Date
Page

Programming introduction -

In computer system, when a user feed certain data for processing then it is up to the system that it respond on it or not. When there is no any action being expressed on it then it converted itself into garbage value. When certain processing happens as per the used software then it turns into an instruction. Every instruction having two components i.e operand and operation. The operand represents address where operation takes place whereas operation decides whether the instruction performs arithmetical, logical or shift operation.

The combination of specified instruction is known as program. The person who writes a program is known as programmer and the way of writing of the program is known as programming. There are two different classes of program i.e-

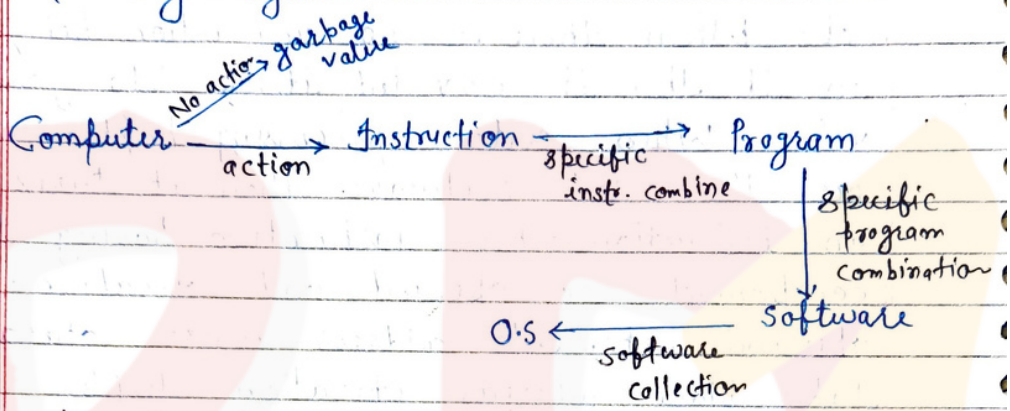
i) **Application program -**

The program whose interface depends on another environment and a user uses such particular environment to develop their own application is known as application programe.

ii) **Language program -**

It is one of the program whose working nature is independent and mostly used to develop the specified type of program using the constructs of language.

The combination of specified programs is known as software. The combination of software is known as operating system.



* Characteristics of good programming -
There are several characteristics of a good programming i,e -

1) Hardware independent -
The program written in any application or language if do not depends on the internal architecture of the system that means the program can execute on any type of the system will treated as good program.

2) Modularity -
In the programming enviornment when the large program being written, then the chance of no of errors might be increased. If the entire programs becomes divided into a no of small sections then each one is known as module and the way through which it creates is known as modularity.

3) Structured -

The way of writing of the program is previously decided as well as their program construct to and the user follows such particular pattern to write the program is known as structured program.

4) General purpose -

When the program supports different nature of software development without making any changes in the way of writing of program construct i,e known as general purpose programming.

5) Easy to ~~debug~~ write -

If the program syntax is user friendly in nature then it is easy to learn and write to develop a new software

6) Easy to debug -

When the programming provides error message along with the line number in which error generates then it is easy to debug.

7) Modifiability -

Once the program being develop and after a period of time, if the user need to make certain changes and if the changes accepted easily then it is treated as good programming.

8. Documentation -

When the program having help file or other methods to get the help about the program constructs then it is known as documentation that behaves as characteristics of a good program.

Date
Page

Flowchart -

In computer system, there are various operations on different nature & invisonment becomes happens either internaly or exterbaly. When such particular representation being diagramatically represent then it is known as flow chart. The flowchart only describes about the flow of data not their working component.

"The flowchart is one of the diagramatical representation that mentains logical flow of processing."

Catagorys of flowchart -

There are two catagories of flow chart i,e -

i) Based on size -

As per the length of the flow chart, it is of two different types -

a) Macro flowchart -

The flowchart whose size is small in nature and finishesh itself on a single page then it is known as macro flowchart.

b) Micro flowchart -

When the length of the flow chart goes more than one page long then it is known as micro flowchart

2) Based on representation -

As per the representation, the flowchart is of three different types -

a) Program flow chart -

The flowchart that describes, the flow of instructions of the program is known as program flow chart. The program flowchart is also known as logical diagram.

b) Operational flowchart -

Whenever certain processing happens inside or outside the system that having set of operations to ~~the~~ achieve the specified task and objectivity then such particular diagrametical representation is known as operational ~~program~~ flowchart.

3) System flowchart -

The flowchart that describes the working of computer system or organisation then such particular type of flowchart is known as system flowchart.

* Rules & guidlines to creat the flowchart -

Once any types of the flowchart when created then there are certain rules to be followed i.e -

i) The flowchart must having the start and stop specification.

ii) The flowchart based flowlines never intersect with one - other.

iii) The symbols of the flowchart must be properly specified in the diagram.

Date
Page

iv) The temporary and permanent storage being seprately located in flowchart.

v) The flowchart never be divided.

vi) The flowchart never be created for the multiple process in the single one.

* Symbols of flowchart

There are various flowcharts symbolic structure used to develop it i,e -

i) Ellipse (⬭) -
Whenever start and stop activity being represented, then it is used.

ii) Parallelogram (▱) -
Whenever the initialization ~~input & output activity need~~ of variable input & output activity need to repres-ent it flowchart then it is used.

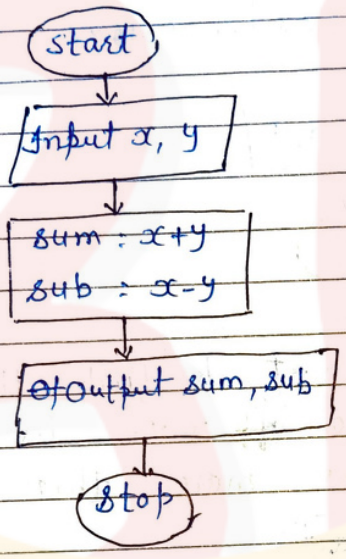
iii) Rectangle (▭) -
All different types of processing mainly aisthmetical & logical when performs then to represent it rectangle is used.

iv) Diamond (◇) - It is used to verify the conditions & returns ~~two states~~ to decides capacity i,e 'yes' or 'no'.

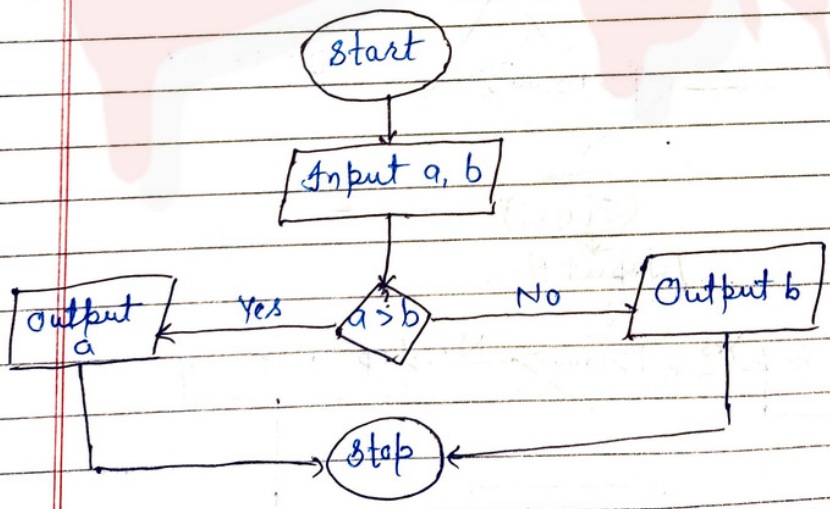
v)

v) Flow line ($\rightarrow$, $\downarrow\uparrow$) -
To specify the flow of the data from one part of the flow chart to another.

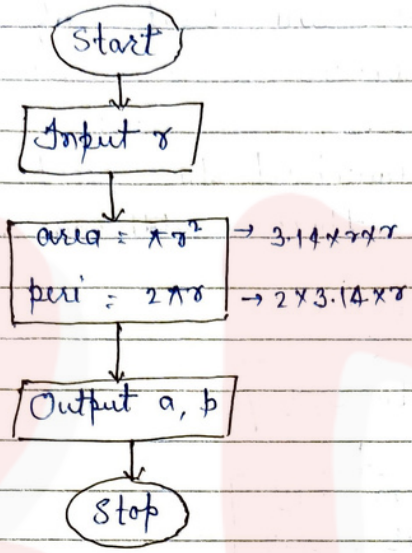
Q Draw a flowchart to accept two numbers and show their addition and subtraction



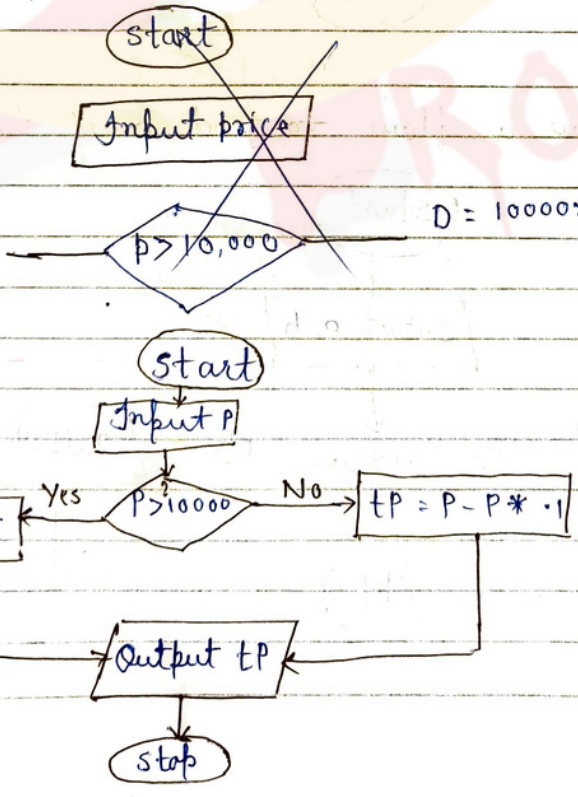
Q Draw a flow to find max" b/w two numbers.



Q Draw a flowchart to find area and perimeter after accepting the radius of circle



Q Draw a flowchart to find the net price by giving 20 ~~off~~ or 10 % discount when inputted price is more than 10000 or less

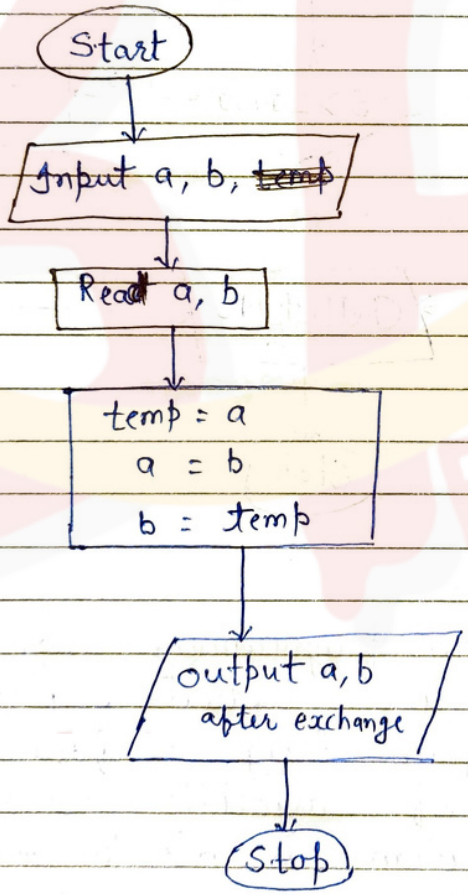


Date
Page

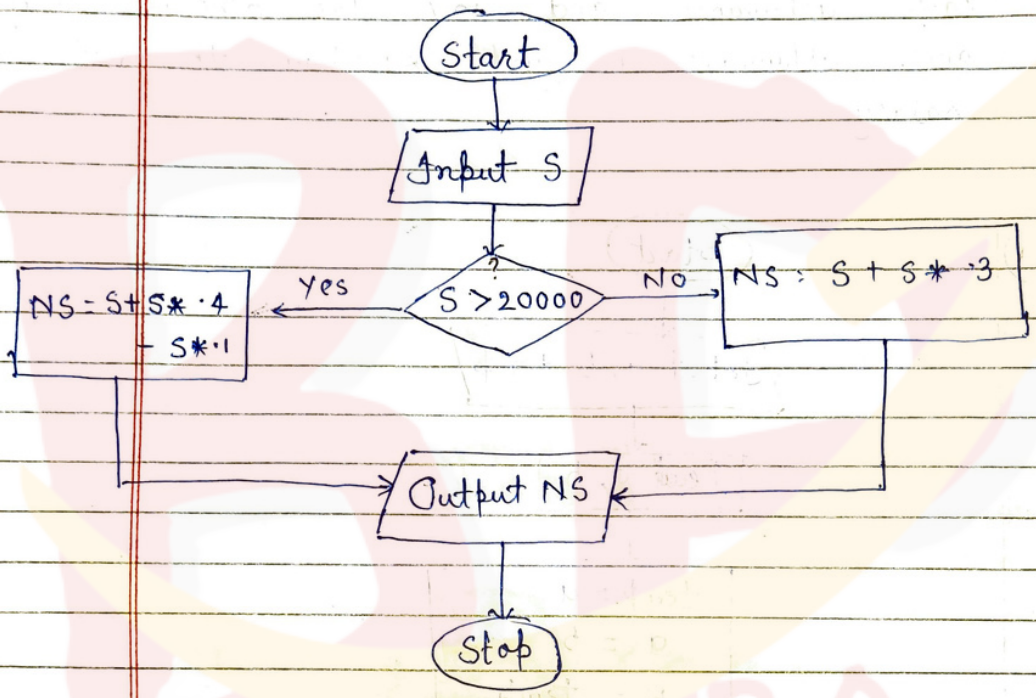
Q1. Draw a flowchart that accept two numbers interchange their value.

Q2. Draw a flowchart that accept salary of an emp if salary is more than 20000 then give 40% allowance and 10% tax otherwise only 30% allowance is given. so Show the net salary.

1)



Q Draw a flowchart that accept ~~two~~ salary of an emp. if salary is more than 20000 then give 40% allowance and 10% tax otherwise only 30% allowance is given. Show the nets salary.



* Algorithm -

In programming, logic and technique whenever the textual representation of the program being perform then the concept of algorithm becomes used. The algorithm is also known as pseudo code The algorithm is the english like statement that discribe the program structured in literal format. Once an algorithm need to be created then there are certain keypoints to be remembered ie-

i) When an algorith of the program is created then it must be complete in nature.

ii) There are a no. of steps to be followed to describe the program algorithm and next ϕ steps of program algorithm the dependency is almost on previous step.

iii) The algorith never be divided ~~in to~~ into components.

iv) Each steps of an algorithm having a specified number.

v) Each step s contains the unique processing description.

vi) The polar is used to separate the process no. and processing syntax.

vi) The output keyword is used to generate the output if form in whereas read statement is used to input the data.

vii) To put the condition on the variable, 'if' statement is used.

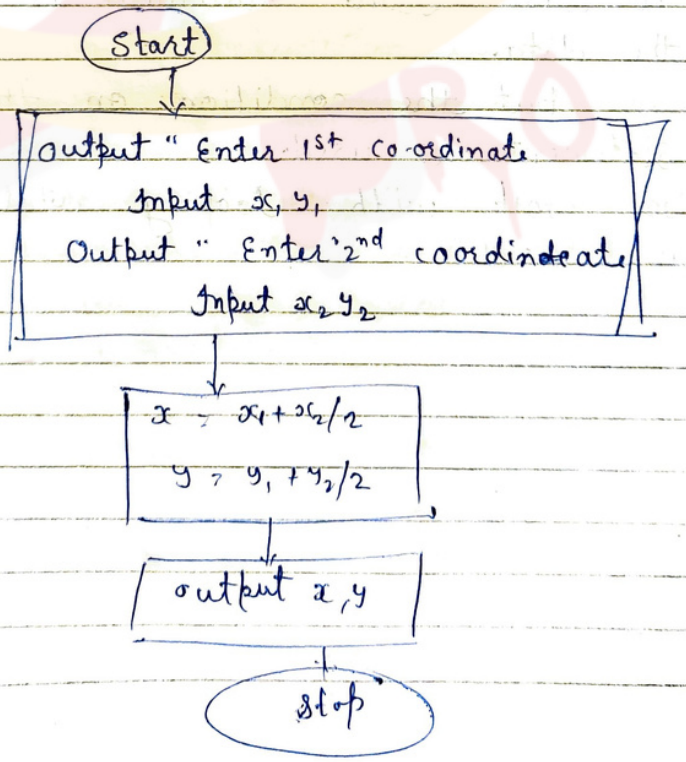
viii) To work with looping while statement is used.

Q Write an algorith to find out the mid point of a line

step 1 : start
step 2 : Output " Enter 1st coordinate of line"
Read x_1, y_1
Output " Enter 2nd coordinate of line
Read x_2, y_2
step 3 : [calculate mid point]
$x = x_1 + x_2 / 2$
$y = y_1 + y_2 / 2$

step 4 : [Mid point value]
out x, y

step 5 : stop



$\frac{C}{5} = F - \frac{32}{9}$

Date
Page

Q Creat a algorithm that accept to convert °C to F

step-1 : start
step-2 : Input temperature in °C
step-3 : [calculate temperature in F]

$F = \frac{C}{5} - \frac{32}{9} : \left(\frac{9}{5} \times C\right) + 32$

Step-4 : [temperature in F]
output temperature in F

Step 5 : - stop

Ans
step-1 : start
step-2 : output "Enter the temp^r in °C"
Read temp^r in °C.
step-3 : [calculate temp^r in F]
$F = \left(\frac{9}{5} \times C\right) + 32$.

step 4 : [temp^r in F]
output temp^r in F.

step 5: stop.

* Basic -(BASIC) -

The basic is a high level language thats supports general purpose programming environment-in which different nature of programs becomes written by the user.

The BASIC stands for beginer all purpose symbolic instruction code.

This language was duclop by two scientist named John Kemeny and Thomas kurtz in 1964.

* Error -

In programming enviornment, the user writes the program syntax and send for translation. The ~~basic~~ BASIC having interpretter as translator. All the program syntaxes written by the users if not ~~rego~~ recognise by the translator then it is known as error in the program. When the programme goes for processing then a register known as flag is responsible to display a error message after it storage.

There are four different classes of an error i.e -

i) Syntax grammatical error -

The errors that occcers due to mistake in writing of the program syntax, is known as syntax error. Due to

↗ Forward slash

5/2 : 2.5 5mod2 : 1 (Remainder)

5\2 : 2 5^2 : 25 Same

Backward slash 5**2 = 25

structure behaviour the grammar of language becomes fixed in the nature.

ii) Logic error -

The error that occurs due to the misplacement of program construct. There is no any error message being from once it happens.

iii) Linker -

The program syntax, if do not properly interact with the hardware files language files then such particular error being generated.

iv) Fatal error -

The program syntax written by the user go for the translation inside the primary memory (RAM) and the primary memory do not have the sufficient space then such particular error has been generated.

* Operators -

Operators are the symbolic structure and words through ~~insid~~ which an expression being written by the user in program.

Operators when combined with operand then it results either numeric value or boolean value.

* Types of operators -

i) Arithmetical operation : +, -, *, /, \, mod, ^, **

ii) Relational operator : >, <, >=, <=, =, <>

iii) Logical operator: AND, OR, NOT, eqv, NOR

iv) Pattern matching: IS, Like

v) Concatenation operator: &, +

* Variable -

The variable is a memory location in which value becomes stored as per data type through which different types of calculation being performed. The address of the variable represented in hexadecimal format where as value is in decimal format. To creat the variable, the BASIC language provides certain naming convention i.e -

i) The variable must be alphanumeric in nature.

ii) The space or symbols are not allowed instead of underscore.

iii) The keywords are not allowed to put the & name.

iv) The length of the variable name goes up to 40 characters long.

v) The name is not case sensitive.

* Stages of variable -

In BASIC programming, there are two different ~~st~~ stages of the variable -

i) Declaration of variable -

When the name and type of variable being decided only then it is known

as declaration of variable. Eg - Let CH$ = "BCAI"

ii) Initilization of variable -

Whenever the name, type and value being assign to the variable then it is known as initilization.

Let Ch$ = "BCAI"

* Types of variable -

There are three different types of variable i,e -

i) Global variable -

The variable that is created at the top of program is known as global variable. The space scope of the global variable is in entire program whereas life ends when the program terminates.

Local

ii) Formal variable -

The variable i,e created at the time of calculation of the program or within the body of program construct is known as local variable.

iii) Formal variable -

The variable i,e created within the parathesis of function is known as formal variable. The life of formal variable is only until function execution is complete where as the scope is within the body of function.

Date
Page

* Datatype -

In programming environment, whenever a user need to stored the value within the variable then type & size of the variable becomes determined by the datatype. The specification of data type is available in the interpreteter of particular language. The data types becomes classified into four different catagory -

i) Character -

The datatype that stores a single character value for processing is known as character datatype. The storage capacity of this data type is of 1 byte.

ii) Numeric -

The datatype that stores numeric value in the variable for processing. It is further divided into two types -

a) Decimal numeric data type -

When the whole no. value stored in the variable then it is used. A single value ranges b/w 32768 to 32767. The storage capacity is of two bytes.

b) Real numeric data type -

When the fraction value becomes stored in the variable then it is used. It is to of two different types -

- Single precision real data type -

Whenever 10 digits of precision & 6 digit of scale value stored within the variable then it is used. The storage capacity is of four bytes.

- Double precision real data type -
When 10 digits of precision & 14 digits of scale value becomes stored in the variable then it is used. The eight bytes of storage capacity belongs to it.

iii) String datatype -
When more than one character are stored in the variable then it is known as string datatype. The variable succeded by '$' sign belongs to string type of variable

iv) Boolean datatype -
When the variable stores true & flase value then it is used. The storage capacity is of 1 byte.

* Output statement -
To make the output at the output screen, the BASIC provides a statement known as "print". There are certain operation can be supported by print statement i,e -

i) To show the message and on output screen the print statement is used.
~~iii) To show the~~ eg - Print "Welcome to BASIC"

ii) To show the message and value of variable it is used.

```
let x = 10
print x
```

iii) We can combine message along with the variable value using print statement.

```
print "sum", z
```

iv) To perform the arithmetical & logical operation and get the resultant.

```
print x+y
print x>y
```

* Input statement -

In the BASIC programming environment when a user need to make input within the variable then the input statement is to be used. There are two different patterns a/c to which the input operation can be performed.

i) We can make the input within the variable without assigning the message.

→ Print "Enter the age"
input age

→ Print "Enter two numbers"
input x, y

ii) The input statement having the facility to make a message first then get the value within the variable. If the ~~sp~~ seperator is semicolon(;) then the interrogative (?) for the message whereas if comm (,) is used then no interrogative becomes placed before the message.

```
Input "Enter the age"; age
Input "Enter the age", age
```

Q WAPTA radius of circle, find out the area and perimeter?

```
Cls
Input "Enter the radius of circle"; r
    area = 3.14 * r * r
    peri = 2 * 3.14 * r
Print "Area=" area
Print "perimeter=" peri
```

Q WAPTA no. of items and price per unit give 10 % discount on their gross price, Show the netprice.

```
Cls
Input "Enter the no. of item & price"; n, p
    gp = n * p
    np = gp - gp * .1
Print "Netprice=" np
```

Q WAPTA length & breadth of rectangle show the area and perimeter.

```
Cls
Input "Enter the length and breadth"; l, b
    area = l * b
    peri = 2 * l + b
Print "Area" area
Print "Perimeter" peri
```

Q WAPTA distance in km convert into m, cm, feet & inch

```
Cls
Input "Enter the distance in km"; dis
    distance in m = dis * 1000
    distance in cm = dis * 100000
    distance in ~~feet~~ inch = dis * 100000 / 2.54
    distance in feet = dis * 100000 / 2.54
```

Or

Cls

Input "Enter the distance in km", dis

m = dis * 1000

cm = m * 100

inch = cm / 2.54

feet = inch / 12

Print "In meter:" m

Print "In cm:" cm

Print "In inch:" inch

Print "In feet:" feet

Q WAPTA salary of a person, find the net salary when D.A is 40% & tax is 10% of salary.

Cls

Input "Enter the salary:", sal

Netsalary = sal + sal * 0.4 - sal * .1

Print "Netsalary =" Netsalary

Date
Page

Conditional statement -

In BASIC programming environment, when certain condition being implemented on variable value then to verify it the concept of condition statement becomes used. The condition when verified then it may returns true or false based on the condition. There are different types of conditional statement becomes ~~so~~ available-

i) if statement -

When certain condition being applied on the variable and the expression only be executed when the condition return true then it is used.

syntax - If condition then
expression
end if

WAPTA any number add 20% of the number when its value is less than 500.

```
cls
Input "Enter any number", n
    if n < 500 then
    n = n + n * .2
    print n
    end if.
```

ii) if else statement -

In BASIC programming environment, when certain condition being applied on the variable and if the expression being executed both the condition becomes true or false. The expression below if executes

when the condition returns true otherwise expression below else executed.

syntax- if condition then
expression
else expression
end if

Q WAPTA any number, check ~~if~~ it is even or odd.

```
Cls
Input "Enter the number", n
  if n mod 2 = 0 then
     print "Even"
  else
     print "Odd"
  endif
```

Q WAPTA no. of items and their price per unit give 20% discount when gross prise is greater than 10000 otherwise 5% discount is given. Show the netprice?

```
Cls
Input "Enter the no. of items", i
Input "Enter the price of one item", p
     GP = i * P
  if GP > 10000 then
     NP = GP - GP * 0.2
     print "NP"
  else
     NP = GP - GP * 0.5
  print "NP"
  endif
```

Q WAPTA cost and selling price of an item. Show the transaction is in profit or loss and how much?

```
Cls
Input "enter the cost price", C
Input " enter the selling price", S
    If  S > C then
        print " Profit", S - C
    else
        print " loss", C - S
    endif if
```

Q WAPTA year by the user then check it is leap year or not?

```
Cls
Input " enter the year", y
    If  y mod 4 = 0 then
      Print " leap year : yes"
    else
     print " leap year : no"
    endif if
```

Date
Page

iii) if else if statement -

In BASIC programming environment whenever more than one conditions being applied then to verify it the concept of conditional statement i,e if else if becomes used. The expression written below the condition gets executed when condition returns true otherwise next condition being verify. If no- ~~of~~ any condition returns true then expression below else executed.

Syntax If condition then
expression
elseif condition then
expression
elseif condition then
expression
else
expression
end if

Q WAPTA any number, check it is +ve, -ve or zero?

```
Cls
Input "enter any no", n
  if n>0 then
   print "+ve"
  elseif n<0 then
    print "-ve"
  elseif
     print @ "zero"
  endif if
```

Q WAPTA marks by the user, show their division.

marks >= 540 then 1st
480 - 539 - 2nd
420 - 479 - 3rd
< 420 - fail

```
cls
Input "enter the marks", m
  if  m>=540 then
      print "first"
  elseif m>=480 and m<=539 then
      print "second"
  elseif m>=420 and m<=479 then
      print "third"
  elseif
      m<420 then
      print "fail".
```

Q WAPTA any character by the user then check it is in uppercase, lower case, digit or symbol.

```
cls  let n=0
Input "enter the char", ch$
  if  n = asc(ch$)
 If  n>=65 and n<=90 then
     ch$ = uppercase
 elseif n>=97 and n<=122 then
     ch$ = "lower case"
 elseif n>=48 and n<=57 then
     ch$ = "Digit"
 else
     ch$ = "Symbol"
 endif
```

Q. WAPTA three angle of a triangle then show it is rightangle, equilateral or isoscale.

```
cls
input "enter the angles", a, b, c
  if a = 90 or b = 90 or c = 90
     print "Rightangle triangle"
  elseif a = b = c then
     print "equilateral"
  elseif a = b or a = c or b = c then
     print "Isoscale"
  endif
```

Q. WAPTA week day no. then show, week day name.

```
cls
input "enter the week day no.", a
  if a = 1 then
    print "Sunday"
  elseif a = 2 then
    print "Monday"
    .
    .
    .
  elseif a = 7 then
    print "Saturday"
  endif.
```

Q. WAPTA calculate the amound of meter reading then show how much amount to be paid by the coustmer.

```
cls
input "enter the amount of meter reading", a
  if a >= 1 and a <= 50 then
  print "payable amount", a * .5
  elseif a >= 51 and a <= 100 then
    print "payable amount", a * .8
  elseif a >= 101 and a <= 150 then
       print "payable amount", a * 1
  elseif a > 150 then
  print "payable amount", a * 2   endif.
```

1 - 50 → 50P/U
51 - 100 → 80P/U
101 - 150 → 1R/U
>150 → 2R/U

Date
Page

* Looping statement in BASIC -

In the BASIC programming environm whenever a sigle line or multiline of expression need to be executed more than one time then the concept of looping statement becomes used. The loop statement is also know as control flow statement or ~~iter~~ iterative statement. The working ~~what we do~~ with looping statemthaving several significances -

i) It reduces the size of program.

ii) No. of variables are decreases therefore the efficiency of the execution of program increases.

There are two catagories of looping statement -

i) Based on placement of condition -

A/c to the placement of the loop is of two different types -

a) Entry level loop

The loop in which the condition is placed at the top of the loop is known as entry level loop. When the condition is placed at the bottom of looping statement then it is known as exit level loop.

ii) Based on no. of iteration -

A/c to the no. of iteration, the loop is of two different types -

a) finite loop b) Infinite loop.

The loop in which the no. of iteration becomes fixed then ~~in~~ it is

known as finite loop whereas the loop in which no. of iteration is not fixed then it is known as infinite loop.

* Types of loops -

There are different types of looping statement supported by BASIC programing language.

i) For loop
ii) While loop
iii) do...while...loop
iv) do...loop...while
v) do...until...loop
vi) do...loop...until

i) For loop :-

The For loop is one of the looping statement that supports finite with entry level looping statement. To work with for loop there are three different specification -

a) Whenever the loop counter go from lower to higher value then the For loop is written as per follows.

```
For lower bound to upper bound
    express
next
```

If this particular for loop the counter value is increases one by on automatically

b) When the countervalue increases more than one times then 'step' keyword along with increment value of counter is written with for loop.

```
For lower bound to upper bound step n
    expression
next
```

c) Whenever a user need to execute the loop from higher to lower bound then the step keyword is written along with negative value with the for loop.

```
For upper bound to lower bound step -n
        expression
    next
```

To forceful to pause the processing a statement is known as "exit for" is written within the for looping statement.

We can put the for loop upto thirteen times in which outer and inner loop being generated.

Q WAPTA any number then check it is prime number on not?

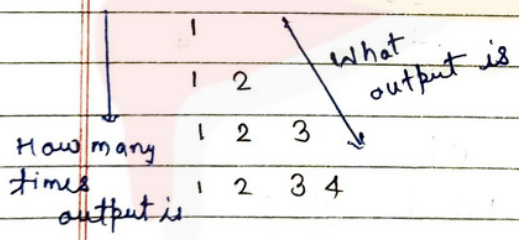
```
cls
Input " enter the no", n
    let c = 0
for i = 2 to n/2
  if n mod i = 0 then
      c = 1
    exit for
  end if
  next i
  if c = 0 then
    print "Prime"
  else
    print "Not prime"
  endif
```

Date
Page

Q WAP TA any number, show their prime factor?

```
Cls
Input "enter to number", n
    let c = 0
for i = 2 to n/2
    if n mod i = 0 then
    for j = 2 to i/2
  if  i mod j = 0 then
        c = 1
        exit for
        endif if
        next j
        endif
    if c = 0 then
        print i
    endif
        c = 0
```

* Pattern




```
Cls
for i = 1 to 4
 for j = 1 to i
   print j; " ";
   next j
   print
   next i
```

1 2 3 4
1 2 3
1 2
1

```
Cls
for i = 4 to 1 step -1
 for j = 1 to i
 print j; " ";
 next j
 print
 next i
```

Date
Page

Q
```
4 4 4 4
3 3 3
2 2
1
```

```
Cls
for i = 4 to 1 step -1
  for j = 4 to i
    print j;" ";
  next
```

```
Cls
for i = 4 to 1 step -1
  for j = 1 to i
    print i," ",
  next j
  print
next i
```

Q 1)
```
1
2 2
3 3 3
4 4 4 4
```

2)
```
4 3 2 1
4 3 2
4 3
4
```

3)
```
1 1 1 1
1 1 1
1 1
1
```

4)
```
1 1 1 1
2 2 2
3 3
4
```

5)
```
1 2 3 4
2 3 4
3 4
4
```

1)
```
Cls
For i = 1 to 4
  forj = 1 to i
    print i;
  next j
  print
  next i
```

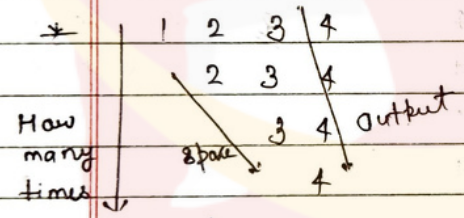
2)
```
Cls
for i = 1 to 4
  forj = 4 to i step -1
    print j;
  next j
  print
  next i
```

Date
Page

3)
```
Cls
for i = 4 to 1 step -1
  for j = 1 to i
    print 1;
  next j
  print
next i
```

4)
```
Cls
for i = 1 to 4
for j = i to 4
  print i;
  next j
  print
  next i
```

5)
```
Cls
for i = 1 to 4
for j = i to 4
  print j;
  print
next i
```




```
Cls
for i = 1 to 4
for j = 1 to i-1
  print "* * *";
  next j
  for k = i to 4
    print k; " ";
  next k
  print
  next i
```

4 3 2 1
3 2 1
2 1
1

```
Cls
for i = 4 to 1 step -1
for j = 4 to ~~i~~ i+1 step -1
  print "* * *";
  next j
  for k = i to 1
    print k; " ";
  next k
  print
  next i
```

Date
Page

Q
```
1 1 1 1
2 2 2
3 3
4
```

```
cls
for i = 1 to 4
for j = 1 to i-1
  print " ";
next j
for k = i to 4
  print i;" ";
next k
print
next i
```

Q
```
*
* *
* * *
* * * *
```

```
cls
for i = 1 to 4
for j = 1 to i
Print "*";
next j
print
next i
```

Q
```
A B C D
A B C
A B
A
```

```
cls
for i = 4 to 1
                step -1
for j = 1 to i
Print chr$(j+64);
next j
print
next i
```

Q
```
4 3 2 1
4 3 2
4 3
4
```

```
cls
for i = 1 to 4
for j = 1 to i-1
  print " ";
next j
for k = 4 to i step -1
  print k;
next k
print
next i
```

Q
```
4 4 4 4
4 4 4
4 4
4
```

```
cls
for i = 1 to 4
for j = 1 to i-1
print " ";
  next j
for k = 4 to i step -1
  print 4;
  next k
  print
  next i
```

Q
```
* * * *
* * *
* *
*
```

```
"
"
"
"
"
"
print "*";
"
"
```

Q

```
A
A B
A B C
A B C D
```

```
cls
for i = 1 to 4
for j = 1 to i
Print Chr$(J+64);" ";
  next j
  print
  next i
```

*

```
      1
    1 2 1
  1 2 3 1 2
1 2 3 4 1 2 3
```

```
cls
for i = 1 to 4
for j = 1 to 4-i
 print "***";" ";
  next j
 for k = 1 to i
  print k;" ";
  next k
 for l = 1 to i-1
   print l;" ";
   next l
  print
  next i
```

*

```
      1
    2 1 1
  3 2 1 1 2
4 3 2 1 1 2 3
```

```
cls
for i = 1 to 4
for j = 1 to 4-i
 print "***";" ";
  next j
for k = i to 1 step -1
  print k;" ";
   for l = 1 to i-1
     print l;" ";
     next l
    print
    next i
```

*

```
      0
    1 0 1
  2 1 0 1 2
3 2 1 0 1 2 3
```

```
cls
for i = 0 to 3
for j = 1 to 3-i
 print "***";" ";
  next j
for k = i to 0 step -1
 print k;" ";
  next k
for l = 1 to i
 print l;" ";
  next l
 print
 next i
```

Date
Page

Q.1)

```
    *
  * * *
* * * * *
* * * * * * *
  * * *
    *
```

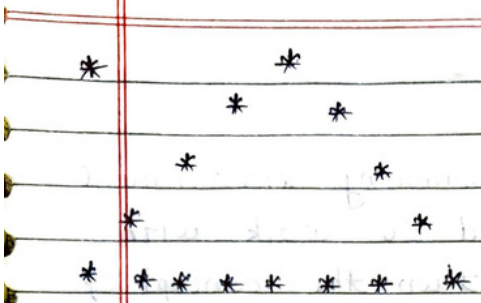
```
cls
for i = 1 to 4
for j = 1 to 4-i
 print " ";
 next j
for k = 1 to i
 print "*";" ";
 next k
for l = 1 to i-1
 print "*";" ";
 next l
print
next i
let c = 3
for i = 1 to 2
for j = 1 to i+1
 print " ";
 next j
for k = 1 to c
 print "*";" ";
 next k
 c = c-2
 print
 next i
```

Q.2)

```
    *
  * * *
* * * * *
* * * * * * *
  * * *
  * * *
  * * *
  * * *
```

```
cls
for i = 1 to 4
for j = 1 to 4-i
 print " ";
 next j
for k = 1 to i
 print "*";" ";
 next k
for l = 1 to i-1
 print "*";" ";
 next l
 print
 next i
for i = 1 to 4
for j = 1 to 2
 print " ";
 next j
for k = 1 to 3
 print "*";
 next k
 print
 next i
```

Date
Page



```
A B C D E D C B A
A B C D   D C B A
A B C       C B A
A B           B A
A               A
```

* Choise based operation -

→ Select -

In the qbasic programming environment whenever the user need to work with choise based operation then the concept of this particular statement becomes used. There are four different pattern q/c to which the choice based operation can be performed

i) Whenever a user need to perform the choise based operation on the basis of a single choise then this particular type of operation can be performed using it -

Syntax -

```
select case choise-variable
    case value
        expression
    case value
        expression
    ee case else
        expression
    endselect
```

Q WAPTA two # number and one of the arithmetical operation. Show their resultant after calculation.

```
Cls
let r = 0
Input " enter two number", a, b
Input " enter arithmetical operator", op$
    select case Asc (op$)
        case asc("+")
        r = a+b
```

```
case asc("-")
    r = a-b
case asc("*")
    r = a * b
case asc("\")
    r = a\b
case asc("%")
    r = a mod b
endselect end select
print r
```

ii) Select case choise variable -

In the BASIC programming environment, if any expression executed on multiple choises then select case statement becomes used. If any choise being matched then the expression written below it executes otherwise next which of choises being matched. If no any ~~case~~ choise matches then expression below case else execute.

Syntax

```
Select case choise variable
  case value1, value2, ----
      expression
  case value1, value2, ---
      expression
  case else
      expression
  endselect
```

Q WAPTA no. of days, month number and year value? Count how many days from starting of the year?

```
cls
let t=0
input " enter day, month & year"; d, m, y
    if d<=31 and m<=12 then
        t = t+d
    for i = 1 to m-1
        select case i
    case 1, 3, 5, 7, 8, 10, 12
        t = t+31
    case 2
        if y mod 4 = 0 then
            t = t+29
        else
            t = t+29
        endif
    case 4, 6, 9, 11
        t = t+30
    endselect
    next i
    endif
    print t
```

iii) A range of choice value available as choice then 'to' clause is used to creat the range of the ~~to~~ choice value.

Syntax:

```
case value 1 to value n
    expression
case value 1 to value n
    expression
case else
    expression
endselect
```

Date
Page

Q WAPTA any character. Check it is in uppercase, lower case, digit or symbol?

```
cls
Input "Enter any character", ch$
    select case asc(ch$)
        case 65 to 90
          st $ = "upper"
        case 97 to 122
          st $ = "Lower"
        case 48 to 57
          st $ = "Digit"
        case else
          st $ = "Symbol"
          endselect
```

iv) When certain condition being applied on variable value then the 'is' clause along with relational operator and the value with in becomes written by the user. The expression below it executes whose condition returns true. The syntax is -

```
Select case expression
Case is relational operator value
        expression
Case is relational operator value
        expression
case else
        expression
end select
```

Q WAP TA marks of a student then, show the division as follows -

marks >= 540	first
480 - 539	second
420 - 479	third
less than 420	fail

```
Cls
Input "Enter the marks", m
  select case m
    case is >= 540
      di$ = "first"
    Case is >= 480
      di$ = "second"
    case is >= 420
      di$ = "third"
    case else
      di$ = "fail"
    end select
    print di
```

* "ARRAY" in basic programming -

In basic programming environment the array is a type of variable that stores more than one values having same datatype in homogeneous memory location. The array when specified the memory location then the amount of memory is available before starting the translation of the program. The array supports static allocation of memory that means onece the memory is allocated then it do not be

be left the residental memory space that do not be utiliged in the program.

There are certain significances of the array i.e -

i) It generates the storage concept within the program.

ii) The operation like insertion, deletion, searching, storting, listing, merging etc can be perform by the use of it.

iii) There are a no. of structure like strac queue, vector, matrix, table etc can be created by the use of an array.

* Stages of array -

There are two different stages of array i.e -

i) Decleration of array -

The 'Dim' keyword is used to creat an array ~~of~~ within the BASIC programming. Whenever the name, datatype, size of array being decided then the concept of decleration of array becomes performed.

Eg:- Dim $Num(10)

ii) Initialization of array -

Whenever the name, data type, size and value of an array being decided then it is known as initialization of array.

Eg:- Dim st$ = "Computer"

* Catagory of array -

In basic language, the catagories of an array being decided by the storage format of the value. There are two different catagories of an array becomes created by the user i,e -

i) Single dimenssion array -

The array that having only one script or dimension is known as single dimension array. The value of single dimension array becomes available or stored in column. The single dimension character type of an array is known as string whereas numeric type is known as vector.

ii) Double dimension array -

When more than one script or dimension becomes available to store the data within the array variable is known as double dimenssion array. In double dimension array, the first script is known as no. of rows and second belongs to no. of columns. The double dimension character type of array is known as table whereas numeric type is known as matrix.

```
Dim name(10)
Dim num(3,3)
```

Date
Page

Q WAPTA to numbers show the list of numbers along with difference in maximum & minimum number?

```
Cls
Dim num(10)
print " Enter 10 no.s"
for i = 0 to 9
  input num(i)
  next i
 ma = num(0)
  mi = num(0)
 for i = 0 to 9
if num(i) > ma then
    ma = num(i)
endif
 if num(i) < mi then
    mi = num(i)
endif
next i
print" list of value are"
    for i = 0 to 9
   print "max =", ma
    print "min=", mi
print "Difference" ma-mi
```

```
Dim num(10)
 let c = 0
print " Enter 10 nos"
 for i = 0 to 9
   input num(i)
 next i
Input " Enter any no. to search", s
      for i = 0 to 9
       if num(i) = s then
          c = 1
          exist for
          endif
          next i
       if c = 0 the
      print " Not found"
      else
     print " Found"
      endif
```

```
Dim num(10)
Print " Enter 10 nos"
  for i = 0 to 9
  input num(i)
 next i
 for i = 0 to 9
  for j = i+1 to 9
   if num(i) > num(j) then
       k = num(i)
       num(i) = num(j)
       num(j) = k
     endif
     next j
     next i
     print "After sorting"
        for i = 0 to 9
        print num(i)
        next i
```

```
Dim num(12)
print "Enter 10 nos"
 for i = 0 to 9
   input num(i)
  next i
Input "Enter new number, position", n, p
       for  i = 9 to p-1  step -1
         num(i+1) = num(i)
         next
        num(p-1) = n
       for  i = 0 to 10
         print num(i)
          next i
```

```
Dim num(12)
print "Enter 10 nos"
 for i = 0 to 9
 input num(i)
 next i
input "Enter the position to delete", t
       for i = p-1 to 9
       num(i) = num(i+1)
       next i
         for i = 0 to 8
           print num(i)
            next i
```

Date
Page

* Single dimension character type array -

In the BASIC programming, when character based operation being required then character type array is created. The starting index location is from 1 and goes up to their length.

WAPTA any name, show their ascii value along with their character?

```
Cls
print "Enter the name"
Input Name$
for i = 1 to len(name$)
    ch$ = mid$(Name$, i, 1)
    print ch$, asc(ch$)
    next i
```

WAPTA any name, check the name is in alphabet or not? (Name may be of more than words)

```
Cls
print "Enter the name"
input name$
for i = 1 to len(Nam$)
    ch$ = ~~isalpha~~ mid$(Name$, 1, 1)
    ~~If ch$ = isalpha(Name ch$) then~~
        ~~print yes~~
If isalpha(ch$) or asc(ch$) = 32 then
    c = 0
else
    c = 1
exit for
```

Date
Page

```
endif
next i
if c=0 then
print " In alphabet"
else
Print " Not "
endif
```

Q1) WAPTA any string, check it is pallindrom or not?

Q2) WAPTA any string, seperate the consonent and vowel of the string then show it?

Q3) WAPTA any string and a character, check the character is available or not?

3)
```
Cls
c=0
Input "Enter the string", st$
Input " For search", h$
for i = 1 to len(st$)
    ch$ = mid$(st$)
    If ch$ = h$ then
        c=1
    endif
next
If c=0 then
    print " Not available"
else
    print " available"
endif
```

* Double dimension numeric array-

When the array contains two script or dimension then it is known as double array. The double dimension numeric array is known as matrix.

Eg:- Dim mat(3,3)

* The first value belongs to number of rows and second belongs to number of columns.

(Raw)	0 (Columns)	1	2
0	0,0	0,1	0,2
1	1,0	1,1	1,2
2	2,0	2,1	2,2

Q. WAPTA 3 * 3 elements of matrix, show it?

```
cls
Dim mat(3,3)
print " Enter 3 * 3 elements of matrix"
 for i = 0 to 2
 for j = 0 to 2
   input mat(i,j)
 next j
 next i
 print " The matrix is"
    for i = 0 to 2
    for j = 0 to 2
  print mat(i,j); " ";
  next j
  print
  next i
```

Date
Page

Q WAPTA 3 * 3 elements of matrix, transpose their elements.

```
cls
Dim mat(3,3)
print "Enter 3 * 3 elements of matrix"
  for i = 0 to 2
    for j = 0 to 2
    input mat(i,j)
  next j
  next i
print " After transpose"
    for i = 0 to 2
    for j = 0 to 2
  print mat(j,i);" ";
    next j
    print
    next i
```

Q WAPTA 4*3 elements of matrix add the rows and columns value?

```
cls
let r = 0
let c = 0
Dim mat(4,3)
print "Enter 4 * 3 elements of matrix"
    for i = 0 to 3
     for j = 0 to 2
     input mat(i,j)
     next j
    next i
print " Addition of rows values matrix"
```

```
for i = 0 to 3
 for j = 0 to 2
 r = r + mat(i,j)
next j
Print "Sum of", i+1, "Row=" ; r
   r = 0
 print
  next i
Print "Addition of columns value of matrix"
  for i = 0 to 3
  for j = 0 to 2
      c = c + mat(j,i)
  next j
  print "The sum of", i+1, "col=" ; c
     print
       c = 0
      next i
```

WAP TA 3*3 elements of matrix check the matrix is symmetric or not?

```
cls
let c = 0
dim mat(3,3)
print "Enter 3*3 elements of matrix"
  for i = 0 to 2
  for j = 0 to 2
  input mat(i,j)
next j
next i
for i = 0 to 2
 for j = 0 to 2
  if mat(i,j) <> mat(j,i) then
     c = 1
endif
next j
next i
if c = 0 then
print "Matrix is
          symmetric"
else
 print "Matrix is
    not symmetric"
endif
```

Q. WAPTA 4 * 3 elements of matrix, show the maximum and minimum value along with their position of row & column.

```
cls
Dim mat(4,3)
print "Enter the no."
 for i = 0 to 3
 for j = 0 to 2
  input mat(i,j)
 next j
 next i
ma = mat(0,0)
mi = mat(0,0)
 for i = 0 to 3
 for j = 0 to 2
 if ma < mat(i,j) then
      ma = mat(i,j)
      row = i
      column = j
   End if
    If mi > mat(i,j)
      mi = mat(i,j)
      r = i
      c = j
   Endif
  next j
  next i
print "Maximum", ma; "Raw = ", row; "Column", column
 print "Minimum=", mi; "Row=", r ; "Column", c
```

Sparse matrix - That matrix which contain mostly zero values.

Q WAPTA 3*3 elements of matrix check it is sparse matrix or not?

```
Cls
Dim mat (3,3)
print " Enter 3*3 elements of matrix"
  for i = 0 to 2
  for j = 0 to 2
    input mat(i,j)
  next i
  let c = 0
  for i = 0 to 2
   for j = 0 to 2
       if mat(i,j) = 0 then
              c = c+1
        Endif
      next j
      next i
      If c >= 5 then
      print " Sparse matrix"
        else
          print " Not sparse matrix"
      & endif
```

Date
Page

Q. Multiplication of matrix (3*3) -

```
cls
Dim m1(3,3) , m2(3,3) , m3(3,3)
print " Enter the elements of first matrix"
    for i = 0 to 2
      for j = 0 to 2
      input m1(i,j)
    next j
    next i
print " Enter the elements of second matrix"
    for i = 0 to 2
     for j = 0 to 2
     input  m2(i,j)
   next j
   next i
  for i = 0 to 2
  for j = 0 to 2
    m3(i,j) = 0
  for k = 0 to 2
    m3(i,j) = m3(i,j) + m1(i,k) * m2(k,j)
   next k
   next j
 print " Matrix after multiplication"
    for i = 0 to 2
    for j = 0 to 2
      print m3(i,j) ; " ";
  next j
  print
  next i
```

Date
Page

* Double dimension character array - [Dim st$(10)]
When the double dimension array is created then the no. of rows are fixed but the columns are in variable nature. To work with the double dimension character array the two ways are used to do so -

i) When the user need to work on the names, ~~character then the loop is~~ both ~~for~~ ~~so~~ not character by character then loop is only iterated for rows.

ii) When the user need to work on the names, character by character then the loop is both for rows and columns are executed.

Q WAPT arrange 5 names in ascending order using length.

```
Cls
Dim name$(5)
Print "Enter 5 names"
  for i = 0 to 4
    input name$(i)
  next i
   for i = 0 to 4
     for j = 0 to 4
        if len(name$(i)) > len(Name$(j)) then
       st$ = name$(i)
       name$(i) = name$(j)
        name$(j) = st$
       endif
     next j
     next i
```

```
Print "List of names
   after sorting"
  for i = 0 to 4
    print name$(i)
   next i
```

Date
Page

* Subprogram -

In the BASIC programming environment whenever a user need to divide the large program into a number of small programs then the concept of subprogram is created. The subprogram in BASIC language being divided into two different categories that is -

i) Function -

When a subprogram is being divided into a number of sections and after execution if it returns certain value to the translator then the concept of function is used. To creat the function a user can use the "function ---- end function" statement. The function of BASIC language is divided into two different types that is -

a) System defined function -

The function whose name, parameter and the way of working and result if it already defined inside the translator of the language is known as system defined function or library function.

b) User defined function -

When the name, parameter, processing and resultant if defined by the user then it is known as user defined function.

Date
Page

Procedure or Subroutine -

The subprogram when written and execute by the user and after the completion if no ~~vo~~ any value being returned to the translator then the concept of procedure or subroutine is created. To create the procedure a user uses "Sub --- end sub" statement.

* When the subprogram is created then the "declare" statement preceded the procedure or function name. The subprogram is used for the following purposes, that is -

a) Modularity -

When a larger program is divided into a number of small sections then each one is known as module and the way through which module is created is known as modularity.

b) Code reusability -

Once the program is written but used many times then the concept of code reusability becomes used. Due to that the size of the program gets reduced, wastage of memory do not happens as well as chance of error being reduced hence the efficiency of the program increased.

c) Abstraction -

When a portion of the program being hide from other portion of the program then the concept of abstraction being achieved. The subprogram supports the abstraction feature.

Date
Page

* Type of subprogram -

When the subprogram concept being used then there are two main aspects being verified that the subprogram having certain parameter or not as well as wether it returns certain value or not after execution.

i) Subprogram without argument and no return type
ii) Subprogram having certain argument but no " "
iii) Subprogram having no argument but have return "
iv) Subprogram having argument and have " "

* Stages of subprogram -

There are three different stages of the subprogram that is -

i) Declaration -

When the name, parameter of the subprogram being decided then this stage is used.

ii) Definition -

When the syntax of the subprogram being written as per requirement then it is used.

iii) Calling -

When the subprogram need to execute then the concept of calling is used.

NOTE:-

1. Function when executes then their value must be stored inside the function itself.

2. The function when called to executes then its returning either store in the same type of variable as returning states or output being directly displayed.

3. When the procedure is created then the value do not be stored inside the variable but displayed at the time if calling.

4. The 'call' statement is used at the time of calling procedure.

* Example of Function -

```
Declare function che $(n)
 cls
input " Enter the numbers"; n
 st$ = che$(n)
 print st$
 end
Function che $(n)
 if n>0 then
   che$ = " Positive number"
elseif  n < 0  then
     che$ = Negative number"
else
     che$ = "Zero value"
endif
end function
```

* Example of procedure -

```
declare sub check(n)
cls
input "Enter the number"; n
call check(n)
end
sub check(n)
if n>0 then
print "positive number"
elseif n<0 then
print "Negative number"
else
print "zero value"
endif
endsub
```

* System defined function / Library function -

The fuction whose name and processing is already defined inside the language translator is known as system defined function. There are different catgories of system defined function that is -

i) Text function / string function -

The function that works on character value is known as text function. The list of functions are -

a) len(text) -

It is used to count how many characters are available in the string.

```
let st$ = "Computer"
print len(st$) - 8
```

b) Ucase$(text) :-
To convert the lower case character into upper case it is used.
let st$ = "basic"
ch$ = Ucase$(st$)
print ch$ - BASIC

c) Lcase$(text) -
let st$ = "BASIC"
print lcase$(st$) - basic

d) Left$(text$, number) :-
To extract certain number of character from the left side of given text this particular function is used.
let st$ = "Computer"
ch$ = left$(st$, 3) Com

e) Right$(text$, number) -
let st$ = "Program"
ch$ = Right$(st$, 4) gram

f) Mid$(text$, number, number) -
When a user need to extract certain numbers of characters from any specified position then it is used.
let st$ = "Computer"
ch$ = mid$(st$, 4, 3) - Put

g) Str$(number) -
To convert the numeric value into string value it is used.
let num = 1234;
ch$ = str$(num)

h) Space$(number)-

To creat a number of space where it is written in the program then it is used.

ch$ = "Computer" & space$(10) & "Program"

i) Chr$(number)-

When a user need to convert the numeric ASCII value into character ASCII value then it is used.

let ch$ = chr$(65) A

j) Asc(text)-

let n = asc("A") 65

k) Instr$(text, character)-

It is used to search the specified character within the string and show their position.

let x = instr$("Monsoon", "o") 2

l) instr$(text, number, character)-

When a user need to search certain specified character from a given position then it is used.

let x = instr$("Monsoon", 3, "o")

Numeric function-

In BASIC programming environment whenever a user need to work on numeric values then the concept of numeric functions becomes used by the user.

1. Abs (number) -

It is used to find the absolute value of the given number. Generally in converts negative value into positive value.

```
let num = -20
x = abs(num)
print x = 20
```

2. Asc (char) -

When a user need to convert the character value into numeric ascii value then it is used.

```
let ch = "A"
num = asc(ch)
print num        65
```

3. Chr $ (number) -

When a user need to convert the numeric ASCII value into characters representation then this particular function is used.

```
let num = 98
let ch$ = chr$(num)
print ch$        b
```

4) Exp (number) -

When a user need to calculate the exponential value of the given inputted value then this function is used. The value of exponential is 2.716

```
let num = 2
let ex = exp(num)
print ex        20.08
```

5) Int (number) -

To get the precision value from the given fraction value it is used.

```
let x = 20.456
print int(x)      20
```

6) Mod (number1, number2)

It is used to divide the first position number by second position and get the remainder value.

```
print mod(5,2)      1
```

7) Sgn (number) -

When a user need to verify the number as per positive, negative or zero then its returning is 1, -1 and 0 resp.

```
let num = -34
x = sgn(num)
prin x          -1
```

8) Sqr (number) -

It is used to find out the square root value of the given number.

```
print sqr(25)      5
```

9) Sin (number) -

It is used to convert the degree value into radian value. The value passed in sin function being convert to its equivalent value.

```
input " Enter the angle in degree"; d 30
    r = d * 3.14/180
    p = sin(r)
print " Sin of"; d; "degree is"; p  .499
```

10) Cos (number) -

It is used to find the cosine value of the inputted degree value.

```
input "Enter the angle in degree"; d
    r = d * 3.14/180
    p = cos(r)
print " Cosin of"; d; "degree is"; p
```

11) Tan (number) -

```
input " Enter the angle in degree"; d
    r = d * 3.14/180
    p = tan(r)
print " Tan of"; d; " degree is" p
```

* Conversion function -

When a user need to convert one data type value into another data type value then this particular function is used.

i) Oct $ (number) -

It is used to convert the decimal number into octal fromat. The octal value is always greater than decimal value.

```
let d = 64
print oct $(d) = ' 100
```

2. Hex $ (number) -

To convert the decimal number into hexadecimal number this particular function is used. The hexadecimal value is always smaller than decimal value.

```
let d = 57
print hex$(57)    39
```

3) Cint (number) -

From the given fraction value whenever the decimal value only be return then this particular function is used.

```
let d = 123.567
print cint(d)    123
```

4) Cdbl (number) -

It is used to convert the numeric value into double type of value.

```
let d = 123.456789
print cdbl(d)
```

5) Csng (number) -

When a user need to show the three digits in scale value from the given fraction value then it is used.