

Program:

- An algorithm expressed using a precise notation is called a program.
- The precise notation is called a programming language.
- The general terminology SOFTWARE is used to describe all programs.
- The terminology HARDWARE is used to describe the electronic circuitry devices, memory etc. which are required to execute a software.
- H/W & S/W are also used to represent many other systems.
- TV is h/w & entertainment pgm is S/W.
- H/W is a one time expense whereas S/W is a continuous expense.
- S/W production is time consuming and expensive whereas H/W can be built to specifications.

System software.

- System software is software designed to provide a platform for other software.
- A software used for system working or make it alive.
- OS, compiler, assembler and interpreter are examples of system software.

Package software.

- A software package is simply multiple applications or code modules that work together to meet various goals & objectives.

→ MS
→ LA-SH

→ Program
software
eg: &
→ Libre
→ intern
→ Skype.

→ Three
i) Opera
ii) Program
iii) utility

→ MS office is a packaged software.

→ LA-SUITE includes Gmail, Docs, Drive, calendar, Meet.

Application software.

→ Programs written for specific applications are called application software.

→ e.g. railway ticket reservation system.

→ LibreOffice writer, Libre office calc etc are applications.

→ internet browsers like Firefox, Safari and Chrome.

→ Skype.

System software.

→ Three types of system software.

i) Operating system

ii) Programming languages and their translators.

iii) Utility Programs.

→ Algorithm to print the day of week.

if ($d = 1$)

Print "Sunday" and so on

1. Start

2. Input d

3. If ($d = 1$)

Print "Sunday"

else if ($d = 2$)

Print "Monday"

else if ($d = 3$)

Print "Tuesday"

else if ($d = 4$)

Print "Wednesday"

else if ($d = 5$)

Print "Thursday"

else if ($d = 6$)

Print "Friday"

else if ($d = 7$)

Print "Saturday"

else {

Print "Invalid code"

4 stop

Algorithm for Rainbow colours.

1. Start

2. Input colour code cc

3. If ($cc = 1$)

Print "Violet"

else if ($cc = 2$)

Algorithm

An

consum

4.5

→

1. Start

2. Input

3. If (

ma

els

```

Print ("indigo")
else if (cc==3)
    Print ("blue")
else if (cc==4)
    Print ("green")
else if (cc==5)
    Print ("yellow")
else if (cc==6)
    Print ("orange")
else if (cc==7)
    Print ("red")
else
    Print ("invalid")

```

4. Step

- Algorithm to calculate the electricity bill
- An electric power distribution company charges, it's domestic consumers as follows.

| Units | Rate. |
|-----------|------------------------------------|
| 0 - 200 | Rs 0.50 Per unit |
| 201 - 400 | 100 + 0.65 * excess units consumed |
| 401 - 600 | 230 + 0.80 * excess units consumed |
| > 600 | 390 + 1.0 * excess units consumed. |

- 1. Start
- 2. Input unit consumed uc
- 3. If (uc < 201)
 - rate = 0.50 * uc
 - else if (uc > 200) and (uc < 401)

$$\text{rate} = 100 + 0.65 * (\text{uc} - 200)$$

else if ($\text{uc} > 400$) and ($\text{uc} < 601$)

$$\text{rate} = 230 + 0.80 * (\text{uc} - 400)$$

else

$$\text{rate} = 390 + 1.0 * (\text{uc} - 600)$$

4. Print ("Rate = ", rate)

5. Stop

For loop

In computer science a for loop is a control statement for specifying iteration which allows code to be executed repeatedly

Syntax is:

for $i = 1$ to n step size.

{
body of the loop
}

next statement.

for eg:- For $i = 1$ to 5 step 1

{
print (i)
}

O/p:- 1 2 3 4

for $i = 5$ to 1 step -1

{
print (i)
}

O/p 5 4 3 2

for $i = 1$ to 10 step 2

{
print (i)
}

O/p:- 1 3 5 7 9

first n natural numbers

1. start
2. input value of n
3. for $i = 1$ to $n+1$ step 1
 - {
 - print (i)
 - }
4. stop

first n natural numbers and their squares

1. start
2. input n
3. for $i = 1$ to $n+1$ step 1
 - {
 - print (i, i^2)
 - }
4. stop

first n natural numbers and their cubes

1. start
2. input n
3. $s = 0$
4. for $i = 1$ to $n+1$ step 1
 - {
 - $s = s + i^3$
 - }
5. print s
6. stop

first n natural numbers and their squares

1. start
2. input n
3. for $i = 1$ to $n+1$ step 1
 Print ($i, i * i$)
4. stop

first N odd numbers

1. start
2. input N
3. for $i = 1$ to $n+1$ step 2
 Print (i)
4. Print "Over"
5. stop

first n even numbers

1. start
2. input n
3. for $i = 2$ to $n+1$ step 2
 Print (i)
4. Print "done"
5. stop

sum of any n numbers

1. start
2. input n
3. sum = 0
4. for $i = 1$ to n
 {
 input = "any number", m
 sum = sum + m
 }

5. Print sum

6. stop

9/2/21
11.

Algorithm to find the factorial of a given number:

$$1! = 1$$

$$2! = 2 * 1 = 2$$

$$3! = 3 * 2 * 1 = 6$$

$$4! = 4 * 3 * 2 * 1 = 24$$

$$5! = 5 * 4 * 3 * 2 * 1$$

$$5! = 5 * 4! = 5 * 4 * 3 * 2 * 1 = 120$$

1. Start

2. fact = 1

3. Input "factorial of which number", n

3. for i = 1 to n+1

$$\text{fact} = \text{fact} * i$$

4. Print fact

5. stop

$$n=4 \quad \text{fact}=1 \quad i=1 \quad \text{fact}=1$$

$$i=2 \quad \text{fact}=2$$

$$i=3 \quad \text{fact}=6$$

$$i=4 \quad \text{fact}=24$$

12. Algorithm to print the multiplication table.

$$1 * 2 = 2$$

$$2 * 2 = 4$$

$$3 * 2 = 6$$

$$4 * 2 = 8$$

$$5 * 2 = 10$$

$$6 * 2 = 12$$

$$7 * 2 = 14$$

$$8 * 2 = 16$$

$$9 * 2 = 18$$

$$10 * 2 = 20$$

1. Start
2. input "Number of rows", n
3. input "multiplication table of which number", m
4. for $i = 1$ to $n+1$
 Print (i , '*', m , '=', $i * m$)
5. stop

| n | i | m | $i * m$ |
|-----|-----|-----|---------|
| 5 | 1 | * 4 | = = 4 |
| | 2 | * 4 | = = 8 |
| | 3 | * 4 | = = 12 |
| | 4 | * 4 | = = 16 |
| | 5 | * 4 | = = 20 |

13 Algorithm to check whether a given number is prime or not.

1. Start
2. input "Enter a number", n
3. flag = 0
3. for $i = 2$ to n

 if ($n \% i == 0$)

 {
 flag = 1
 break
 }

4. if (flag == 0):
 Print ("Prime")
 else
 Print ("not prime")

5. Stop.

while loop.

Another iterative statement similar to for loop is while loop
syntax is

```
while(expression)
```

```
{
```

```
    Body of loop
```

```
}
```

next stmt

Example

For $i = 1$ to 10

Print(i)

$i = 1$

while ($i < 10$)

```
{
```

```
    Print( $i$ )
```

```
     $i = i + 1$ 
```

```
}
```

```
Print("over")
```