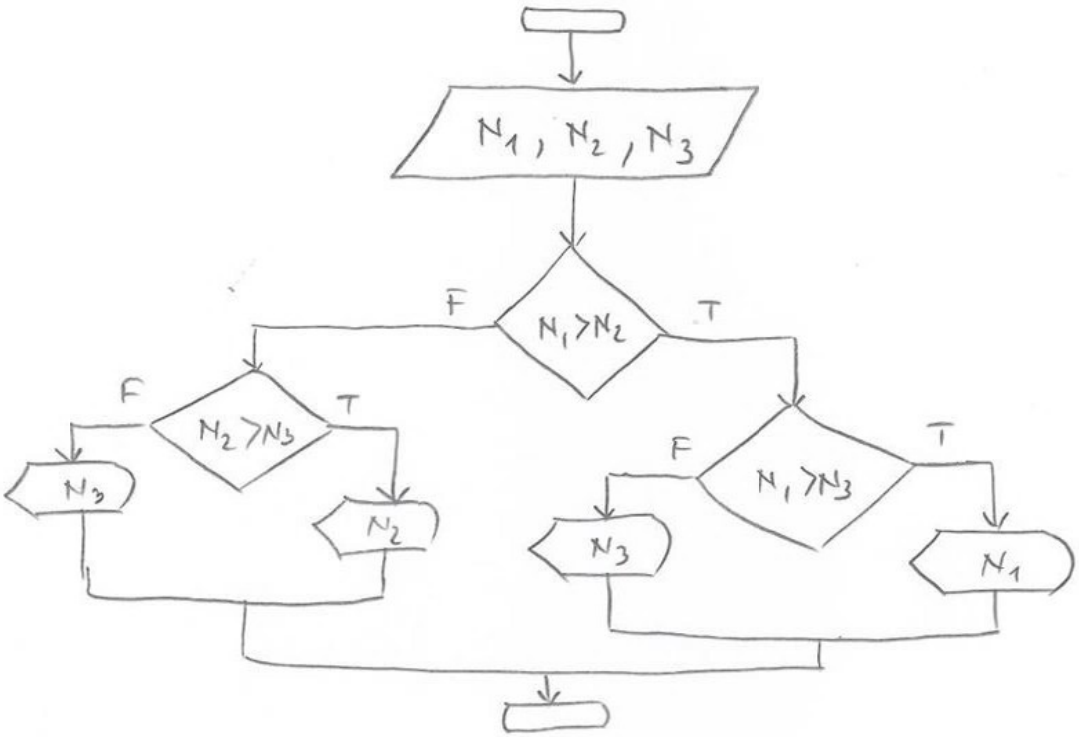



```
Read Num1 and Num2
if (Num1 > Num2)
    then ( Print "Num1 is larger")
if (Num1 = Num2)
    then (Print "These numbers are equal")
    else ( if (Num1 < Num2)
              then ( Print "Num2 is larger")
         )
```

Compare the two algorithms in terms of performance (number of operations, comparisons, etc.)

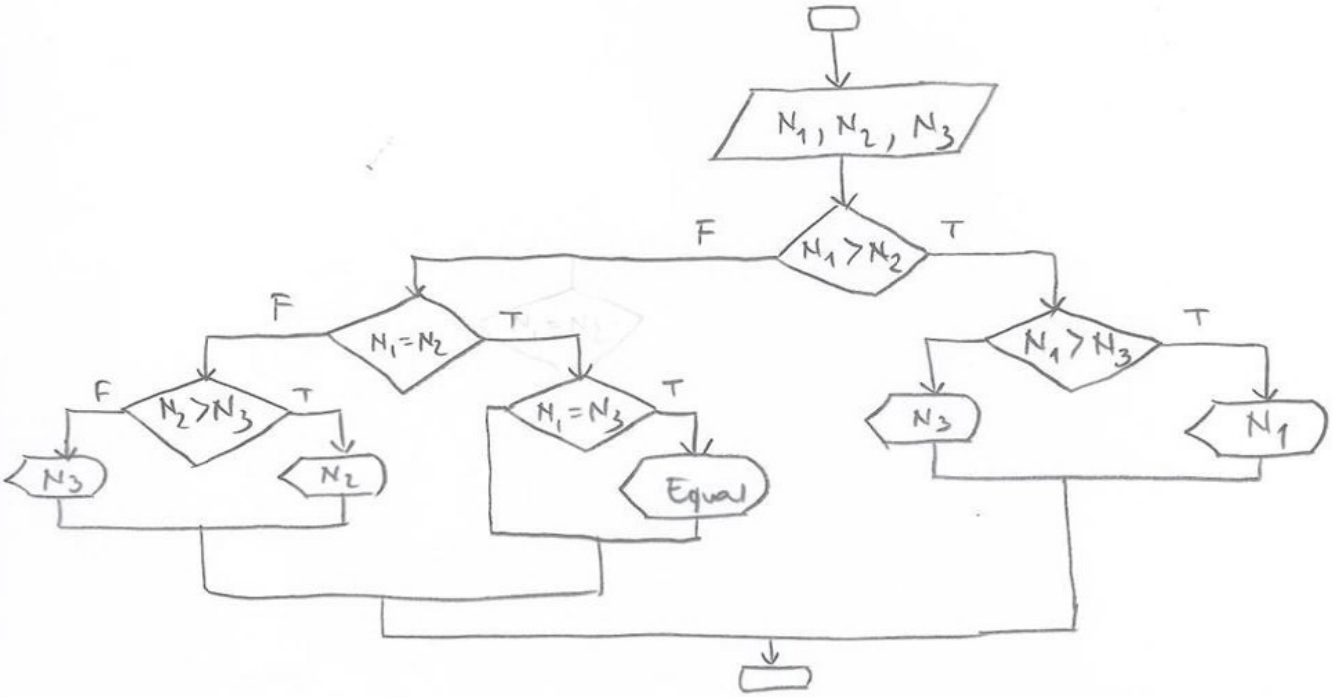
② Read three numbers and print the largest.

```
Read N1, N2, N3
if (N1 > N2)
    then ( if ( N1 > N3 )
               then ( Print N1 )
               else ( Print N3 )
           )
    else ( if ( N2 > N3 )
               then ( Print N2 )
               else ( Print N3 )
           )
```



③ Read three numbers and print the largest. If all of them are equal inform the user.

```
Read N1, N2, N3
if (N1 > N2)
   then ( if (N1 > N3)
            then (Print N1)
            else (Print N3)
        )
   else ( if (N1 = N2)
            then ( if (N1 = N3)
                     then (Print "Equal")
                   )
            else ( if (N2 > N3)
                     then (Print N2)
                     else (Print N3)
                   )
        )
```



④ Read five numbers and find the maximum.

```
Read N1, N2, N3, N4, N5
if (N1 > N2)
    then ( if ( N1 > N3 )
                then ( if (N1 > N4)
                            then ( if (N1 > N5)
                                        then ( Print N1 )
                                        else ( Print N5 )
                                   )
                            else ( if ( N4 > N5 )
                                        then ( Print N4 )
                                        else ( Print N5 )
                                   )
                       )
                else ( if ( N2 > N4 )
                            then ( if ( N2 > N5 )
                                        then ( Print N2 )
                                        else ( Print N5 )
                                   )
                            else ( if ( N4 > N5 )
                                        then ( Print N4 )
                                        else ( Print N5 )
                                   )
                       )
         )
    else ( if ( N2 > N3 )
                then ( if ( N2 > N4 )
                            then ( if ( N2 > N5 )
                                        then ( Print N2 )
                                        else ( Print N5 )
                                   )
                            else ( if ( N4 > N5 )
                                        then ( Print N4 )
                                        else ( Print N5 )
                                   )
                       )
                else ( if ( N3 > N4 )
                            then ...
```

④ cont.

```
Read N1, N2, N3, N4, N5
Max ← N1
if ( Max < N2 )
    then ( Max ← N2 )
if ( Max < N3 )
    then ( Max ← N3 )
if ( Max < N4 )
    then ( Max ← N4 )
if ( Max < N5 )
    then ( Max ← N5 )
Print Max
```

⑤ Read five numbers and find the minimum and the maximum

```
Read N1, N2, N3, N4, N5
Max ← N1, Min ← N1
if ( Max < N2 )
    then ( Max ← N2 )
    else ( if ( Min > N2 )
                then ( Min ← N2 )
         )
)
if ( Max < N3 )
    then ( Max ← N3 )
    else ( if ( Min > N3 )
                then ( Min ← N3 )
         )
)
( ... same statements for N4 & N5 ... )
Print Max, Min
```

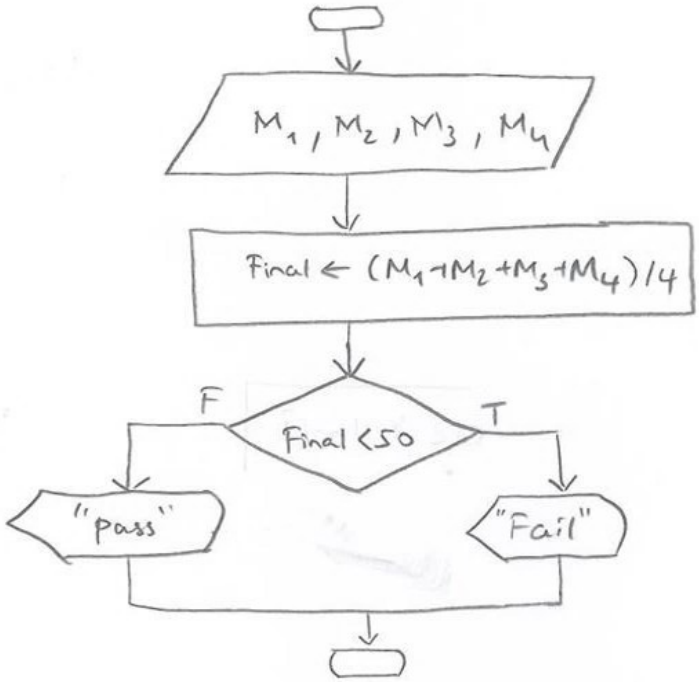
★

Instead of ★

```
if ( Max ≤ N2 )
    then ( Max ← N2 )
    else ( Min ← N2 )
)
```

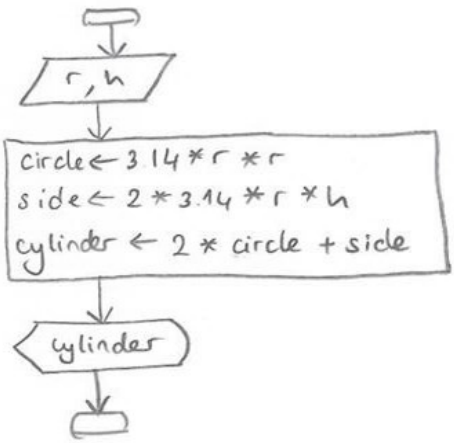
(6) Read 4 marks for a course, find the final grade (average of four) and indicate whether it is passing of failing (<50).

```
Read M1, M2, M3, M4
Final ← (M1 + M2 + M3 + M4) / 4
if ( Final < 50 )
    then ( Print "Fail" )
    else ( Print "Pass" )
)
```



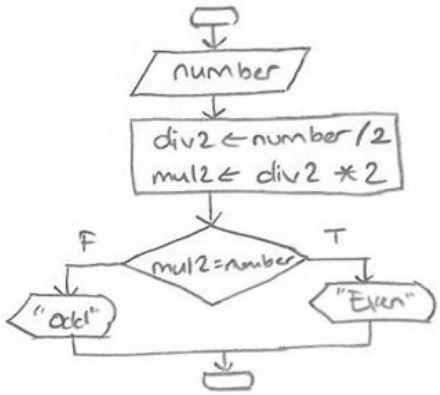
(7) Read the radius and height of a cylinder and find its area.

```
Read r, h
Area_Circle ← 3.14 * r * r
Area_Side ← 2 * 3.14 * r * h
Area_Cylinder ← 2 * Area_Circle
                + Area_Side
Print Area_Cylinder
```



(8) Read a number and determine whether it is even or not.

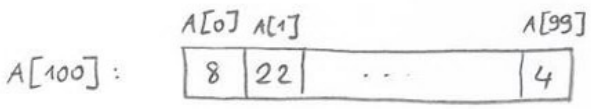
```
Read number
div2 ← number / 2
mul2 ← div2 * 2
if ( mul2 = number )
      then ( Print "Even" )
      else ( Print "Odd" )
   )
```



(9) Read two numbers and determine whether the first one is a multiple of the second one or not.

```
Read n1, n2
a ← n1/n2
b ← a * n2
if (b = n1)
     then (Print "Yes, multiple")
     else (Print "No, not multiple")
```

(10) Find the maximum of 100 numbers which are in an array A[0..99]




```
max ← A[0]
i ← 1
while (i < 100) do
    ( if (A[i] > max) then
          (max ← A[i])
      i ← i + 1
    )
Print max
```