No: _____ Date: __/__/____

30.05.22

Module : cs102.3 programming in c Language.

Q. what is a computer ?

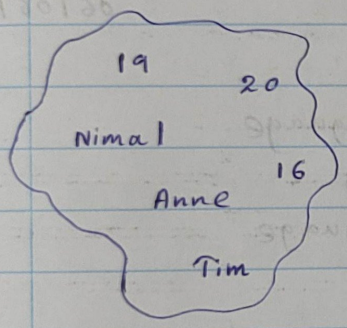
A. • a computer is an electronic Device which is used to convert data into information.

• Data are raw facts. In computing these data transformed into information.

• In other words information are processed facts (Data).

• converting Data into information is called as Data processing.

Data processing



Process

Name	age
Nimal	16
Anne	19
Tim	20

Data (Input)

Information (output)

Total computer System.

- Hardware
  1) Input Devices
  2) output Devices
  3) cpu / processor
  4) Storage.
- Software.
  - System s/w
    - operating System
    - utilities
      - virus guard
      - firewall
  - application s/w.
    - Generic (ready made)
      - MS office
      - Adobe s/w
      - Games
    - Bespoke (coustamize
      - Banking app
      - Library s/w.

No: _____ Date: __/__/____

> Software.

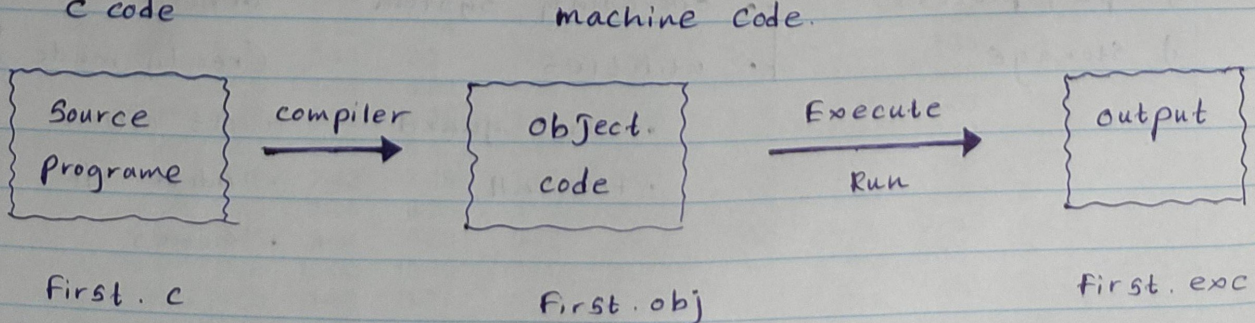
- simply software are collection of programs.
- a program is a set of instructions given to the computer by using a programming language.
- programming languages are broadly divided into 3 catogaries,
  1) machine language (binaries)(0,1)
  2) Assembly language (set up symbols)
  3) High level language (similar to natural english language statement)
     ex:- C, Java, Python, PHP.

06/06/22

- There are 3 language translators in use,
  1) Assembler → Assembly language.
  2) compiler
  3) Interpreter } High level language.

> Language translators

- The pourpose of programming language translators is to convert the cord written by programmer into machine code.
- The compiler process the whole setup lines at once. but in Intepreter the translation happens line by line.



No: Date: / /

C programming

- First program in c language - Hello world program
- install a c editor such as codeblocks, turboc or any other c compaitable editor.
- Type the following c sourcecode and save the file.

```
#include <studio.h>
int main ()
{
        printf ("Hello world !");
}
```

- compile the programe using a editor.
- During a compilation if any error exist it will be displayed correct the errors and recompile the program
- to view the output run the program

Data input in c language.

- In programming the input is associated with variables. a variable is a temporary memory location.
- you can store different types of values inside a variable
- prior to using variables it must be diclared.

variable diclaration

in programming variabels are declared with the data type.

ex:- int no1, no2

variable names.

No: _____ Date: __/__/____

Q) write a c programm to input 2 numbers and display the total.

```
#include <stdio.h>
int main()
{
    // variable declaration
    int no1, no2, total;
    // data input
    printf("Enter first number");
    scanf("%d", &no1);
    printf("Enter second number");
    scanf("%d" &no2);
    /// process
    total = (no1 + no2);
    // output
    printf("the total is %d", total);
}
```

Data type	conversion Specifier
int	%d
float	%f
char (single character)	%c
double	%lf

Q) write a programm to input two numbers with fractional values and display the avaraye value.

```
#include <stdio.h>  ← header files
int main()          standerd input & output
{
    // variable declaration.
    float no1, no2, avg;
    // Input
    printf("Enter first number");
    Scanf("%f", &no1);
    printf("Enter Second number");
    Scanf("%f", &no2);
```

No: Date: / /

```
// Process
Avg = (no1 + no2) / 2;
// output
printf("the total is %.2f\n", Avg);
}
```

20.06.2022

- working with strings

```
#include <stdio.h>
int main()
{
    // character array
    char name[20];
    printf("Enter your name");
    scanf("%s", &name);
    printf("Hey %s", name);
}
```

%c → char. (single character)
%s → (multiple characters)

a) write a c program to allow the user to input name, birth year & display name with age.

```
#include <studio.h>
int main()
{
    // character array
    char name[30];
    printf("Enter your name");
```

No: _____ Date: __/__/____

```
#include <stdio.h>
int main ( )
{
    int byear, age;
    char name [20];
    printf ("Enter your name");
    Scanf ("%s", &name);
    print f ("Enter your Birth year");
    Scanf ("%d", &byear);
    age = 2022 - byear;
    printf ("Hey %s you are %d years old", name, age);
}
```

Control Structures

- In a computer program the flow of program statements is called as control structure.
- There are three types of control Structures,

1) Sequence -
   * In here statements are executed line by line in the order.

2) Seletions - The flow of statements are controled by the conditions provided by the following two constructs
   a) if ( ) b) switch ( )

3) Iteration / repetition (Loops)
   * Set of statements will repeatadly execute until the given condition remains true. this is represented by using the following 3 types of loops.

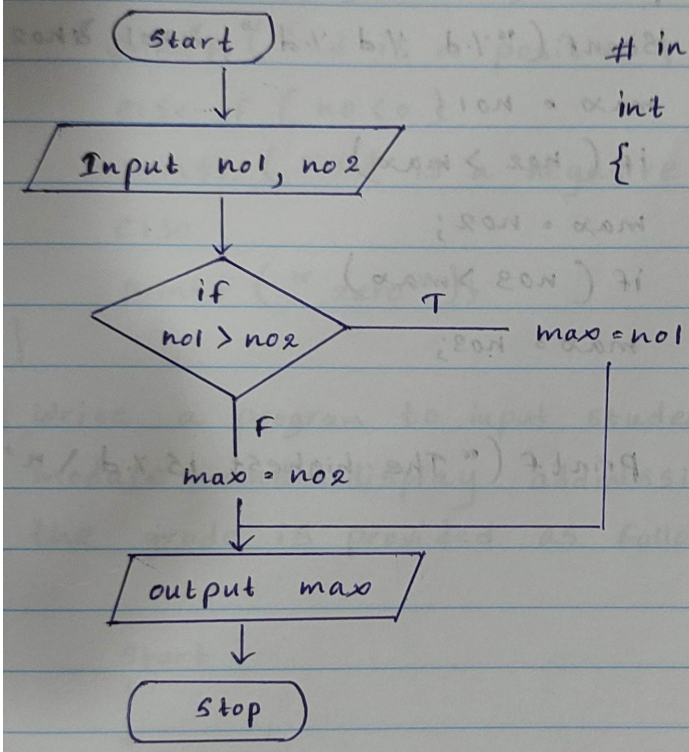
   a) For loop b) while loop c) do while loop

No: _____ Date: __/__/____

Selection Structure

if ( ) condition

Example 1 - write a program to allow the user to input 2 numbers and display the Highest number.




```
#include <stdio.h>
int main ( )
{
    int no1, no2, max;
    printf ("Enter two numbers");
    Scanf ("%d %d", &no1, &no2);
    if (no1 > no2)
    max = no1;
    else
    max = no2;
    // output
    printf ("The Highest is %d \n",
            max);
}
```

modify the above program to allow the user to input 3 numbers and display the highest number.

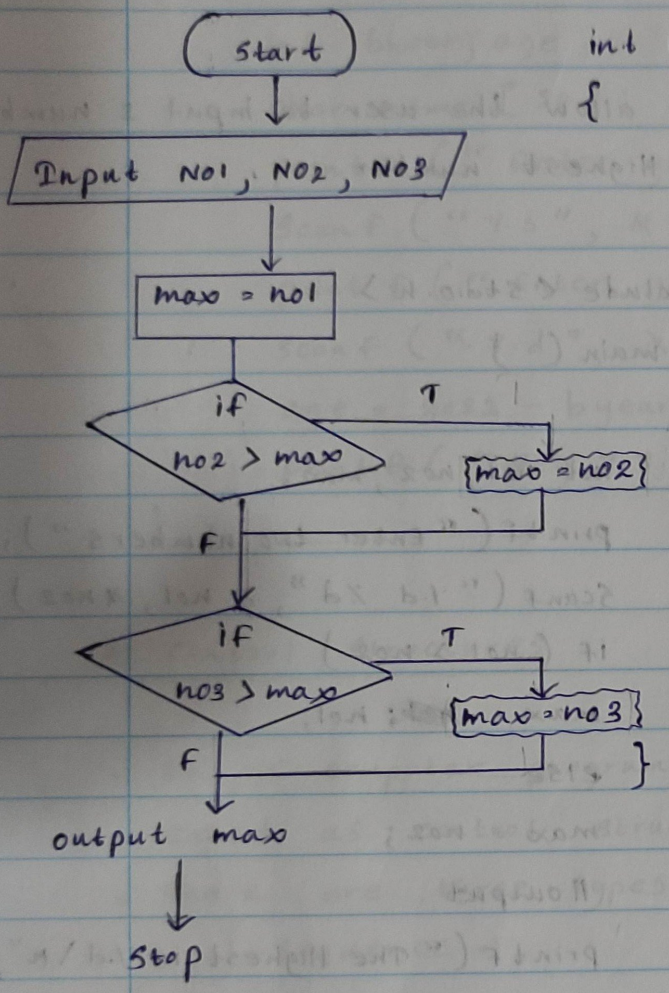
```
#include <stdio.h>
int main ( )
{
    int no1, no2, no3, max;
    printf ("Enter three numbers");
    Scanf ("%d %d %d, &no1, &no2, &no3);
    if (no1 > no2)
    max = no1;
```

No: Date: / /

lateral thinking

If (no1 > NO3)

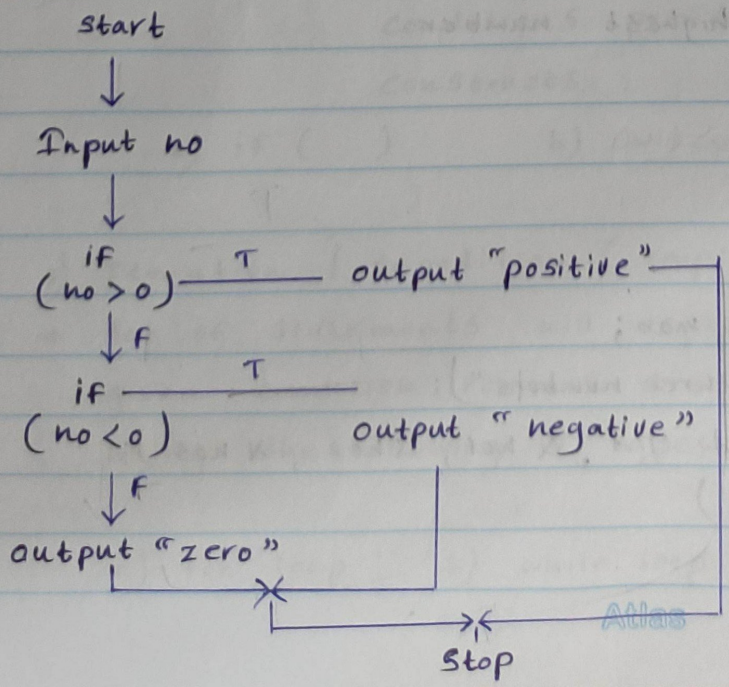
max = no1;




```
#include <stdio.h>
int main( )
{
int no1, no2, no3, max;
printf ("Enter three numbers");
Scanf ("%d %d %d", &no1, &no2, &no3);
max = no1;
if ( no2 > max)
max = no2;
if ( no3 > max)
max = no3;
printf ("The highest is %d \n", max);
```

27.06.22.

Q) write a program to input a number and display the entered number is a positive, negative or a zero.

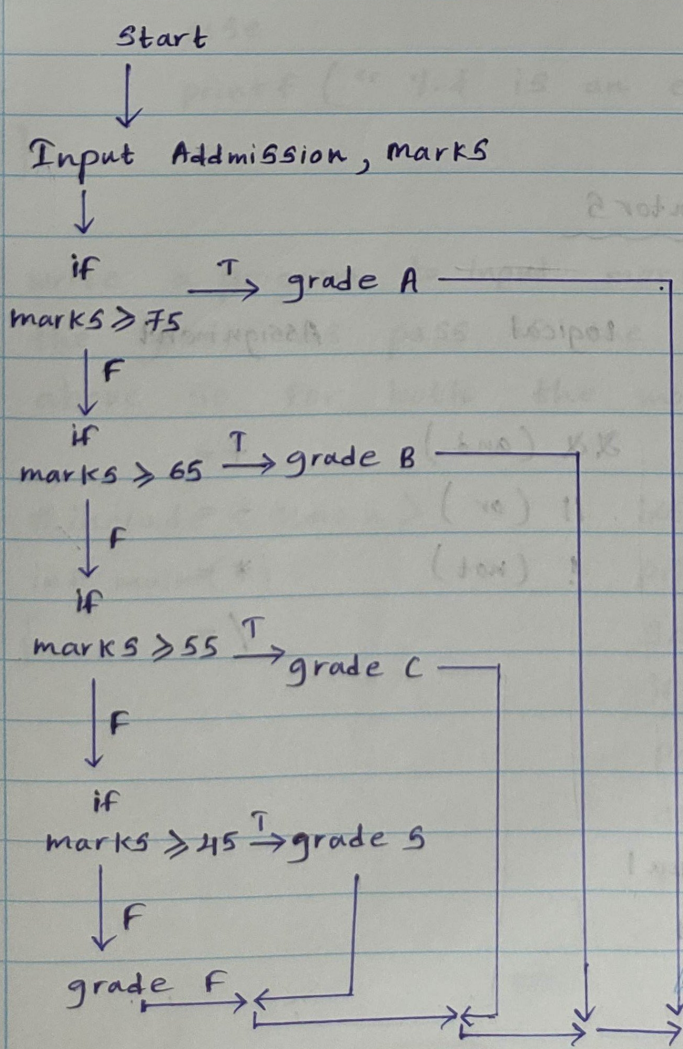


No: _____ Date: __/__/____

```
#includ < stdio.h >
int main ( )
{
    int no;
    printf ( " Enter a number " );
    Scanf ( " %d ", & no );
    if ( no > 0 )
    printf ( " %d is a positive \n ", no );
    else if ( no < 0 )
    printf ( " %d is a negative \n ", no );
    else
    printf ( " zero " );
}
```

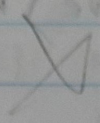
marks	grade
≥ 75	A
65 - 75	B
55 - 65	C
45 - 55	S
< 45	F

Q) Write a program to input student addmission number marks of a module and display addmission number with the grade. the grade is provided as follows.



* char represent within a single code / character ' '.

No: _____ Date: __/__/____



> operators

Arithmetic	Relational	Logical	Assignment
+	>	&& (and)	+=
-	<	|| (or)	-=
*	>=	! (not)	*=
/	<=		/=
% (modulous /remainder)	!=		
	==		

(!=) → not equal

(==) → equal

No: _____ Date: __/__/____

* Increment / Decrement

++ (increment by one)	x++ (post increment)
-- (decrement by one)	x-- (" Decrement)
	++x (pre increment)
	--x (" Decrement)

1) write a program to input a number, check and display the enterd number is an odd or even number.

```
#include <stdio.h>
int main( )
{
    int no, ans;
    printf ("Enter a number");
    scanf ("%d", &no);
    ans = no %2;
    if ( ans == 1 )
    printf (" %d is an odd number \n", no);
    else
    printf (" %d is an even number \n", no);
}
```

2) write a program to input marks of 2 modules and display the grade as pass if the student has obtain marks above 50 for both the modules?

```
#include <stdio.h>
int main( )
{
    int m1, m2;
    printf ("Enter marks of two modules");
    scanf ("%d %d", &m1, &m2);
    if ( m1 >=50 && m >=50 )
    printf( "pass \n");
    else
    printf (" Fail \n");
}
```

No : Date: __/__/__

* 1) if (condition)
{ ..... }

2) if (condition)
{ ..... }
else
{ ..... }

3) if (condition 1)
{ ..... }
else if (condition 2)
{ ..... }
else
{ ..... }

• Logical operators.

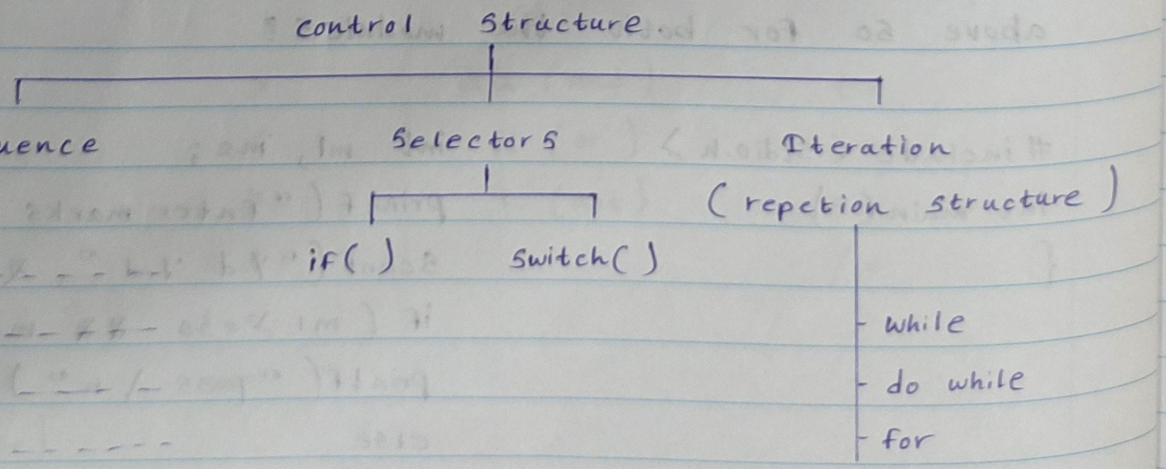
if ! (a > b) → if not a greater than b on.07.22
if (a > b && a > c) → a > b and a > c
if (m1 ≥ 50 || m2 ≥ 50) → m1 ≥ 50 or m2 ≥ 50

* ex:- write a program to input 2 numbers and display the following outputs by comparing the two numbers.

1) is that numbers are equal ?
2) is that numbers are not equal ?
3) which number is the highest ?

• Assignment operaters

2) write a output of the following program.



(lines)

No: _____ Date: __/__/____

Switch

* The structure is an alternative to the if structure. in certain programs using switch you can write the program in less number of statements.

a) write a program to output the month when enter the number

```
#include < stdio.h >
int main ( )
{
        int mo;
        printf ( " Enter month number " );
        scanf ( " %d ", &mo );
if      if ( mo == 1 )
        printf ( " January " );
else if if ( mo == 2 )
        printf ( " February " );
else if if ( mo == 3 )
        printf ( " march " );
else if if ( mo == 4 )
        printf ( " April " );
else if if ( mo == 5 )
        printf ( " may " );
else if if ( mo == 6 )
        printf ( " June " );
else if if ( mo == 7 )
        printf ( " July " );
        :
else if if ( mo == 12 )
        printf ( " december " );
else    if ( mo > 12 && mo < = 0 )
        printf ( " the entered number is invalid " )
```

No: _____ Date: __/__/____

code using switch:

```
#include <stdio.h>
int main ( )
{
        int  m;
        printf ( " Enter month number " )
        Scanf ( " %d " , & m );

        Switch ( m ) ← .switch value must be intiger or char
        {
              case 1 : printf ( " January " ); break ;
              case 2 : printf ( " February " ); break;
              case 3 : printf ( " march " ); break;
              case 4 : printf ( " April " ); break;
              case 5 : printf ( " may " ); break;
              case 6 : printf ( " June " ); break;
              case 7 : printf ( " July " ); break;
              case 8 : printf ( " Auguest " ); break;
              case 9 : printf ( " September " ); break;
              case 10 : printf ( " octorber " ); break;
              case 11 : printf ( " november " ); break;
              case 12 : printf ( " December " ); break;

              default : printf ( " %d is invalid number " ); break;
        }
}
```

* from logical operator part.

```
#include <stdio.h>
int main ( )
{
    int no1, no2 ;
    printf ( " enter first number " );
    Scanf ( " %d \n " , & no1 );
```

No: Date: / /

```
printf ("enter second number");
scanf ("%d\n", &no2);
if (no1==no2)
printf ("both numbers are equal");
if (no1 != no2)
printf ("numbers are not equal");
if (!(no1 > no2))
printf ("the second value is greater than first \n");
else
printf ("the first number is greater than second");
}
```

Q) write the output of following program.

```
#include <stdio.h>
int main()
{
  int a=10;
  a+=20;
  printf ("a= %d \n", a);
  a-=5;
  printf ("a= %d \n", a);
  a*=10;
  printf ("a= %d \n", a);
  a/=5;
  printf ("a= %d \n", a);
}
```

output

a = 30
a = 25
a = 250
a = 50

* ++ (Increment by one) and ++x (Pre Increment)

* -- (Derement by one) and --x (pre decrement)

No: Date: / /

```
#include <stdio.h>
int main ( )
{
    int x = 10;
    x++;
    printf ("%d \n", x );
    ++x;
    printf ("%d \n", x);

    int y = 10;
    printf ("%d \n", y++ );
    printf ("%d \n", ++y );

    int g = 10;
    printf ("%d \n", ++g );
    printf ("%d \n", g++ );

    int z = 10;
    printf ("%d \n", z-- );
    printf ("%d \n", --z);

    int w = 10;
    printf ("%d \n", --w );
    printf ("%d \n", w--);
}
```

```
11
12
10
12
11
11
10
8
9
9
```

* Incriment / Decrement operators

++ (Increment by one)
-- (decrement by one)
x ++ (post Increment)
++ x (pre Increment)
x -- (post decrement)

No : _____ Date: __/__/____

--x (pre Decrement)

Q) get the output

```
# include < stdio.h >
int main ( )
{
   int x = 10 , y = 20 , z;
   z = x++ + y++ ; // assign the sum of x and y to z
   // then Increment the value of x and y by one.

   printf ( "%d \n", z );
```

output : 30

Q)

No: Date: __/__/____

11.07.22.

Q) write a program using switch structure to allow the user to input a character and check and display the entered character is vowel or not an vowel.

```
#include <stdio.h>
int main( )
{
    char vowel;
    printf( "input a character \n" );
    scanf( "%c", & vowel );
    switch ( vowel )
    {
        case 'a': printf( "a is a vowel \n"); break;
        case 'e': printf( "e is a vowel \n"); break;
        case 'i': printf( "i is a vowel \n"); break;
        case 'o': printf( "o is a vowel \n" ); break;
        case 'u': printf( "u is a vowel \n" ); break;
        default : printf( "%c is not a vowel \n", vowel ); brea
    }
}
```

• It's not neccesary cause it's the end of the code (last line)

Iteration (repition Structure)

* in this structure the given statements repeatedly execoute untill the given condition remains true.
* all the loop must begin with the value and their must be a end point

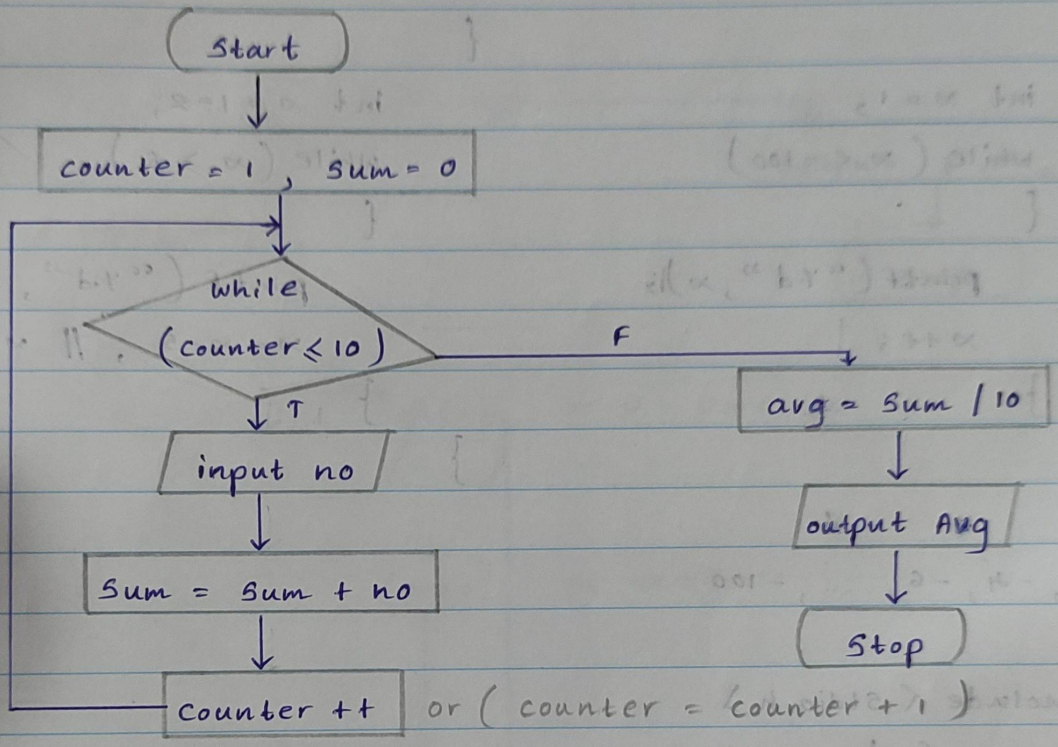
1) 'while' loop →

Syntax :

```
while ( condition )
{
// body
}
```

No: _____ Date: __/__/____

ex:1) Draw a program flow chart and write a C program to allow the user to input 10 intigers and then to display the average value.




```
# include < stdio.h >
int main ( )
{
    int   counter = 1 , sum = 0 , no;
    float  avg ;

    while ( counter <= 10)
    {
        printf ( " enter %d number ", counter);
        Scanf ( " %d ", & no );
        Sum = ( Sum + no);
        Couter ++ ;
    }
    Avg = ( float) Sum / 10 ;
    Printf ( " the average is %f \n ,avg " );
}
```

No: Date: / /

1) 1, 2, 3, ..., 100

```
#include<stdio.h>
int main()
{
    int x=1;
    while (x<=100)
    {
        printf("%d", x);
        x++;
    }
}
```

2) 100, 95, 90 ..., 5

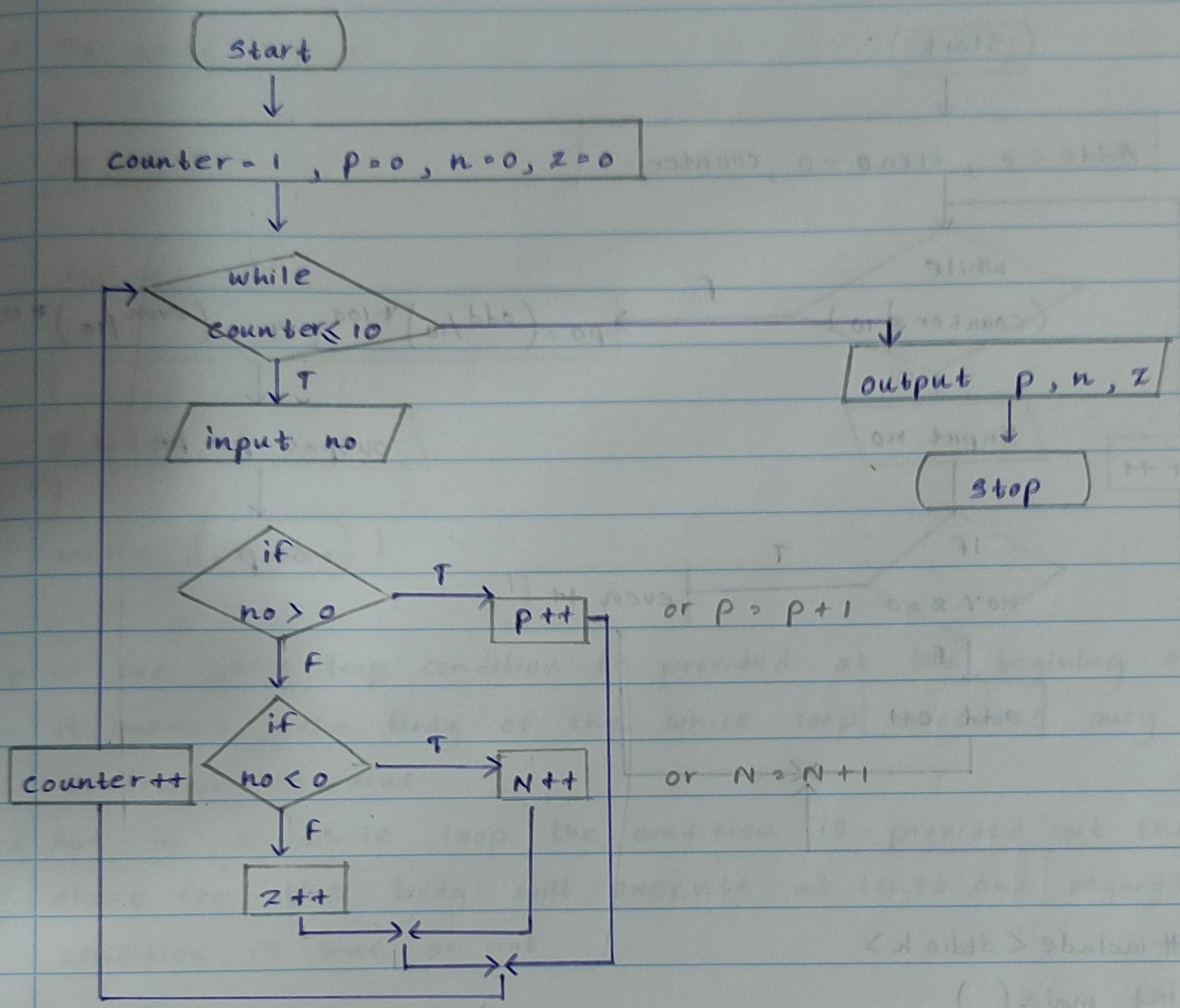
```
#include <stdio.h>
int main()
{
    int x=100;
    while (x>=5)
    {
        printf ("%d", x);
        x=x-5 , // x-=5
    }
}
```

3) -2, -4, -6, ..., -100

```
#include <stdio.h>
int main()
{
    int x = -2;
    while (x>=-100)
    {
        printf("%d", x);
        x=x-2;
    }
}
```

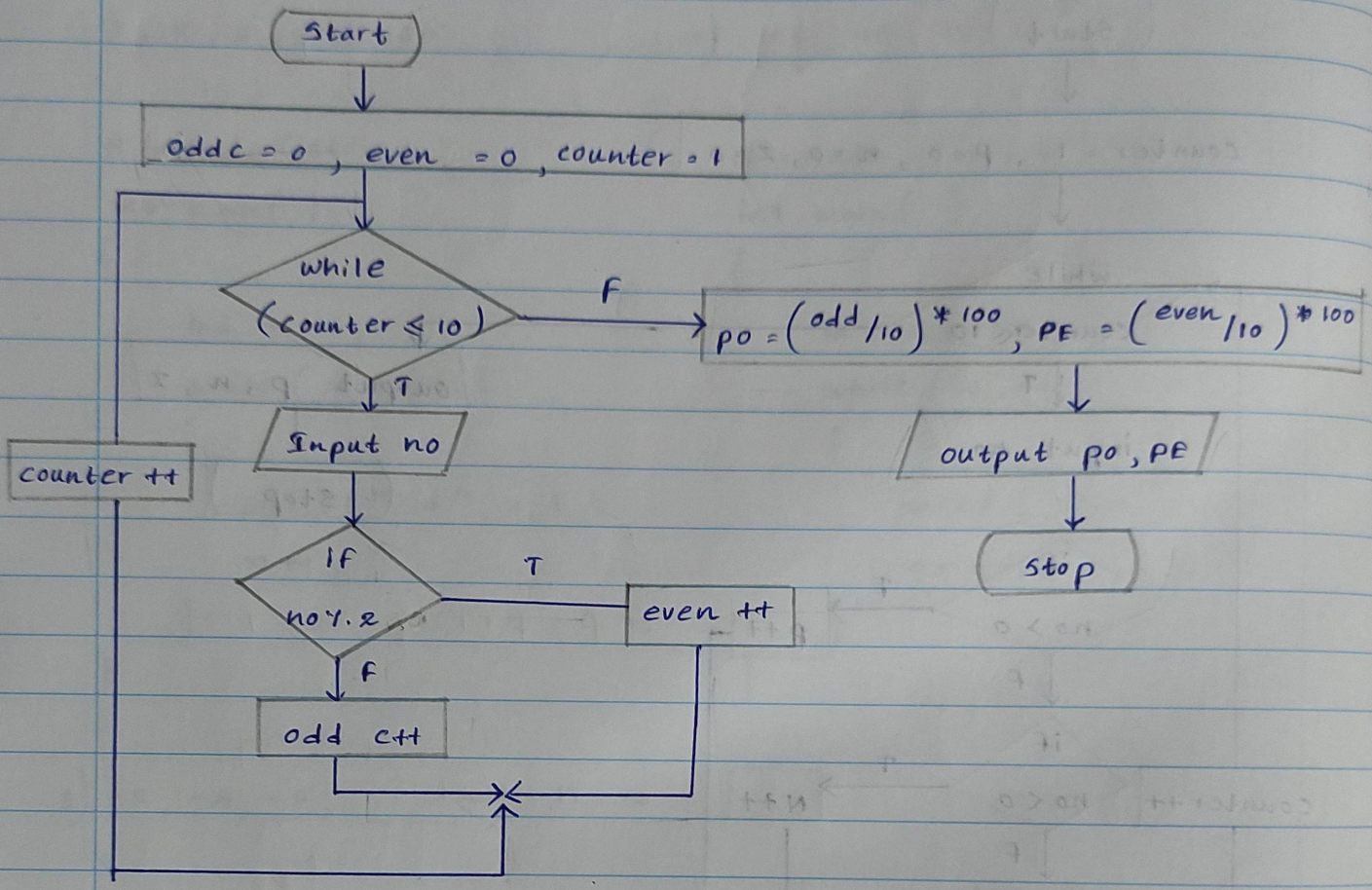
4) Draw a program flow chart and write a program to allow the user to input 10 numbers and to display the total number of positives, negatives and zeros in the entered number series.

No: Date: / /



18.07.22

Q) Draw a flowchart and write a c program to allow the user to input 10 numbers and display the presentage of odd, even numbers in the entered number series.




```
#include <stdio.h>
int main( )
{
   int odd=0, even=0, couter=1, no;
   float po, pe;
   while ( counter <=10)
   {
      printf ( " enter %d number", counter );
      scanf ( " %d ", &no );
      if (no%.2 ==0 )
      evenc++;
      else
      oddc++;
      counter++; }
po= (float) (oddc/10) * 100.0; pe = (float) (erenc/10) * 100.0;
printf ( " precentage of odd's %..2f \n ", po);
printf ( " percentage of even's %.2f \n ", pe );
}
```

No: Date: / /

Do while loop.

do while.

Syntax:

```
do
{
// body
}
while (condition)
```

* in the while loop condition is provided at the begining of the loop. it means, the body of the while loop executes only if the condition is true.
* but in do while loop the condition is provided at the end. there for the body will execute at least one regardless the condition is true or not.

Questions.

01) print 10-100 using do while loop. gap is 10.

```
#include <stdio.h>
int main()
{
    int x = 10;
    do
    {
        printf("%d", x);
        x = x + 10;
    }
    while (x <= 100);
}
```

No: _____ Date: __/__/____

2) print 1 - 100 using do-while loop.

```
# include < stdio.h >
int main ( )
{
```

3) print 100 - 1 using do-while loop

```
# include < stdio.h >
int main ( )
{
```

No: _____ Date: __/__/____

25.07.2022

01) by using do while loop write a program to allow the user to input 10 numbers and the total count of odd, even numbers in the entered number series.

```
#include <stdio.h>
int main ( )
{
    int counter = 1, oddc = 0, evenc = 0, no;
    do
    {
        printf ("enter %d number", counter);
        scanf ("%d", &no);
        if ( no % 2 == 0 )
        evenc ++;
        else
        oddc ++;
        Counter ++;
    }
    while ( Counter <= 10 );

    printf ("the total odd count is %d \n", oddc);
    printf ("the total even count is %d \n", evenc);
}
```

02) write a program to allow the user to input two numbers and display the output of Basic Aurithmetic operations. in the program user can repeatedly select the Aurithmetic Calculatio

```
#include <stdio.h>
int main ( )
{
```

No: _____ Date: __/__/____

```
int no1, no2, opt, add, sub, mul, Div;
// allow the user to input two numbers.

do
{
    // display the meme.
    printf ("select an option");
    scanf ("%d", opt);
    switch (opt)
    {
        // case.
    }
}
while (opt != 5);
}
```

2)
```
# include <stdio.h>
int main ( )
{
  int no1, no2, opt;
  printf ("enter two numbers");
  scanf ("%d %d", &no1, &no2);
  do
  {
    printf ("1. + \n 2. - \n 3. * \n 4. / \n 5. exit \n");
    printf ("select an option \n");
    scanf ("%d", &opt);
    switch (opt)
    {
      case 1: printf ("the addition is %d \n", (no1 + no2)); break;
      case 2: printf ("the subtra. is %d \n"), (no1 - no2)); bre
      case 3: printf ("the multipli. is %d \n"), (no1 * no2)); bre
      case 4: printf ("the division is %d \n"), (no1 / no2)); bre
  }
  while (opt != 5); }
```

No: _____ Date: __/__/____

> for loop.

a) print 1 to 100 using for loop.

```
#include <stdio.h>
int main ( )
{
    int x;
    for (x=1; x<=100; x++)
    printf ("%d", x);
}
```

a) printf 100 to 2 gap is 2 between 2 digits.

```
#include <stdio.h>
int main ( )
{
    int x;
    For (x=100; x>=2; x-=2)
    printf ("%d", x); }
```

a) print 5 to 100 gap is 5

```
#include <stdio.h>
int main ( )
{
   int x;
   for (x=5; x<=100; x+=5)
   printf ("%d", x);
}
```

using nested for loops display the following outputs

* * * * *
* * * * *
* * * * *

No: _____ Date: __/__/____

01. 08. 2022.

01) By using two for loops display the following output.

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

```
#include <stdio.h>
int main()
{
    int r, c;
    for(r=1; r<=5; r++)
    {
        for(c=1; c<=5; c++)
        {
            printf("*");
        }
        printf("\n");
    }
}
```

by this to this

change c<=5 to (c<=r)

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

for(c=1; c<=5; c++) → c<=r

➤ Arrays

- An array is data structure.
- we can use arrays to store set of values with simillar data types.
- In c language we discuss the following two types of arrays.
  1) Single dimensional arrays (vectors)
  2) multi dimensional arrays (matrix)

example 01

declare a single dimensional array to store 10 integer values. input 10 values in to the array and display the values.

01) How to declare an array ?

```
int   arr[10];
 ↓     ↓   ↓
Data  array array
type  name  size
```

Steps

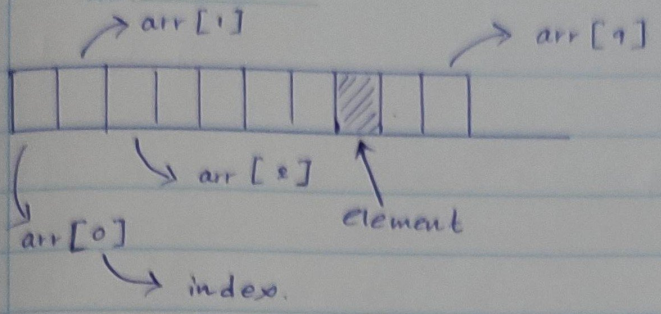
// declare (1)

// input (2)

// display (3)

No :

Date: / /



* in an array the elements are uniquely identify by using an index
* the index always begins with zero and ends with array size. -1.
  (array in 10 ends with a / 20 → 19)
* you can input values into an array by using a loop.
* again you can use a loop to display values.

No: _____ Date: __/__/____

15.08.2022

functions

- A functions in c language has it's on task to perform.
  ex:- printf and scanf are functions which is used to display and input data.

- The functions are broadly divided in to 2 categories.
  i) pre - defined functions.
  ii) user - defined functions.

i) pre - defined functions.

Q) write a c program to use at existing function Sqrt( ) which is a part of math.h header file. and display the square root values of the numbers from 1 to 100.

```
#include< stdio.h >
#include <math.h>
int main ( )
{
    float x, ans; // here Sqrt is the pre- defined function.
    for ( x=1.0; x<=100.00; x++ )
    {
        ans = sqrt (x);
        printf ( " square root value of %.2f is %.2f \n ", x, ans)
    }
}
```

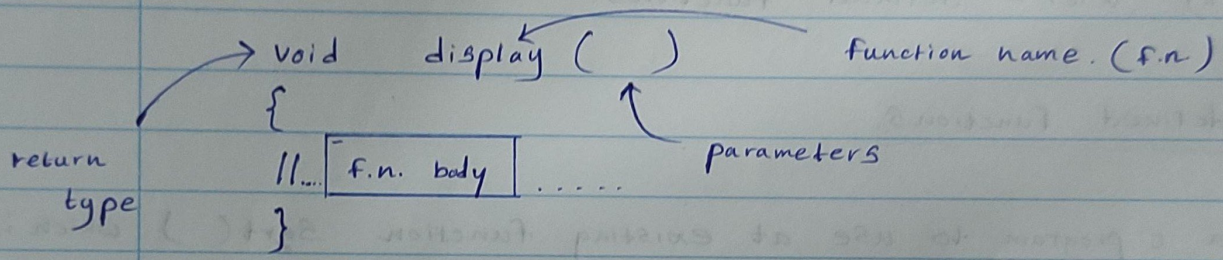
2) user - defined functions.

- In the user define function we have to provide the logic of the program

No: Date: / /

- There are 4 categories of user define functions.
  1) no - return type , no parameters.
  2) no - return type , with parameters.
  3) with return type , no parameters.
  4) with return type , with parameters.

1) NO - return type , no parameters.



- whenever we have using 'void' it's no-return type.
- when we use other type's it's with- return type.

```
#include <stdio.h>
// a funestion with no return type and no parameteres
// called funcsion
Void display ( )
{
    int x;
    for ( x=1 ; x<=10 ; x++ )
    printf ( "c is very easy to lean \n");
}
int main ( )
{
    // call the above functions
    display ( );  -> 1 represent 10 becouse x<=10
    display ( );  -> 2 " 20
}
```

No: _____ Date: __/__/____

- In the above program we are calling the display function inside the main method.
- you can call the function display inside the main method any number of times.
- The main advantage of function is it's reusability.

Q) write a C program to create following two functions.

1) A function to display your name 20 times.
2) A funtion to display your school name 10 times.

after creating the functions call the above functions inside main method.

```
# include < stdio.h >
Void displayName ( )
{
    int x ;
    for ( x = 1 ; x <= 20 ; x++ )
    printf ( " NSBM \n " );
}
void displayschool ( )
{
    int x ;
    for ( x = 1 ; x <= 10 ; x++ )
    printf ( " NSBM green university \ n " );
}
int main ( )
{
    displaytheschool ( ) ;
    displayname ( ) ;
}
```

No: Date: __/__/____

2) no-return type with parameters.

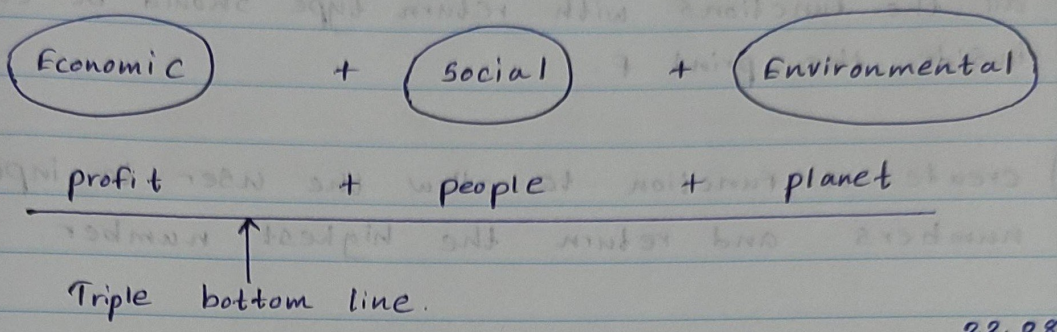
a) create a function which accept two integers as parameters. and display the total.

```
#include <stdio.h>
void findsum( int a , int b )
{
    int sum = a + b ;
    printf ( " The total is %d \n " , sum );
}
int main ( )
{
    int x , y ;
    printf ( " Enter two numbers " );
    scanf ( " %d %d " , &x , &y );
    findsum ( x , y );
}
```

b) write a program to allow the user to input 2 numbers as parameters and display the highest number.

No: Date: 06/06/22

Sustainability development.



22.08.22

01) Create a function which eccept mark as the parameters, find and display the grade of module. the grade is calculated as for the following table.

Marks	grade.
⩾ 75	A
50 - 75	B
< 50	f

after crate the function call the function inside the main method.

function name

return type → void findgrade (int marks) ← parameters.

3) with return type, no parameters

in a function return type names any data type in c language other than the term void.

return type was given in the question

02) create a function which allows the user to input 2 indigers and return the total.

* a function with return type always returns a single value.

No: _____ Date: __/__/____

Q) create a function to allow the user to pass a integer parameter and display the enterd number is a positive, negative or zero.

No: Date: __/__/__

all the functions with return type should be called inside the print f.

a) create a function to allow the user to input two numbers and return the highest number.

No: Date: __/__/__

Q) write a program using loop to find a factorial value of a given number. for the above create a function which accept a number as a parameter and return the factorial value.

factorial value. → 5!

5 × 4 × 3 × 2 × 1 = 120.

```
int Find fact ( int n )
{
//. . . . . .
}
```

```
#include <stdio.h>
int find fact ( int n )
{
    int ans = 1;
    for ( x = 1 ; x <= n ; x++ )
    ans = ans * x ;
return ans;
}
int main( )
{
    printf ( " The answer is %d \n ", findfact (5) );
}
```

The above program is written using the loops but the same program can be implemented by using reccurtion without loops.

No :

Date: / /

Pointer

A pointer is a special type of variable. it is use to display the memory address of an existing variable.

&x → memory address of x.

```
#include <stdio.h>
int main ( )
{
    // a variable.
    int x = 10;
    // a pointer
    int *p;
    // this is to store the memory address of an existing
       variable.
    p = &x;
    // display the memory address.
    printf ("memory address of x is %.p \n", p);
```