

## Assignment - I

Date / /

Page \_\_\_\_\_

STUDY BUDDIES

Ques 1. Explain the five basic operation performed by any computer system?

Ans:- All computer systems perform the following five basic operations for converting raw input into information, which is useful to their users:-

1. Inputting :- The process of entering data & instructions into the computer systems.

2. Storing :- Saving data & instructions to make them readily available for initial or additional processing, as & when required

3. Processing :- Performing arithmetic operations or logical operations on data to convert them into useful information.

4. Outputting :- The process of producing useful information or results for the users, such as a printed report or visual display.

5. Controlling :- Directing the manner & sequence in which all of the above operations are performed.

Ques 2. What is Rom? Why is it so called? Give few typical uses of Rom.

Ans:- Read Only Memory (Rom) is a type of non-volatile memory used in computers & other electronic device. Data stored in Rom can only be modified slowly, with difficult, or not at all, so it closely tied to specific hardware.

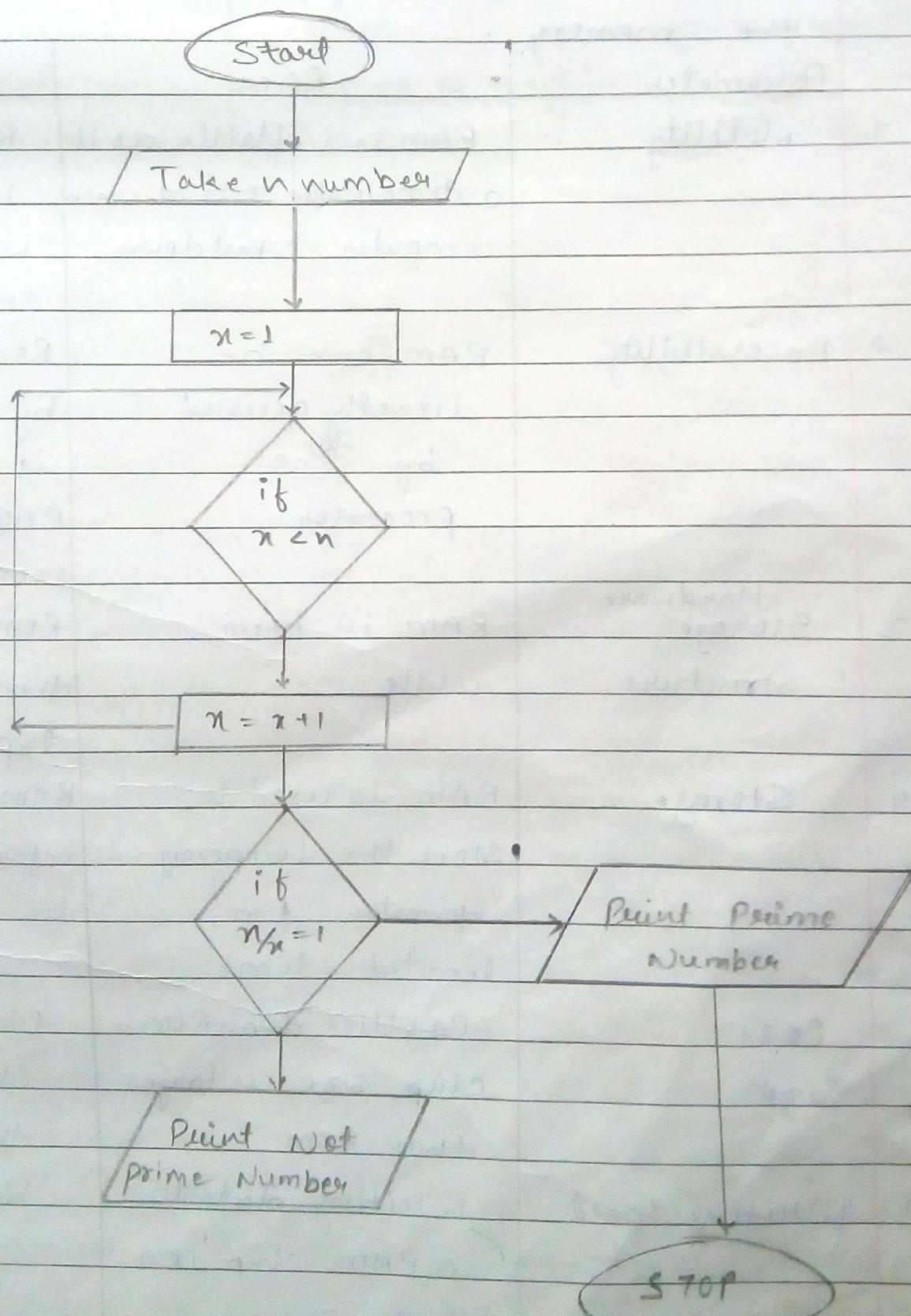
is mainly used to store firmware or application software in plug in cartridges.

Strictly, read-only memory refers that is hard wired, such as diode matrix & the later mask-ROM (MROM), which cannot be changed after manufacture. Although discrete circuits circuits can be altered in principle, integrated circuits (ICs) cannot, & are useless if the data is bad or requires an update. That such memory can never be changed is a disadvantage in many applications, as bugs & security issues cannot be fixed, & new features cannot be added.

Programmable read only memory (PROM) or One time programme Rom (OTP), can be written to or programmed via a special device called PROM Program typically, this device uses high voltage to permanently destroy or create internal links within the chip.

- Erasable programmable read only memory (EPROM) is based on a similar structure to EEPROM, but allows its entire contents to be electrically erased, then rewritten electrically so that they need not be removed from the computer
- Electrically alterate read only memory (EAROM) is type of EEPROM that can be modified one bit at a time

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.



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 accessed 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 drives made of magnetic tapes
4. Storage	RAM is used to store the temporary information for limited time	ROM is store permanent information which can't be deleted
5. Cost	Costlier than ROM	Cheaper 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.

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(ma) of data, up to 4mb or more per chip.

9. Examples

Static & Dynamic RAM

PROM, EEPROM, &

10. Content

Store Information temporarily

EEPROM are types of RAM which stores information permanently.

QUESTION

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

Ans

An Operator is a symbol which operates on a value or a variable

→ Arithmetic Operators : Arithmetic operators are used for performing arithmetic operations

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

Unary +

Unary -

& (Address extraction operator)

Return the address of variable

\* (Dereferencing operator)

Pointers to a variable

size of ()

Return the size of given datatype

( $<$  type cast  $>$ ) type cast operator

use for temporary

conversion of datatype

++ Increment

Increases value by 1

-- Decrement

Decreases value by 1

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

- (iii) Binary operators (operate upon two operands)
- + Addition Operator
  - Subtraction Operator
  - \* Multiplication Operator
  - / Division Operator
  - % Modulus Operator

→ Relational Operators :-

Relational op's 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
and	logical AND. True only if all operands are true	$((c==5) \& (d>5))$
	logical OR. True only if either one operand is true	$((c==5)    (d>5))$
!	logical NOT. True only if the operand is 0	$! (c==5)$

## → Bitwise Operators :-

During computation, mathematical operations like: addition, subtraction, multiplication & 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 arithmetic operators (\*, /, %, +, -) is higher than relational operators (==, !=, >, <, >=, <=) & precedence of relational operator is higher than logical operators (&&, || & !).

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

$(1 > (2+3)) \& \& 4$  return  $\rightarrow 0$

## ⇒ Associativity of operators

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

e.g.  $1 == 2 != 3$

$0 != 3$

return  $\rightarrow 1$

Ques-2

What is recursion? Write a recursive function to calculate sum of digits of 5 digit number entered through the keyboard.

Ans

Recursion occurs when a thing is defined in terms of itself or of its type. Recursion is used in a variety of disciplines ranging from linguistics to computer science.

Ans

The process in which a function calls itself directly or indirectly is called recursion & the corresponding function is called recursive program.

The idea is represent a problem in term of one or more smaller problem add one or more base condition that stop recursion.

for example we compute factorial n if we know factorial of (n-1) Base case of factorial would be n=1 we return 1 when  $n=0$ .

If base case is not reached or not defined then stack overflow problem may arise.

A function called direct recursive if it calls the same function. A fun is called indirect recursive if it call another fun "fun-new" fun-new & fun-old fun directly or indirectly.

A recursive function call tail recursive if recursive caller is the last thing executed by the function.

```
# include < stdio.h >
# include < conio.h >
void main ()
{
    int num, sum ;
    printf ("Enter any number");
    scanf (" %d ", & num)
```

Ans :- A function is a group of statement that together perform a task. Every C program has at least one function, which is main(), & all the most trivial program can define additional fun<sup>n</sup>

Page \_\_\_\_\_  
STUDY BUDDIES

```
sum = sum of digit (num)
printf (" Sum of digit of %d = %d ", num, sum);
return (0);
}
int sum of digit (int sum)
{
    if n == 0
        return 0;
    return (num % 10) + sum of digits (num/10);
}
```

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 argument declaration & the body of the function).

The first line of a function definition contains the type specification of the value returned by the function followed by the function name & set of argument separated by commas & enclosed in parenthesis each arguments is preceded by its associated type declaration.

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 there are other numerous library function defined under < stdio.h> such as `scanf()`, `printf()`, `getch()`

Advantage :- have gone through

hence

- ① These function<sup>1</sup> is use multiple rigorous testing & hence 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

< ctype.h> = character type function.

< locale.h> = localization function

< math.h> = mathematics function

< setjmp.h> = Jump function

User defined function :- As mentioned earlier, C allows 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.

When compiler encounters function name () inside the main function control of the program jumps to there and the compiler starts executing the code inside the user defined function. The control of the program jumps to statement next to function name one all code inside the function definition.

Advantage:-

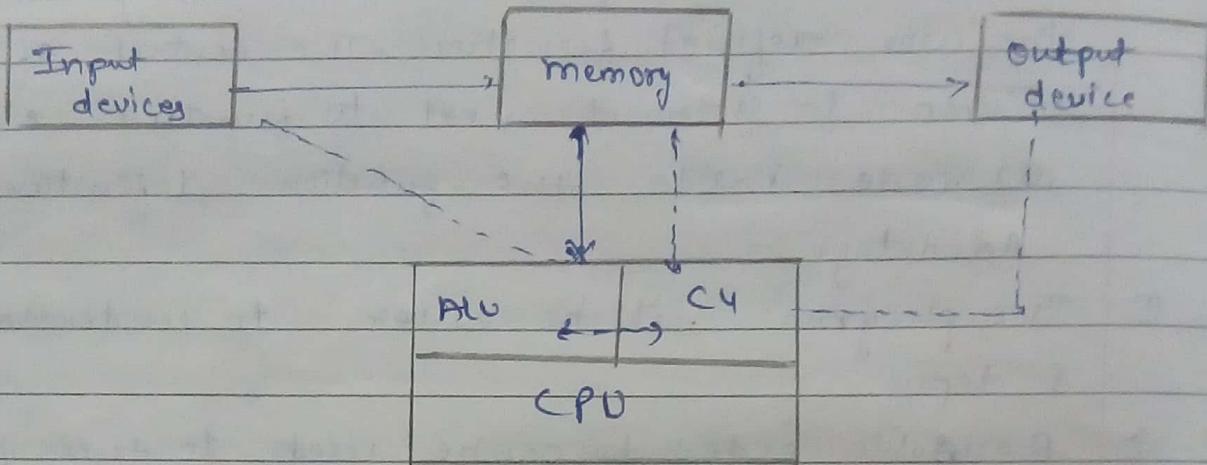
- ① The program will be easier to understand, maintain & debug.
- ② Reusable code that can be used to their program.
- ③ A large program can be divided into smaller modules hence a large project can be divided among many programs.

Parameter passing method:-

1. Pass by result:- This method uses out mode semantic. Just before control is transferred back to the caller the value of the formal parameter is transmitted back to actual parameter.
2. Pass by value result:- This method uses in/out mode semantic. Just before control is transferred back to the caller the value of the formal parameter is transmitted back to actual parameter.
3. Pass by name :- The technique is used in programming language such as Algol. In this technique symbolic name of variable passed which allows in both.

Ques 8.9.

What is flow of the computer system & explain briefly.



The computer has few 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 useable by the computer whatever input is supplied by the input device first goes to the memory - The CPU acts as brain CPU is responsible for the overall working of all component of the computer It consists of two parts : Arithmetic , logic unit & control unit . The ALU performs arithmetic 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

is started 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 faster in speed less 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:-

(i)

### Compiler

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'

2. The compiler takes as input the preprocessed code generated by preprocessor

### Assembler

The assembler takes as a input the assembly code generated by the compiler & translate into relocatable machine code.

It take assembly code as input.

3. The compilation takes place in two phases analysis & synthesis phase input gas analyzer logical syntax analyzer, semantic analyzer takes place in ~~inter~~ via intermediate code, generator code optimizes code generator
4. The assembly code generated by the compiler is a mnemonic version of machine code

The phase detects the instruction identify & allot address to them in the second phase the assembly code transmitted to binary code.

Relocatable machine code generated by assembler as binary relocatable code.

X<sup>xx</sup>

### Hardware

- Physical parts of the computer called hardware
- We can touch, feel & see hardware
- Hardware is constructed using physical material & component.
- Computer is the hardware which operates under the control of a software
- If hardware is ~~damaged~~ not affected by computer virus it is replaced by new one
- Hardware is not transferred from one place to another electrically through network

### Software

A set of instruction given to the computer is called software

We can't touch & feel software

Software is developed by writing instruction in program language.

The operation of computer controlled through the software.

If software is corrupted its back up copy is reinstalled

It is transferred from one place to another place electrically through network

7. Hardware is not affected by computer virus
8. User cannot make new duplicate copies of hardware

It is affected by the computer virus.

User can make many new copies of the software

### ~~System Software~~

- ① System software used for operating computer hardware
- ② System software installed on the computer when Operating System is installed
- ③ In general the user does not interact with system software because of its background.
- ④ System software runs independently it provides platform from remaining application software
- ⑤ Compiler, assembler, debugger

### ~~Application Software~~

Application software is used by user to perform specific task.

Application software is installed according to user's requir.

In general user interact with the application

Application software can't independently they can't without the presence of system software.  
word processor, Web browser  
media player.

### ~~Compiler~~

Compiler first scans the whole program for error. If no error, the compiler will convert source code to machine code called object code

### ~~Interpreter~~

Interpreter reads each line of source code statement line by line into machine code & also gets it executed

- |  |   |
|--|---|
| <ul style="list-style-type: none"><li>④ Pro Program runs fast : because the translated version already available &amp; can run directly</li><li>⑤ No error can escape the eye of the compiler</li></ul>              | <p>Program is easy, because it stops translation when the first error is met.</p> <p>Certain erroneous statement may escape detection, becaz the interpreter checks only those statement for errors that it executes.</p> |
| <ul style="list-style-type: none"><li>⑥ Compiler based programs occupy <del>less</del> <sup>less</sup> space on disk</li><li>⑦ Programming language like PASCAL, FORTAN, COBOL, <u>C, C++</u>, <u>JAVA</u></li></ul> | <p>Interpreter based program <del>more</del> <sup>less</sup> space on disk.</p> <p>Programming language like <u>B, BASIC, FOXPRO, Python</u>.</p>   |

*Final 10/10  
28/11/18*

# Assignment - 2

Date / /

Page \_\_\_\_\_

**STUDY BUDDIES**

**Ques.** Write down the difference b/w mini, micro & mainframe computer.

**Ans**

**Micro computer**

**Mini computer**

**Mainframe computer**

① It was introduced in 1970	It was introduced in 1960	It was introduced in 1975
② It generally consists of one processor	It generally consists of two or more processor	It generally consists of multiple processors
③ Its storage capacity is in GB	Its storage capacity is in TB	Its capacity is in PB
④ Its memory is in MB	Its memory is in GB	Its memory is in GB
⑤ It can normally be used by 1 user at a time	It can serve two to thousands connected user at a time	It can serve hundred to thousands connected users at a time
⑥ Its size is very small	Its size is bigger than microcomputer	Its size is bigger than micro & mini computer
⑦ Its price \$500 to \$ 5000	Its price is \$ 18000 to \$50000	Its price is \$50000 to \$million
⑧ It is used at home, offices & educational institutes	It is used by business institutes & departments	It is used by large organizations like banks
⑨ Examples are IBM-PC & Apple Macintosh	Examples are VAX-8800 & MV-1500	Examples are IBM-370 & NEC

**Ques.**

Explain the concept of scope & lifetime of a variable with their storage class inc.

**Ans.**

The scope of a variable is local to the function in which it is defined

#include < stdio.h >

#include < conio.h >

```
void display (int) ;  
void main()  
{ int i=20 ; display(i) ; return 0 ; }  
void display (int j)  
{ int k=35  
    printf ("%d\n",j) ; printf ("%d\n",k) ; }
```

The presence of a *i* is known only to function *main()* & not to any other function. Similarly the variable *k* is local to the function *display()* & hence it is not available to *main()*; that is why to make the value of *i* available to *display()* we have explicitly pass it to *display*. C language uses 4 class storage:-

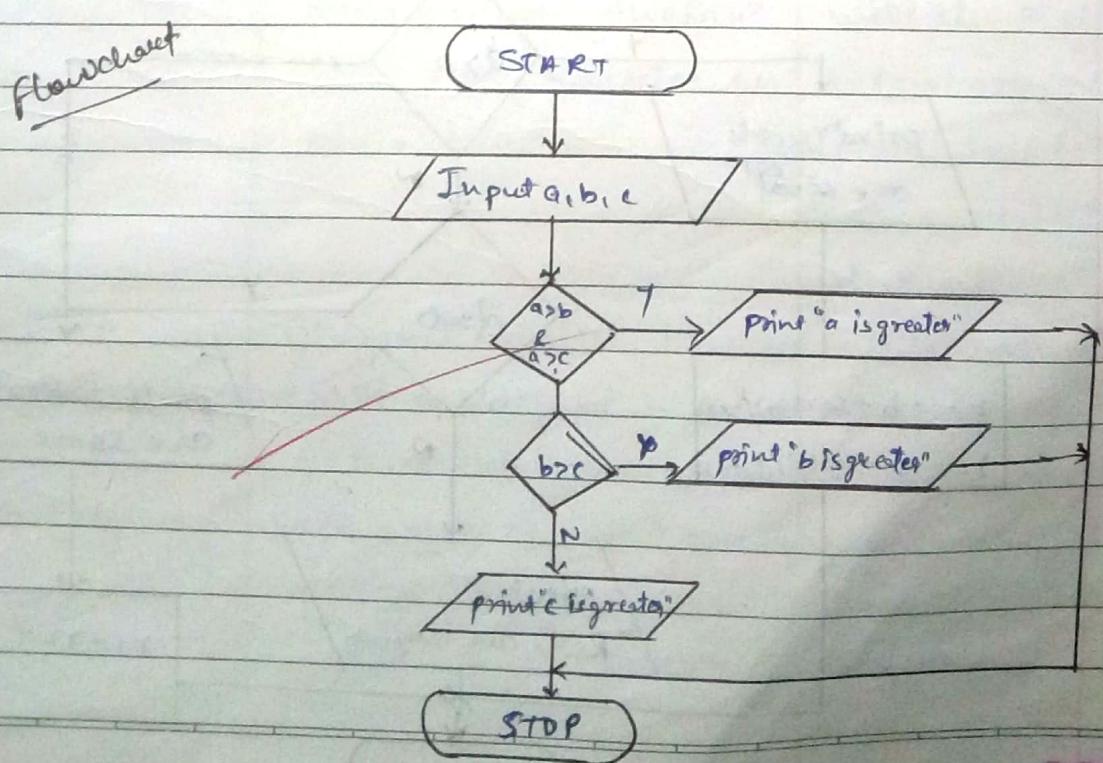
1. Auto :- This is default storage class for all variable declared inside a function or a block. Hence keyword is rarely used hence using a *Auto* variable rarely used can be accessed within the block / function they have been declared & not outside them (which define their scope) of course these can be accessed within nested block within the parent block / function in which the auto variable was declared in which parent block declared they are assigned garbage value by default.
2. Extern :- Extern storage class simply tells us that the variable is defined elsewhere & not within the same block where it is used. Basically the value is assigned to it in a different block & overwritten / changed in a different block as well as so an extern variable nothing but a global variable initialize with a legal

value where it is used elsewhere it can be accessed within any function.

3. Static :- This storage class is used to declare static variable which are properly used while writing C program. Static variable have a property of preserving their value even after they are out of their scope. Hence static variable preserve the value of their last use in their scope. We can say that they are initialized once & exist till the terminate of the program.

4. Register :- Register is base same functionality as the auto variable the only difference the compiler tries to store these variable in the register of microprocessor if free register is available their make use of register variable much be faster than variable store in memory.

Ques 3 Write down the algorithm<sup>& flowchart</sup> to find the largest of three numbers?



Algorithm:-

Step 1 - Start

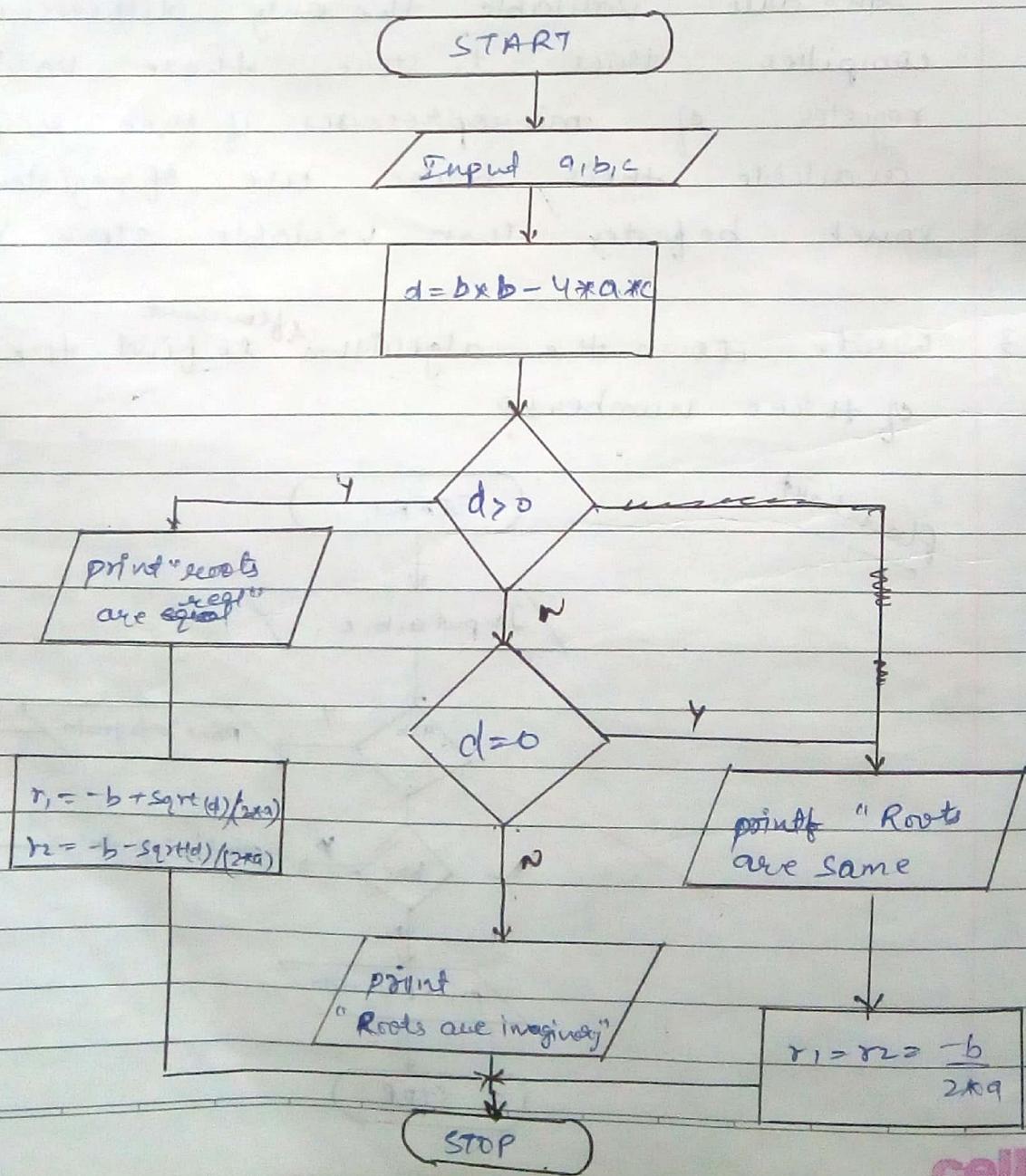
Step 2 - Declare three variable a, b, c

Step 3 - If  $a > b \& a > c$  then print a, is greater  
else b > c then print b, is greater  
else print c greater.

Step 4 - Stop

Ques 9. Draw flowchart for finding the roots of a quadratic eqn.

Ans.



#7

~~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 programming 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 func. func.
Arguments	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 func are not reflected on other functions	Any changes made in the formal arguments will also reflect in the actual arguments. Changes made inside the func are reflected outside the functions as well.
Value modification	<del>Original value is not modified</del>	Original value is modified
Safety	Actual arguments remain safe, they cannot be modified accidentally	Actual arguments are not safe. They can accidentally modified

Ques. 6.

Write a program to sort an array in ascending order.

Ans

```
#include < stdio.h >
```

```
#include < conio.h >
```

```
void main()
```

{

```
int arr[10], i, j, temp; clrscr();
```

```
printf ("Enter elements of array : ");
```

```
for (i=0; i<=9; i++)
```

```
scanf ("%d", &arr[i]);
```

```
for (i=0; i<=9; i++)
```

```
    for (j=0; j <= 9-i; j++)
```

```
        if (arr[j] > arr[j+1])
```

```
            temp = arr[j];
```

```
            arr[j] = arr[j+1];
```

```
            arr[j+1] = temp; }
```

```
printf ("\n sorted array is in ascending order : \n");
```

```
for (i=0; i<=9; i++)
```

```
    printf ("%d", arr[i]);
```

```
getch();
```

}



~~Ques. 7~~ Explain the "switch" case with suitable example.

In computer programming languages, a switch statement is a type of selection control mechanism used to allow the value of a variable or expression to change the control flow of program execution via search & map.

Switch statements function somewhat similarly to the `if` used in programming language like C/C++, C# .NET, Visual Basic.

~~switch~~ statements come in two main variants : a structured switch, as in Pascal, which takes exactly one branch, & an unstructured switch, as in C, which function as a type of goto. The main reasons for using a switch include improving clarity, by reducing otherwise repetitive coding, & also offering the potential for faster execution through easier compiler optimization in many cases.

Ques 8 Add & Subtract the following numbers

$$(11000)_2 - (10101)_2$$

~~Ques 8.~~ What is volatile volatile memory? Explain different types of volatile memory.

Ans. Volatile memory is a contrast to non-volatile memory, is computer memory that requires power to maintain the stored information, it retains its contents while powered on but when the power is interrupted, the stored data is quickly lost.

Volatile memory has several uses including as primary storage. In addition to usually being faster than forms of mass storage such as a hard disk drive, volatility can protect sensitive information, as it becomes unavailable on power-down. Most of the general-purpose random-access memory (RAM) is volatile.

1. Dynamic :- Dynamic RAM (DRAM) is very popular due to its effectiveness. DRAM stores each bit of information in a different capacitor within

The integrated circuit DRAM chips need just one single capacitor & one transistor to store each bit of information - This makes it space efficient & inexpensive.

2. Static :- The main advantage of static RAM (SRAM) is that it is much faster than dynamic RAM. Its disadvantage is its high price. SRAM does not need continuous electrical refreshes, but it still requires constant current to sustain the difference in voltage. Every single bit in a static RAM chip needs a cell of six transistors, whereas dynamic RAM requires only one capacitor & one transistor, whereas dynamic RAM requires only one capacitor & one transistor.

Ques 8. Add & subtract :-

$$(9) \quad (1110001)_2 - (1010101)_2$$

$$(64 + 32 + 16 + 0 + 0 + 0 + 1) - (64 + 16 + 0 + 4 + 0 + 1)$$

$$(113) - (85)$$

$$28 \Rightarrow (11.1\ 0\ 0)_2$$

2	28	
2	14	→ 0
2	7	→ 0
2	3	→ 1
2	1	→ 1
2	0	→ 0
6	—	000

$$\begin{array}{c} \text{B}^6 \\ \diagdown \quad \diagup \\ 5 \quad 1 \quad x \\ \diagup \quad \diagdown \\ 5 \quad 3 \\ \diagup \quad \diagdown \\ 5 \quad 0 \end{array} \quad \text{(b)} \quad (14_2)_5 + (12_4)_5$$

$$(1 \times 5^2 + 4 \times 5^1 + 2 \times 5^0) + (1 \times 5^2 + 2 \times 5^1 + 4 \times 5^0)$$

$$(25 + 20 + 2) + (25 + 10 + 4)$$

$$(25 + 20 + 2) + (25 + 10 + 4)$$

(32)  $\zeta$

$$\begin{array}{c|cc} \hline 5 & 86 \\ \hline 5 & 17 & \rightarrow 1 \\ 5 & 3 & \rightarrow 2 \\ \hline & 0 & \rightarrow 3 \end{array}$$

Ques. Convert the following Numbers in specified No.

$$(a) (69)_{10} \Rightarrow (x)_5 , n=?$$

$$(234)_5$$

$$\begin{array}{r} 5 | 69 \\ 5 | 13 \rightarrow 4 \\ 5 | 2 \rightarrow 3 \\ 0 \rightarrow 2 \end{array}$$

$$(b) (\text{8E})_{16} = (x)_2 , n=?$$

$$1000 \quad 110$$

$$(1000110)_2$$

$$\begin{array}{r} 15 \\ 100 \\ 3421 \end{array}$$

$$\begin{array}{r} 1110001 \\ \hline 2^0 \quad 2^1 \quad 2^2 \quad 2^3 \quad 2^4 \quad 2^5 \quad 2^6 \\ \downarrow \qquad \qquad \qquad \qquad \qquad \qquad \qquad \end{array}$$

$$2^0 + 2^2 + 2^4 + 2^6$$

$$64 + 16 + 4 = 84$$

$$16 + 32 + 64 = 96$$

$$= 29$$

$$\begin{array}{r} 2 | 28 \rightarrow 0 \\ 2 | 14 \rightarrow 0 \\ 2 | 7 \rightarrow 1 \\ 2 | 3 \rightarrow 1 \\ 2 | 1 \rightarrow 1 \\ 2 | 0 \rightarrow 0 \end{array}$$

$$110 - 84$$

$$\begin{array}{r} 2 | 28 \quad 0 \\ 2 | 14 \quad 0 \\ 2 | 7 \quad 1 \\ 2 | 3 \quad 1 \\ \hline & & 1 \end{array}$$

$$(11100)_2$$

Binary to Decimal

$$\begin{array}{r} 011100 \\ \hline 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0 \end{array} \Rightarrow 2^0 \times 0 + 2^1 \times 0 + 2^2 \times 1 + 2^3 \times 1 + 2^4 \times 1 + 2^5 \times 0 = 28$$

# Assignment No. - 3

Date \_\_\_\_\_  
Page \_\_\_\_\_  
**STUDY BUDDIES**

Ques

Why we goto statement should be avoided in programming & Differentiate b/w for, while do while loop.

Ans

C has goto statement but one must ensure not to use too much of goto statement in their program because its functionality in their program because is limited & it is only recommended as a last resort if structural sol'n are much more complicated. First let us see what the goto statement does, its syntax & functionality.

for loop

1.

Syntax :

for (initialization;  
Condition; updating),  
{ statements ; }

2.

It is known as  
entry controlled loop

3.

If the condition  
is not true first  
time that control  
will never  
enter in a loop

4.

There is no  
semicolon(;) after  
the condition  
in a syntax of the  
for loop

5.

Initialization & updating  
is part of the  
syntax

while loop

Syntax ;

while (condition),  
{ statement ; }

It is known as  
entry controlled loop

If the condition is  
not true first  
time then control  
will never enter  
in a loop.

There is no semicolon  
after the condition  
in the syntax of  
the while loop

Initialization & updating is  
not a part of the  
syntax

do while loop

Syntax

do { Statement ; }  
while (condition);

It is known as  
exit controlled loop

Even if the cond  
is not true for the  
first time the  
control will  
enter in a loop.

There is a  
semicolon(;) before  
the condition in  
the syntax of the  
do while loop.

Initialization & updating  
is not a part of the  
syntax

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:-

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

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

`Char label1 = "single"`

`Char label2 = "Married"`

`Char * label ptr`

`label ptr = label`

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

label @ 2000  
 lsl; lnl give lvi  
 label 2 @ 3000  
 M1 a label lvi  
 label ptr @ 4000  
 1 20001

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 datatype 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;

3

A structure variable can either be declared with structure declaration or as a separate declaration like basic datatype.

// A variable declaration with str.

struct point

{ int x, y;

3 p1; // The variable P3 is declared with point

// A variable declaration like data type

struct point

{ int x, y; }

int main

{ Struct point p1; // The variable 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 initialize member  
int y=0; // compiler error ; cannot initialize 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 initialized 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.

*Bird  
15/2/18.*

Ques What are different data type in C language. Explain each with 4 examples along with their memory allotment.

Ans Two types of data type

① Primitive datatype :-

integer :- Keyword used for integer data type is int integers typically require 4 byte of memory space & range from -2147489648 to 2147489647

character :- character datatype require for storing character char is used for 1 byte memory space & ranges from -128 to 127.

Boolean :- Boolean data type used for storing boolean & logical value A Boolean variable can store either true or false is used as a Bool.

float :- float point data type is used for storing single precision floating point value & decimal value keyword used for data type is float it require 4 byte of memory space.

Double floating point :- Double floating point data type is used for storing double precision floating point of value or decimal value . keyword is double it require 8 byte of memory space.

void :- void means without any value void data type used as valueless entity.

② Abstract or user define datatype :- These data type are defined by user itself like, defining classes in C.

Ques What is self referential structure? Explain with example.

Ans struct tag

{ member 1;

member 2;

.....

struct tag \* name; } ;

where name refers the name of a pointer variable thus the structure of type tag will contain a member that point to another structure of type tag . Such structures are known as self referential structures.

Ans

Self referential structures are those structures that have one or more pointers which point to the same type of structure , as their member.

Example

struct mode {

int data 1;

char data 1;

char data 2;

struct node\* link;

};

int main ()

{ struct node ob;

};

}

In the example 'link' is a pointer to a structure of type 'node'. Hence, the structure 'node' is a self-referential structure with 'link' as the reference pointer.

An important point to consider is that the pointer should be initialized properly before accessing, as by default it contains garbage value.

Two types referential structure:-

1. Self referential structure with single link
2. Self referential structure with multiple links

~~Ques~~

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=%lf", sum);
    return 0;
}
```

Enter a n<sub>1</sub> : 2.4  
Enter a n<sub>2</sub> : 4.5  
Enter a n<sub>3</sub> : 3.4  
Enter a n<sub>4</sub> : -3  
Sum = 10.30

→ 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 no. d: ", i);
        scanf("%lf", &number);
        if(number < 0)
            continue;
        sum += number
    }
    printf("sum = %.2lf", sum);
    return 0;
}
```

Output : 1.1

Enter a n<sub>1</sub> : 2.2

Enter a n<sub>2</sub> : 3.3

Enter a n<sub>3</sub> : 4.4

Enter a n<sub>4</sub> : -3.4

Enter a n<sub>5</sub> : -45.5

Enter a n<sub>6</sub> : 34.5

Enter a n<sub>7</sub> : -4.2

Enter a n<sub>8</sub> : -1000

Enter a n<sub>9</sub> : 12

Enter a n<sub>10</sub> :

Enter a n<sub>11</sub> sum = 59.70

Ques Write advantage & disadvantage of pointer in C

Ans :- Advantages :-

- pointers provide direct access to memory.
- pointers provide a way to return more than one value.
- Reduces the storage space & complexity of the program to the fun<sup>n</sup>.
- Reduces the execution time of the program.
- Provides an alternate way to access array elements.
- pointers can be used to pass information back & forth b/w the calling fun<sup>n</sup> & called fun<sup>n</sup>.
- pointers allows us to resize the dynamically allocated memory block.
- Address of objects can be extracted using pointers.

### Disadvantages :-

- Uninitialized pointers might cause segmentation fault.
- Dynamically allocated block needs to be freed explicitly otherwise, it would lead to memory leak.
- Pointers are slower than normal variables.
- If pointers are updated with incorrect values, it might lead to memory corruption.

~~Ques~~

What do you mean by dynamic memory allocation?

Define calloc(), malloc(), realloc() & free() function.

~~Aus~~

Dynamic memory allocation refers to managing system memory at run time. Dynamic memory management in programming language is performed via a group four function namely malloc(), calloc(), realloc() & 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 heap 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 ; sizeoflimit);`

**realloc()** :- ~~realloc()~~ is used when an allocated block of dynamic memory needs to be resized. If necessary, realloc() completely ~~reallocates~~ reclaims the memory block. Assume you make an initial memory allocation using malloc() as follows:-  
`int * buffer = (int *) malloc (size);`

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 * size of (int))`;

free() :- Dynamically allocated memory created with either calloc() or malloc() doesn't get freed by itself. You must explicitly use free() to release the space.

Syntax : `free(ptr);`

~~Stand  
02/01/19~~

## STRUCTURE

### ① Access Members

We can access all the members of structure at anytime.

### ② Memory allocation

Memory is allocated for all variables.

### ③ Initialization

All members of structures can be initialized.

### ④ Keyword

'Struct' keyword is used to declare structure.

### ⑤ Syntax

```
struct struct name  
{
```

    structure element 1 ;

    structure element 2 ;

    structure element n ;

```
} struct - var - nam ;
```

struct item mst

{

    int rno ;

    char nm[50] ;

```
} it ;
```

## UNION

Only one member of union can be accessed at anytime.

Allocates memory for variable which variable require more memory.

only the first member of a union can be initialized.

'Union' keyword is used to declare union.

union union name

{

    union element 1 ;

    union element 2 ;

    union element n ;

```
} union - var - nam ;
```

union item mst

{

    int rno ;

    char nm[50] ;

```
} it ;
```

### local variable

① Those variable which declare inside the function scope

② Local variable is non-shareable

```
#include <stdio.h>
```

```
void func()
```

```
{
```

```
printf ("%d", a);
```

```
}
```

```
void main()
```

```
{
```

```
int a=5;
```

```
error
```

③ By default they implemented in RAM (auto)

④ By default they implemented value when user not assigned a value

⑤ They alive till end of function scope

⑥ Local variable has higher priority than global variable

### Global Variable

Those variable which declare inside compiler scope but not inside function scope

Global variable is shareable

```
int a;
```

```
void main() void hello()
```

```
{ }
```

```
{ }
```

```
}
```

③ By implement they implemented in Rom (static)

④ By default value 0 (zero) when user not assigned a value.

They alive till end of compiler scope or program

Global variable has lower priority than local variable