Ques.3 What is an Operating System? Explain main functions of an operating system & write the name of operating system available in the market?

Ans An Operating system is a system software that manages computer hardware & software resources & provides common services for computer programs.

1→ Resource management :- Operating system also known as the resource manager. means Operating System will manages all the Resources those are attached to the System means all the resources like memory & processor & all the input output devices those are attached to the system are known as the Resources of the computer system & all the input output devices those are attached to the operating system will manage all the Resources of the system.

2→ Storage management :- Operating System also controls the all storage operation means how the data or files will be stored into the computers & how the files will be accessed by users etc.

3→ Process management :- The Operating system also treats the process Management means all the processes those are given by the user or the process those are System's own process are handled by the operating system.

4→ Memory management :- Operating system also manages the memory of the computer System means provide the memory from the process.

→ Extended machine :- Operating system also behaves like extended machine operating system also provides us sharing of files b/w multiple users, also provides some Graphical environment & also provides various language for communication.

→ Mastermind :- Operating system also perform functions & for those reasons we can say that Operating system is a mastermind. It provides Booting without an Operating System & provides facility to increase the logical memory of the computer system by using the physical memory of the computer system & also provides various type of formates like NTFS & FAT file systems.

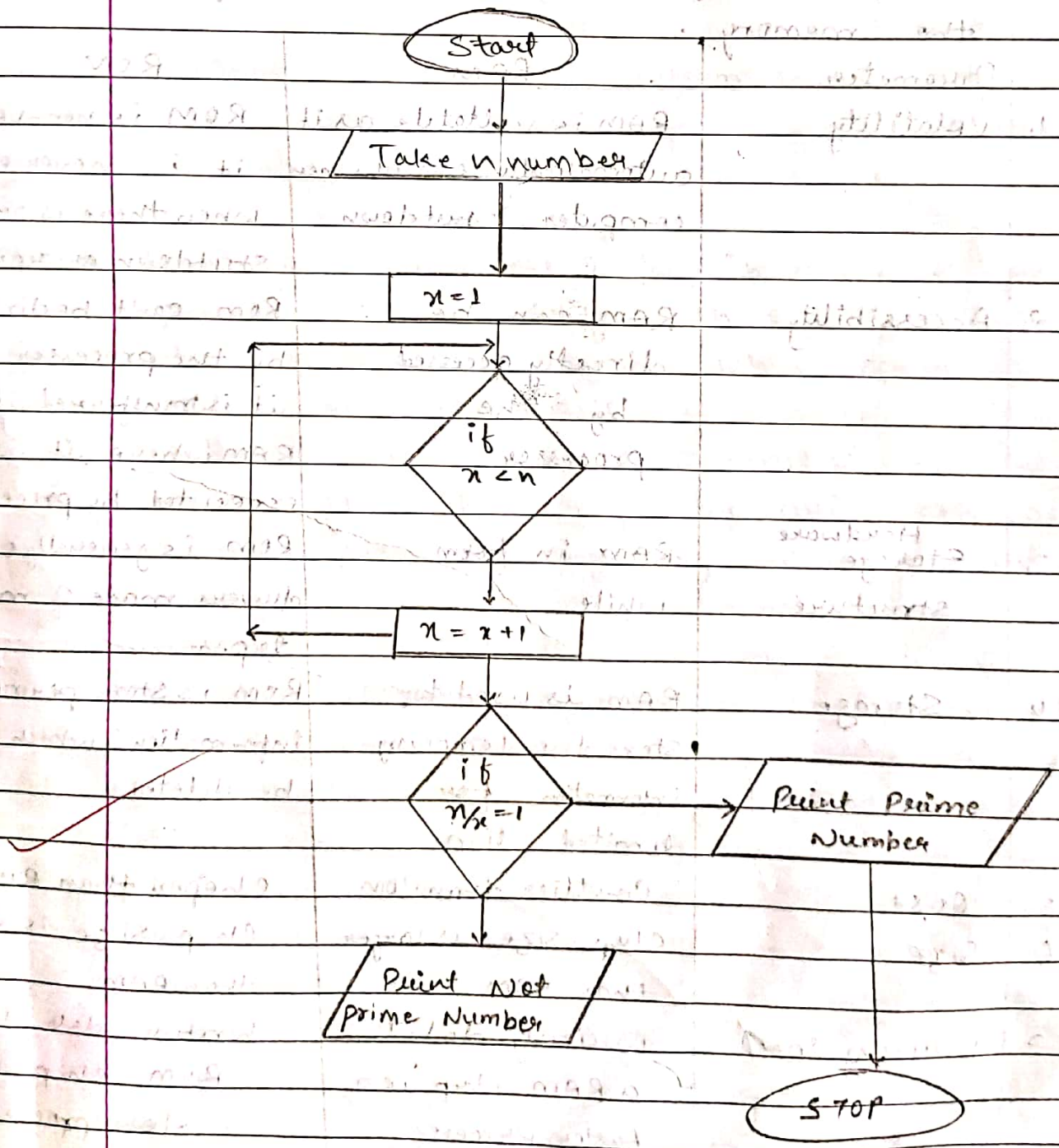
Types of operating systems :-

1. Single & multi-tasking
2. Single & multi-user
3. Distributed
4. Templated
5. Embeded
6. Real time
7. Library

Ques 4. What is Flowchart ? Draw a flowchart to print a given number is prime or not.

Ans A flowchart is a type of diagram that repents an algorithm, workflow or process. The flowchart shows the steps as boxes of various kinds, & their order by connecting

the boxes with arrows. The diagrammatic representation illustrates a solution model to a given problem. Flowchart are used in analyzing, designing, documenting or managing a process or program in various fields.



Date __/__/__
Page _____
STUDY BUDDIES

Ques. 5. What is RAM ? Write down diff. b/w RAM & ROM ?

Ans:- Random - access memory is form of computer data storage, that allows data items to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory.

Parameter	RAM	ROM
1. Volatility	RAM is volatile as it automatically erased when computer shutdown	ROM is non-volatile it is never erased when there is any shutdown or restart
2. Accessibility	RAM can be directly accessed by the processor	Rom can't be directly by the processor since it is transferred into RAM where it is executed by processor
3. Hardware Storage structure	RAM in form while	ROM is generally optical drivers made of magnetic tapes
4. Storage	RAM is used to store the temporary information for limited time	ROM is store permanent information which cannot be deleted
5. Cost	Costlier than ROM	Cheper than RAM
6. Size	Chip size is larger than ROM	Chip size is smaller than RAM
7. Writing speed	Writing data to a RAM chip is a faster process	Writing data to a ROM chip is a slow process

Date __/__/__
Page _____
STUDY BUDDIES

8.	Storage limit	A RAM chip can store multiple gigabytes of data, up to 16GB or more per chip.	A ROM chip typically stores only several gigabytes(MB) of data, up to 4MB or more per chip.
9.	Examples	Static & Dynamic RAM	PROM, EPROM, & EEPROM are types of ROM
10.	Content	Store information temporarily	stores information permanently.

Ques. 6 What are Operators ? Explain their categories, precedence & associativity with eg. of each.

Ans An Operators is a symbol which operates on a value or a variable

→ Arithmatic Operators :- Arithmatic operators are used for performing arithmetic operations

(i) Unary operators (they only operate on a single operand)

Unary +	
Unary -	
& (Address extraction operator)	Return the addr of variable
* (Defreferencing operator	Pointer to a variable
size of ()	Return the size of given data
(< type cast >) type cast operator	use for temporary conversion of data type
++ Increment	Increases value by 1
-- Decrement	Decreases value by 1

⇒ Increment & Decrement operator must be either pre or post.

Date __/__/__
Page
STUDY BUDDIES

(iii) Binary operator (operate upon two operands)

+	Addition Operator
-	Subtraction operator
*	Multiplication Operator
/	Division Operator
%	Modulus Operator

→ Relational Operators :-

Rebtional op. are used in decision making & loops

Operator	Meaning of Operator	Example
==	Equal to	5==3 return 0
>	greater than	5>3 return 1
<	less than	5<3 return 0
!=	Not equal to	5!=3 return 1
>=	Greater than or equal to	5>=3 return 1
<=	less than or equal to	5<=3 return 0

→ logical Operators :-

logical Operators are the expression containing logical operator return either 0 or 1 depending upon whether expression results true or false

Operator	Meaning of Operator	Example.
&&	logical AND. True only if all operands are true	if((c==5) && (d>5))
||	logical OR. True only if either one operand is true	if((c==5)||(d>5))
!	logical NOT. True only if the operand is 0	if !(c==5)

Date __/__/__
Page _____
STUDY BUDDIES

→ Bitwise Operators :-

During computation, mathematical operations like: addition, subtraction, addition & division are converted to bit level which makes processing faster & saves power.

Operators	Meaning of Operators.
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR
~	Bitwise complement
<<	Bitwise Shift left
>>	Bitwise Shift right

⇒ Precedence of operators

~~If more than~~ The precedence of arithmatic operators (*, %, /, +, -) is heigher than relational operators (==, !=, >, <, >=, <=) & precedence of relational operator is heigher than logical operators (&&, || & !).

e.g. (1 > 2+3 && 4) ~~return~~

(1 > (2+3)) && 4 return → 0

⇒ Associativity of operators

The expression on the left is executed first & moves towards the right.

e.g 1 == 2 != 3

0 != 3

return → 1

Ques. 8. What is Function ? Type of function parameter passing method of function ?

Ans A function definition has two principle component The first line (including the arguement declaration & the body of the function.

The first line of a function definition contains the type specification of the valve returned by the function followed by the function name & set of arguement seperated by commas & enclosed in parenthess each arguements is preceded by its associated typedeclaration

Two type

Standard library function :- The standard library function are built in function in c programming to handle tasks such as mathematical computation, I/O processing, string handling, etc.

These function are defined in header file when you include the header file these function are available for use. for example : The printf ( ) is a standard library function send formatted output the

screen (display output on screen). This function is defined in "stdio.h" header file these are other numerous library function defined under <stdio.h> such as Scanf(), printf, getch()

Advantage :- have gone through Hence
① These function is use multiple rigorous testing & are easy to use
② In the process they are able to create the most efficient code optimized for maximum performance.
③ It saves valuable time & your code may not always be the most efficient
④ The function are portable

C Header file
<assert.h> = program assertion function
<ctytype.h> = Character type function.
<locale.h> = localization function
<math.h> = mathematics function
<set jump.h> = Jump function

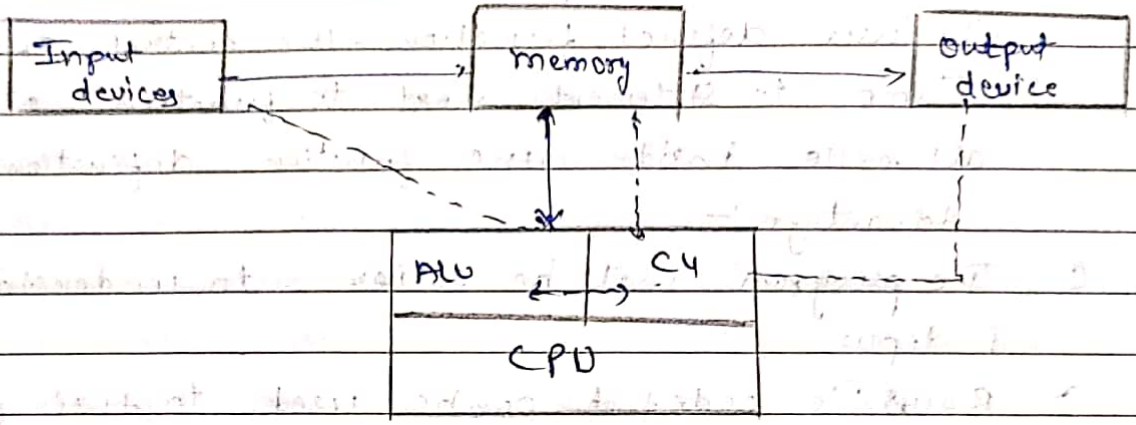
User defined function :- As maintained earlier, C allow programmers to define function such function created by the user are called user defined function

```
# include <stdio.h>
void function name()
{ ...
}
int main()
{ ... function Name(); ... }
```

The execution of main program begins from the main() function.

Date __/__/__
Page _____
STUDY BUDDIES

Ques 8.9 What is Draw of the computer system & explain briefly.



The computer has four main component.

(i) Input devices
(ii) memory
(iii) Central processing unit
(iv) Output devices

The input devices use to input data to computer it converts input information to the form which is usable by the computer whatever input is supplied by the input device first goes to the memory. The C.P.U. acts as brain CPU is responsible for the overall working of all component of the computer It consists of two parts : Arithmatic, logic unit & central unit. The ALU performs arithmatic operation & conducts the comparison of information for logical decision. The control unit is responsible for sending & receiving signal from to all component The dotted line in the above diagram represent the communication of control of CU the data come input devices to memory it is proceed in ALU & Result

Date __/__/__
Page _____
STUDY BUDDIES

is ~~started~~ stored back in the memory the proceed result are converted to a form that can be understand easily by human being & is displayed with an output device. The memory of computer is two type primary memory & secondary memory fastes in speed loss in size easier it consist of ROM & RAM. ROM is very small amount of memory used to make the computer ready to work this process is called booting.

RAM consists all type of data & intermediate & temporary data to be used by the CPU infact CPU can be work /process only that which present in the ram. Any data present in secondary memory needs to be brought to RAM & only then can it be used by the CPU.

Ques 10 Compare the following :-

	Compiler	Assembler
(i)	The compiler is a computer program that reads a program written in a source language translates it into assembly language & forward assembly level code to assembler also show error 'when program was wrong'	The assembler takes as a input the assembly code generated by the compiler & translate into relocatable machine code.
(ii)	The compiler takes as input the preprocessed code generated by preprocessor	It take assembly code as input.

3.	The compilation takes place in two phases analysis & synthesis phase input gas analyzer logical synter analyzer, semantic analyzer takes place ~~in inter~~ via intermediate code, generator code optimices code generator	The phase detects the inten identify & allot address to them in the second phase the assembly code transmitted to binary code.
4.	The assembly code generated by the compiler is a mnemonic version of machine code	relocatable machine code generated by assembler as binary relocatable code.

	Hardware	Software
1.	Physical parts of the computer called hardware	A set of instruction given to the computer is called software
2.	We can touch, feel & see hardware	We can't touch & feel Software
3.	Hardware is constructed using physical material & component.	Software is developed by writing instruction in program language.
4.	Computer is the hardware which operates under the control of a software	The operation of computer controll through the Software.
5.	If hardware is ~~not affected~~ damaged by computer virus it is replaced by new one	If Software is curpted its back up copy is reinstalled
6.	Hardware is not transferred from one place to another.	It is transferred from one place to another place electronically through network

Compiler	Interpreter
Compiler first scans the whole program for error. If no error, the compiler will convert source code to machine code called object code.	Interpreter reads each line of source code statement line by line into machine code & also gets it executed.

Date __/__/__
Page _____
STUDY BUDDIES

② Pro Program run fast ; because the translated version already available & so can run directly	Program is easy, because it stops translation when the first error is met
③ No error can escape the eye of the compiler	Certain erroneous statement may escape detection, because the interpreter checks only those statement for error that it executes
④ Compiler based programs occupy more space on disk	Interpreter based program less space on disk.
⑤ Programming language like PASCAL, FORTAN, COBOL, C, C++, C# = AVA	Programming language like B, BASIC, FOXPRO, Python.

28/11/18

Date __/__/__
Page _____
STUDY BUDDIES

Ques: Explain the diff. b/w parameter passing mechanism call by value" & "call by reference". which is more efficient & why?

	Call by value	Call by reference
Description	A function to pass data or value to other function	A function to pass data or value to other function
language used	c based programing languages	c based programming languages such as C++ Java but not c itself.
Purpose	To pass arguments to another function	To pass arguments to another fun.
Arguements	A copy of actual arguments is passed to respective formal arguments	Reference the location or address of the actual arguments is passed to the formal arguments.
Changes.	Changes are made in the nonpersonal copy made inside the fun are not reflected on other functions	Any changes made in the formal arguments will also reflect in the actual arguments. Changes made inside the fun are reflected outside the functions as well.
value modification	Original value is not modified	Original value is modified
Safety	Actual arguments remain safe, they cannot be modified accidentally	Actual arguements are not safe. They can accidently modified

cello

Date / /
Page
STUDY BUDDIES

Ques 3. What do you mean by pointer? How pointers can be used with the string of the character?

Ans A pointer is a variable that represent the location of data item such as variable & array element pointer can be used to pass information back & forth b/w a function & its reference point:-

- Pointer provide a way to return multiple data item from a function via function argument
- Pointer also permit reference to other function to be specified as argument to given function.
- Pointer are also closely associated with array & therefore provide an alternate way to access individual array element.
- Pointer provide a convenient way to represent multidimensional array allowing single dimensional array to be replaced by lower dimensional array of pointer.

→ Pointer only hold an address they cannot hold all the character array it means that or when we use Char * to keep track of string the character array containing the string must already exist.

Char label [] = "single"

Char label2 [] = "Married"

Char * labelptr

labelptr = label

We would have something like the following in memor (eg Supposing that the array label started at memory address 2000 etc.

Date / /
Page
STUDY BUDDIES

label @ 2000
| s | n | g | l | e | v |
label2 @ 3000
| M | a | r | k | i | e | d | 10 |
label ptr @ 4000
| 2000 |

Ques 3 What do you mean by structure in c ? Explain with an example how we declare & initialize the structure.

Ans A structure is a user define data type ~~type~~ in c. A structure is a creates a data type that can be used to group ~~struct~~ item of possibly different type into a single type struct type keyword is used to create a structure

```
struct address
{ char name [50];
  char street [100];
  char city [50]; int pin;
}
```

A structure variable can either be declared with structure declaration or as a seperate declaration like basic data type.

```
// A variable declaration with str.
struct point
{ int x, y;
} p1; // The variable p3 is declared with point
// A variable declaration like data type
struct point
{ int x, y; }
```

Date / /
Page
STUDY BUDDIES

int main
{ Struct point P1; // The variation P1 is declared
}

Structure member cannot be initialized with declaration for example the following c program fails in compilation.

Struct point
{
int x=0; // compiler error; cannot initilize member
int y=0; // compiler error; cannot initilize member
};

The reason for above error is simple when data type declared no memory is allocated for it memory is allocated for it when variable are created Structure member can be initilize using curly braces {} for example following is valid initialization Struct point.

{ int x,y;
};

int main
{ Struct point P1 = {0,1} // Avoid initialization member get value of y.

Que. Discuss with the help of examples the action of break Statement & continue Statement.

Ans → Break Statement :- The break Statement terminates the loop (for, while, & do-while loop) immediately when it is encountered.

example :-

```
int main()
{int i;
 double number, sum=0;
 for (i=1; i<=10; i++)
 { printf ("Enter a n%d : ", i);
  Scanf ("%.lf", &number);
  if ( number<0)
  { break ; }
 sum += number
}
 printf ("Sum=%.2lf", sum);
  return 0;
  }
```

```
Enter a n1 : 2.4
Enter a n2 : 4.5
Enter a n3 : 3.4
Enter a n4 : -3
Sum = 10.30
```

Date / /
Page
[STUDY BUDDIES]

-> Continue Statement :- The continue statement skips statement after it inside the loop.

Its syntax is : continue;

The continue statement is almost always used with if .. else statement

example :-

```
int main()
double number, sum=0;
for(i=1; i<=10; i++)
{printf("Enter a n%d: ",i);
 scanf("%lf", &number);
 if(number<0)
 {continue;}
  sum+=number
}
printf("sum=%.2lf",sum);
return 0;
}
```

Out-put :

```
Enter a n1 : 2.2
Enter a n2 : 5.5
Enter a n3 : 4.4
Enter a n4 : -3.4
Enter a n5 : -45.5
Enter a n6 : 34.5
Enter a n7 : -4.2
Enter a n8 : -1000
Enter a n9 : 12
Enter a n10 :
Enter a n Sum = 59.70
```

Ques What do you mean by dynamic memory allocation? Define calloc(), malloc(), realloc() & free() function.

Ans Dynamic memory allocation refers to managing system memory at runtime. Dynamic memory management in programming language is performed via a group four function namely malloc(), calloc() realoc() & free(). These four dynamic memory allocation functions of the c programming language are defined in c standard library headed file <stdio.h>. Dynamic memory allocation uses the head space of the system memory.

calloc() :- Calloc() allocates a user-specified number of bytes & initializes them to zero. unlike malloc(), this function takes two arguments the number of memory chunks to be allocated & the size of each memory chunk.

int *buffer = (int*)calloc (NUMBER OF ELEMENTS ; sizeof(int));

realloc() :- realloc() is used when an allocated block on dynamic memory needs to be resized. If necessary, realloc() completely reallocates the memory block. Assume you make an initial memory allocation using malloc() as follows:-

int *buffer = (int*) malloc (size);

Date __/__/__
Page _____
STUDY BUDDIES

Now if you need to change the size of *buffer to new size, use realloc() as follows:-
buffer = (int*) realloc (buffer, new_size);

Malloc() :- The malloc function reserves a block of memory of the specified number of bytes. And it returns a pointer of type void which can be casted into pointer of any form
Syntax :- ptr = (cast-type*) malloc (byte size)
example :- (int*) malloc (100 * sizeof (int));

free() :- Dynamically allocated memory created with either calloc() or malloc() doesn't get freed on their own. You must explicity use free() to release the space.
Syntax : free (ptr);

Friend
02/01/19

	STRUCTURE	UNION
① Access Members	We can access all the members of structure at anytime	Only one member of union can be accessed at anytime.
② Memory allocation	Memory is allocated for all variables	Allocates memory for variable which variable require more memory.
③ Initialization	All members of structures can be initilized	Only the first member of a union can be initialized
④ Keyword	'Struct' keyword is used to declare structure	'Union' keyword is used to declare union.
⑤ Syntax	struct struct name { structure element 1; structure element 2; . . . . . structure element n; } struct_var_nam; struct item ms { int rno; char nm[50]; } it;	union union_nam { union element 1; union element 2; . . . . . unionelement n; } union_var_nam; union item ms { int rno; char nm[50]; } it;
⑥ eg		

Date __/__/__
Page
STUDY BUDDIES

	Local variable	Global variable
①	Those variable which declare inside the function scope	Those variable which declare inside compiler scope but not inside function scop

②	Local variable is non-shareable
	#include <stdio.h>
	void fudio ()
	{
	printf ("%d", a);
	}
	void main ()
	{
	int a=5;
	}
	error
		Global variable is shareable
		int a ;
		void main ()
		{
		}
		void hello ()
		{
		}

③	By ~~defalt~~ default ~~they~~ implemented in RAM (auto)	By implement they implemented in Rom (Static
④	By default ~~they~~ value garbage implemented value when user not assigned a value	By default value 0 (zero) when user not assigned a value.
⑤	They alive ~~till~~ end of function scope	They alive till end of compiler scope or progra
⑥	Local variable has higher priority than global variable	Global variable has lower priority than local variable

Date / /
Page
STUDY BUDDIES

* Add/sub of two matrix.

```
#include <stdio.h>
#include <conio.h>
void main()
int a[3][3], b[3][3], i, j;
printf("Enter the elements of matrix");
  for (i=0; i<=2; i++)
    {
      for (j=0; j<=2; j++)
    {
    scanf("%d", &a[i][j]);
     }
   }
for (i=0; i<=2; i++)
   {
     for (j=0; j<=2; j++)
     {
        scanf("%d", &b[i][j]);
      }
   }
for (i=0; i<=2; i++)
   {
     for (j=0; j<=2; j++)
     {
       printf("%d", a[i][j]+b[i][j]);
     }
     printf("\n");
   }
getch();
}
```

Date __/__/__
Page _____
STUDY BUDDIES

Transpose of Matrix

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int a[3][3], i, j;
printf("Enter the elements of matrix");
  for (i=0; i<=2; i++)
  {
    for (j=0; j<=2; j++)
    {
     scanf("%d", &a[i][i]);
    }
  }
for (i=0; i<=2; i++);
   {
     for (j=0; j<=2; j++)
      { printf
       printf("%d", a[j][i]);
      }
     printf("\n");
   }
 getch();
}
```

Date / /
Page
STUDY BUDDIES

* Sum of diagonal elements, main

```
#include <stdio.h>
#include <conio.h>
void main()
{
int a[3][3], i, j, k=0, l=0;
printf("Enter the element of matrix");
for (i=0 ; i<=2 ; i++)
{
for (i=0 ; j<=2 ; j++)
{
scanf("%d", &a[i][j]);
}
}
for (i=0 ; i<=2 ; i++)
{
for (j=0, j<=2 ; j++)
{
if (i==j)
{
k = k + a[i][j]
}
}
}
printf ("%d", k);
getch();
}
```

```
if (i+j==2)
{
l = l + a[i][j]
}
```

* multiplication of an matrix

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[3][3], b[3][3], i, j, k, c[3][3], sum=0;
for(i=0; i<=2; i++)
  {
   for(j=0; j<=2; j++)
   { scanf("%d", &a[i][i]); }
  }
for(i=0; j<=2; i++)
  {
   for(j=0; j<=2; j++)
   { scanf("%d", &b[i][j]); }
  }
for(i=0; i<=2; i++)
  {
   for(j=0; j<=2; j++)
    {
     for(k=0; k<=2; k++)
      {
       sum = sum + a[i][j] * b[k][j];
      }
     c[i][j] = sum;
      sum = 0;
    }
  }
getch(); for(i=0; i<=2; i++)
}
        {
         for(j=0; j<=2; j++)
         {
          printf("%d", c[i][j]);
         }
         printf("\n");
        getch();
```

Date / /
Page
STUDY BUDDIES

* Sum of any series

```
#include <stdio.h>
#include <conio.h>
void main()
{
int n, i, k = 0;
scanf("%d", &n);
for (i = 1; i <= n; i++)
  {
    k = k + i * i;
  }
printf("%d", k);
getch();
}
```

eg: use $i * i = i^2 + 2^2 + 2^3$

$i * i * i = i^3 + 2^3 + 2^3$

$(2*i-1) \; A \; (2*(i-1)) = 1^2 + 3^2 + 5^2 - -$

$[= 1 + 2 + 3 + 4.$

$1^2 + 2^3$ power(i, i) $= i^2 + 2^2 + 3^3 + 4^4 - -$

Date __/__/__
Page _____
STUDY BUDDIES

* Factorial using recursion.

```
#include <stdio.h>
#include <conio.h>
int fact(int);
void main()
{
  int n, f;
printf("enter an element to find its factorial");
  scanf("%d", &n);
  f = fact(n);
printf("factorial of %d is %d", n, f);
getch();
}
int fact(int n)
{
 if(n==0)
  { return 1; }
   else
   {
     return(n*fact(n-1));
  }
```

fibonacci series

```
#include <stdio.h>
#include <conio.h>
  int fib(int) ;
  void main ()
  {
     int n, f ;
  printf ("enter the nth number in fibonacci series");
    scanf ("%d", &n);
    f = fib (n);
  printf (" the %d number in fibonacci series is
                    %d \n", n, f)
     getch( );
  }
int fib(int n)
{
  if(n==0)
  {
   return 0;
  }
  else if (n==1)
   {
     return 1;
   }
else
  {
    return (fib(n-1) + fib(n-2));
  }
}
```

Flowchart of factorial no.