

$R = A + 3 < B * 1$ or C and D

$A = 1$ $B = 5$ $C = -1$ $D = \text{True}$

$4 < 5$ or -1 and true

True or -1 and true

True or false

True .

$d = a + b * c$, write an algorithm for evaluating the given expression.

1. start
2. Input the values of a, b, c .
3. find the product of b and c
4. Store the product in a variable temp .
5. find the sum of a and temp .
6. Store the result in d .
7. Print the value of d .
8. stop.

Pseudocode: -

1. start
2. Read (a, b, c)
3. $d = a + b * c$
4. Print (d)
5. stop

An algorithm describes a systematic way of solving a problem. It is a step by step procedure that produces an output when given the necessary inputs. An algorithm uses pure english phrases or sentences to describe the solution to a problem.

A pseudocode is a high level representation of an algorithm that uses a mixture of natural language and programming language. It is more structured than an algorithm input. It uses mathematical expressions with english phrases to capture the essence of a solution. Pseudocode is not a true program and thus independent of any programming language. It is not executable.

Importance of Pseudocode

- i) Ease of understanding
- ii) Focus on logic
- iii) Consistent
- iv) More legible
- v) Easy translation to a programme.
- vi) Identification of flaws.

Constructs of a Pseudocode

It should follow the structured programming approach. structured coding aims to improve the readability of Pseudocode by ensuring that

the execution sequence follows the orders in which the code is written. Such a code is said to have a linear flow of control. Sequencing, selection and repetition (loop) are three programming constructs.

In the sequence structure, all instructions in the pseudocode are executed once without skipping any. On the other hand, with selection and loop structures it is possible to execute certain instructions repeatedly, or even skip some.

Sequence

The instructions of the algorithm are executed in the order listed. It is the logical equivalent of a straight line.

Common Action keywords

Input - INPUT, READ, OBTAIN, GET

Output - PRINT, OUTPUT, DISPLAY, SHOW

Compute - COMPUTE, DETERMINE, CALCULATE

initialisation - SET, INIT

Add one - IMPLEMENT, BUMP

Selection

A selection structure consists of a test condition together with one or more blocks of statements. The result of the test determines

which of these blocks is executed.

1. If structure

(a) If

```
if (condition)
    instructions
end if
```

Eg:- if ($x > 0$)
Print(x , "is positive")
end if

(b) If else

```
if (condition)
    true instructions 1
else
    false instructions 2
end if
```

Eg:- if ($x > 0$)
Print(x , "is +ve")
else
Print(x , "is -ve")
end if

(c) If else if else

```
if (condition 1)
    True instruction 1
else if (condition 2)
    True instruction 2
else
    False instruction
end if
```

Eg:- if ($x > y$)
Print(x , "is greater than y")
else if ($x < y$)
Print(x , "is less than y")
else
Print("x and y are same")
end if

2. Case structure

(case of (expression))

Case 1 value 1

BLOCK 1

Case 2 value 2

BLOCK 2

Eg:- Case of (dir)

```
case 'N'
    Print("North")
case 'S'
    Print("South")
case 'E'
    Print("East")
break
```

default

DEFAULT BLOCK

endcase

case "W"

Print("West")

break

default

Print("Invalid direction")

endcase

Repetition or loop

When a certain block of instructions is to be repeatedly executed, we use the repetition of loop construct. Each execution of the block is called an iteration or a pass. If the number of iteration is known in advance it is called definite iteration. Otherwise, it is called indefinite or conditional iterate.

1. while loop

```
while [Condition]
```

```
True instructn
```

```
end while
```

2. Repeat until loop

```
False instruction
```

```
until [condition]
```

3. For loop

3.i) for var = begin to end

```
loop instructn
```

```
end for
```

3.2) for var = begin down to end
loop instructn
end for

3.3) for var = begin to end by step
loop instructn
end for

3.4) for var = begin down to end by step
loop instructn
end for

3.1 for var = 1 to 10 1, 2, 3, ..., 10

3.2 for var = 10 downto 1 10, 9, 8, ..., 1

3.3 for var = 2 to 20 by 2 2, 4, 6, 8, ..., 20

3.4 for var = 20 down to 2 by 2 20, 18, 16, ..., 2

Symbol

Description



Flattened ellipse indicates the start and end of a module.



Rectangle is used to show arithmetic calculations.



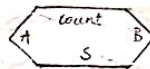
Parallelogram denotes an input/output operation.



Diamond indicates a decision box with a condition to test. It has two exits. One exit leads to a block specifying the actions to be taken when the tested condition is TRUE and the other exit leads to a second block specifying the action for FALSE case.



Rectangle with vertical side-lines denotes a module. A module is a collection of statements written to achieve a task. It is known by the name function in the programming domain.



Hexagon denotes a for loop. The symbol shown here is the representation of the loop: for count = A to B by S.



Flowlines are indicated by arrows to show the direction of data flow. Each flowline connects two

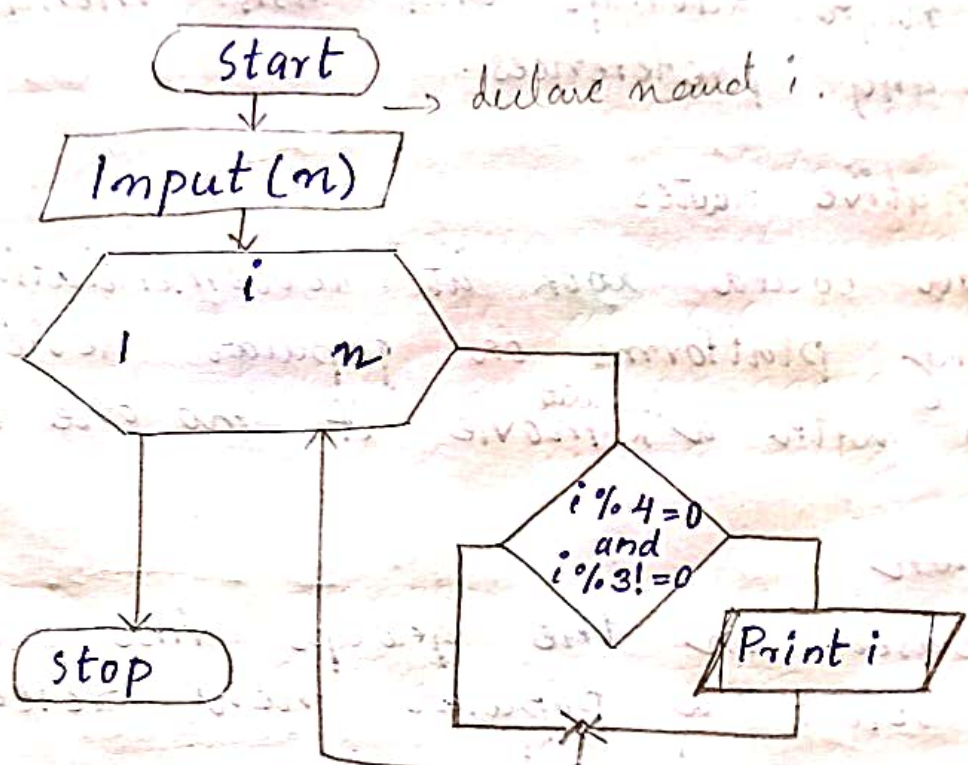


This indicates an on-page connector. This is used when one part of a long flowchart is drawn on one column of a page and the other part in the other column of the same page.



This indicates an off-page connector. This is used when the flowchart is very long and spans multiple pages.

Q. Draw a flowchart to print the numbers that are divisible by 4 but not of 3, in a list of positive numbers.



$\% \rightarrow$ modulus
 \downarrow
remainder
 $= 0$

- 1.) To determine the larger of two numbers, write down the algorithm and flowchart.
- 2.) To find the simple interest draw the flowchart and write down the algorithm.

1.) Algorithm: -

1) Start

2) Input the values of a and b.

3) If a is greater than b, then set ^{larger} as 'a'.

4.) Else, set ^{larger} as 'b'. 5) end if 6) Print (larger)

7) Stop

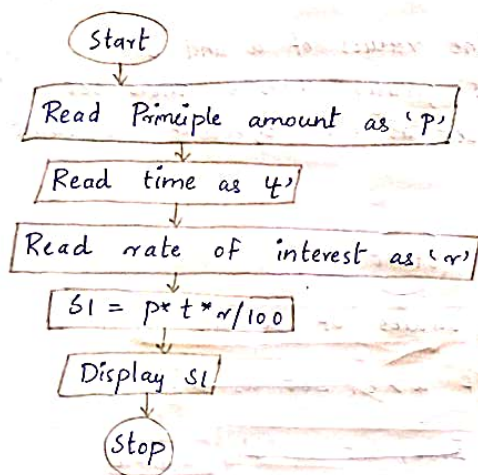
Start

Read principal amount as 'p', time as 't'
and rate of interest as 'r'.

$$SI = p \times t \times r / 100$$

Display SI

Stop

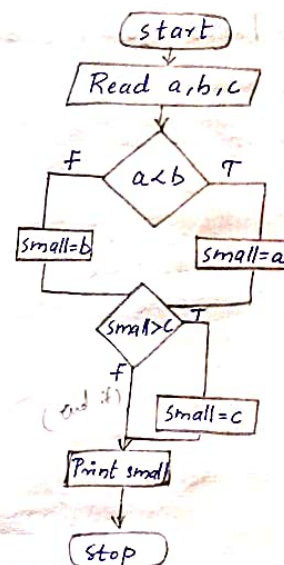


3. To determine the small of three numbers.

Algorithm:-

- 1.) start
- 2.) Read a, b, c.
- 3.) If ~~a > b~~ ($a < b$)
- 4.) ~~smaller = a~~
- 5.) else if $b < a$
- 6.) smaller = b
- 7.) ~~else < end if~~
- 8.) IF ($smaller > c$)
- 9.) smaller = c
- 10.) end if
- 11.) Print smaller
- 12.) Stop.

Flowchart:-



4. To determine the entry ticket fare in a zoo based on age as follows:

Age fare

<10 7

>=10 and <60 10

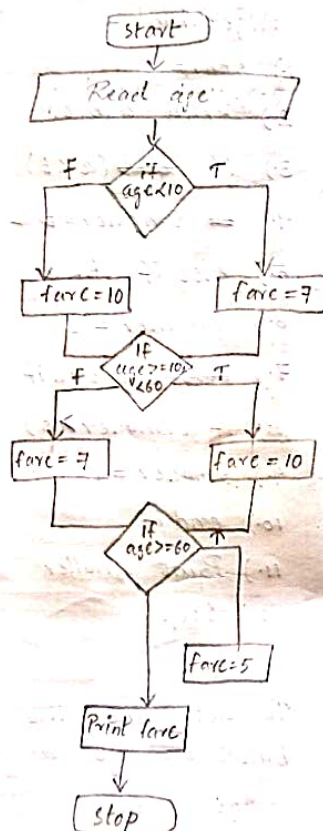
>=60 5

Write Algorithm and flowchart.

Algorithm:-

- 1) start
- 2) Input age
- 3) If age < 10
- 4) fare = 7
- 5) else if age >= 10 and < 60
- 6) fare = 10
- 7) endif
- 8) If age >= 60
- 9) fare = 5
- 10) end if
- 11) Print fare
- 12) Stop.

Flowchart:-



To print the colour based on the code value as follows.

grade	Message
R	Red
G	Green
B	Blue
Any other value	Wrong code

Flowchart:-

1. Start
2. Read code

3. Case of (code)

4. Case 'R':

5. print ('red')

6. Break

7. Case 'G':

8. print ('green')

9. Break

10. Case 'B':

11. print ('blue')

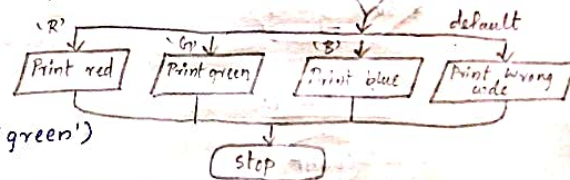
12. Break

13. default:

14. print ('wrong code')

15. end case

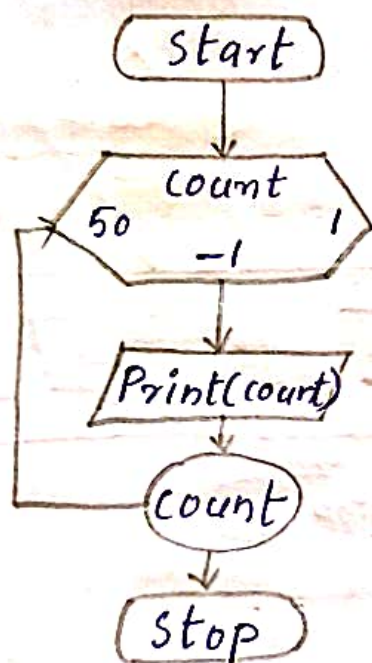
16. Stop.



Q. Write down the algorithm and flowchart to print the numbers from 1-50 in descending order.

- 1.) start
- 2.) for count = 50 down to 1
- 3.) Print (count)
- 4.) end for
- 5.) stop

Flowchart :-



2. Find the factorial of a number.

- 1.) start
- 2.) Read (n)
- 3.) fact = 1

$$5! = 1 \times 2 \times 3 \times 4 \times 5$$

4. for ~~variable~~ var = n down to 1.

5. fact = fact * var

6. end for

7. Print fact

8. Stop.