B.Tech.

Second Semester Examination, May-2011

Fundamental of Computer and Programming in C (CSE-101-F)

Note : Attempt five questions in all. Question No. 1 is compulsory.

Q. 1. Define the following terms :

(a)	Micro processor	(b)	Operating system
(c)	Compiler	(d)	Graphical user interface
(e)	DNS	(f)	Data type
(g)	Array	(h)	Pointer
(i)	Operand	(j)	Computer network.

Ans. (a) Micro Processor : A microprocesser incorporates the functions of a computer's central processing unit on a single integrated circuit. It is a multipurpose programmable, clock driven, register-based electronic device that accept binary data as input. Process it according to instructions. Stored in its memory, and provides results as output.

During 1960s, computer processors were often constructed out of small and medium scale ICs containing form tens to a few hundred transistors. The integration of a whole CPU onto a single chip greatly reduced cost of processing power.

(b) Operating System : An operating system is a software which is collection of program that provide an interface between user and computer. An operating system creates an environment in which a user interact with hardware or computer hardware.
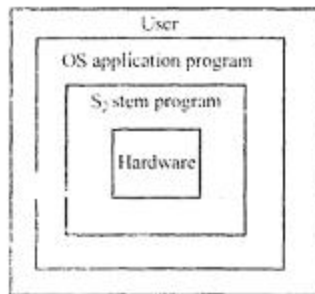

Block Diagram for an O.S.

On operating system is marking like a resume manager resource such as memory management, process management, Device management, file management etc. Operating system also provide various services such as program enation, program execution and finally program termination.

(c) Compiler : A compiler is a computer program (or set of programs) that translate the text written in a computer language or (source languages) into another computer language (target language).

The term compiler used for program that translate some code written in high level language into a lower

level language (e.g., assembly-language or machine language).
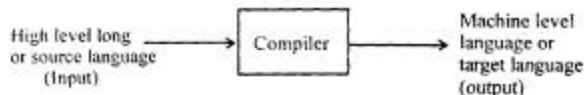
A compiler perform many operation such as lexical analysis, semantic analysis, parsing code generation, ad code optimization.

(d) Graphical User Interface (GUI) : It is common for software packages to provide graphical user interface. A major component of a GUI is a window manager that allows a user to display multiple-windows areas each window can contain graphical or non-graphical displays. To make a particular window active we simply click in that window using an interacting pointing device.

Interface also display menus and items for fast selection of processing options. The advantage of icon are that they take up less screen space and they can be understood more quickly if well designed.

(e) DNS (Domain Name System) : It is hierarchical domain based naming scheme and a distributed data bore system for implementing this naming scheme. It is applied in mapping host names and email destinations to IP addresses with various purpose. DNS is define in RFCs 1034 and 1035.

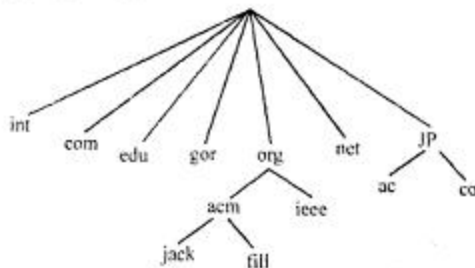IP addresses are mapped by library procedure called resolver.

Figure Shows Internet DNS

Domain → Description

. int → International Organization

.com → Commercial Application

.edu → Educational Institute

.gov → U.S. Governmental Organisations.

.net → Network etc.

(f) Data Type : A data type is a finite set of values along with a set of rules for allowed operation. Data type in 'C' are following type :
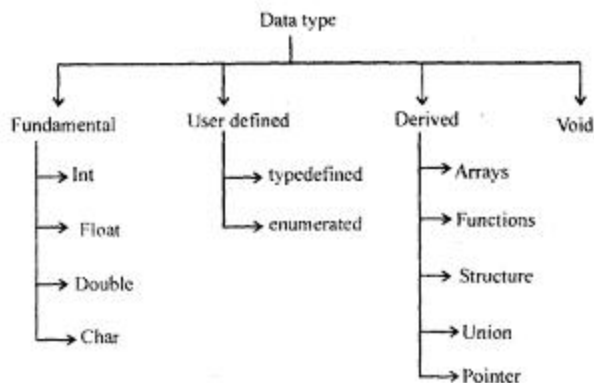
1. **Fundamental Data Types :** One those data type which one used for actual data representation in memory of computer such as int for integer, char for character or string.

2. **User Defined Data Type :** Which is define by user such as enum.

Example :   enum identifier (var 1, var 2............... var)

enum month (Jan. Feb .... Dec.)

3. **Derived Data Type :** Which one derived from fundamental D Types.

(g) **Array :** In General array can be define as collection of similar data type (int, char, float, etc.)

For Example : On array of student, on array of elements.

Int   a   [10]; or int a [10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
↓   ↓   ↓

Type of   Name of   Size of
array   array   array

General formate for dexterity one-D array data-type name of [size of array]

	a [0]	a [1]	a [2]	a [3]	a [4]	a [5]	a [6]	a [7]	a [8]	a [9]
int a [10] =	1	2	3	4	5	6	7	8	9	10

Storing Data in Array :

State int a [0] = 10;

10 store at first position is array named a.

Declaration of 2-D Array :

int a   [i]   [j];
         ↑   ↑
        row   column

(h) **Pointer :** In reference to 'C' language, poiiter is a special index variable which hold address of other variable.

Declaration of a Pointer :

Data type * pointer_name

int   *   a;

   ↑   ↑

  Data type   Name of pointer

You have defined a pointer variable (a) and one ready to hold an address into it. For this, 'C' language provide on operations or address operator or it is called unary operator. Address operated denoted by and sign.

Pointer variable = & variable_name

$\boxed{a = \&b}$.

Here address of variable b is stand in a, a is a pointer variable.

Printing the Contents of a Pointer :

printf ("%d", a)

(i) Operand : Operands one define as collection of constant or variable on which operator one used to calculate or to get a designed result.

To understand concept of operand we consider following example :

$\boxed{c = a + b};$

Here, a, b, c one variable called operand + is a Arithmetic plus operator which add value of a and b and then finally stored in C.

(j) Computer Network : Computer networks is define as collections of computing device which form a network in which data or thoughts or information can be exchanged or you can say communication is take place.
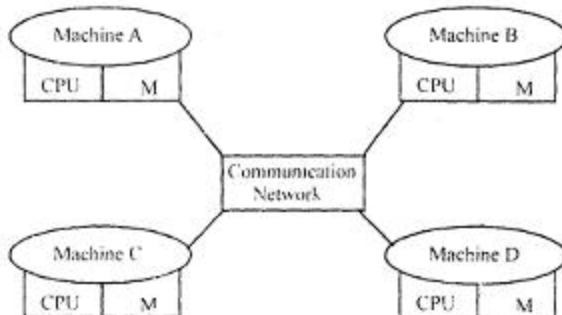
The computer system may communicate over a computer network and share their resources. There networks enable the computer to communicate with a remote computer which may be located at a distance of few meters to several thousand kilometers. Compute network connecting several peripheral device and storage & computing device shown in above figure.

Q. 2. (a) What are the different components of a computer? Explain each of them.

Ans. Different components of computer : A computer is an electronical device or hardware device in

which I/P is supply, processing them and finally output is generating.

A computer is consists of various component which ion shown in given below :
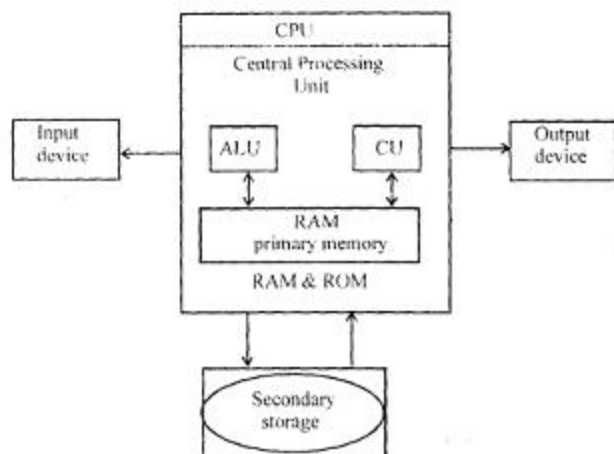
A computer is consists of following units :

(i)   Input units

(ii)   Output units

(iii)   Central processing units

    (a)   AL Unit

    (b)   Control Unit

    (c)   Memory Units

(i) **Input Units :** Inputs units are used to provide means for supplying data or instruction to the computer. In other words. What ever you supply to the computer for processing is supplied through input units.

If you are going to calculate simple Interest than you supplied principal, rates, time there values are providing by some devices called input units. Such as keyboard, mouse, scene, touch screen, trackball etc.

(ii) **Output Units :** The rate of output units is to show the result of processing or you can say that computer display all the result on its outputs units.

For example, if you one calculating simple interest on the computer, it will display interest amount on its outputs. Units called monitor. Other output devices which one used in computer one :

(i) **VDU (Visual Display Units) :** To display the contents that one being typed through keyboard.

(ii) **Printer :** Printer is an output device which is used to print the output on the paper. There are various type of printer such as lower printer.

(iii) **Speakers :** Dot matrix printer, Inkjet printers. Speaker D produced audio outputs with the help of speaker you can hear song, speak, tune etc.

(iii) **CPU :** (Central Processing Units). The task of performing the operation like arithmetic and logical

operation is called processing. CPU take data from input device, processing them and generate result to output device.

Central processing unit is consisting of :

(a) Arithmetic logical units

(b) Control Unit

(c) Primary memory (RAM & ROM)

(a) **Arithmetic Logical Units** : Basically it perform all arithmetic and logical operation. After you enter data through input device, it is stand in primary data storage. Major operation performing by ALU one addition, substraction, divisions and multiplication, logic and comparison. When ever data is required it is transferred from primary storage. After processing, data again stored in primary storage for future purpose.

(b) **Control Unit** : Control unit is just like a brain of a computer systems. CU acts like the supervisor seeing that things one done in proper fashion. The control units determine the sequence in which computer program and instruction are executed.

It is also acts as a switchboard operator when several users access the computer simultaneously. Therefore it is the manager of all the operation mention above.

(c) **Primary Memory** : Primary memory is directly accessible by processing unit. The data which is store in primary memory is lost if power is switch off. Example of primary memory is RAM. (Random Access Memory).

Q. 2. (b) What is meant by an input device? Describe different categories of input devices.

Ans. Input Device : Input devices and used to provide means for supplying data or instruction to the computer. Basically data is raw facts before processing so we need to processing this data or raw facts to get a desired result or information.

There are following input devices are :

(i)	Keyboard	(ii)	CD-ROM drive
(iii)	Mouse	(iv)	Light pen
(v)	Touch screen	(vi)	Joystick
(vii)	Bar code reader	(viii)	Scanner
(ix)	OCR	(x)	MICR.

(i) **Keyboard** : Keyboards consists of many keys. Keys have alphabets, numbers, characters or special character and words written on them. When ever keyboard is connected to the computer a blinking sign (1 or –) is appearing on the monitor. This sign is called cursor. When you press an alphabets or any number key the character-written on the key gets typed at current cursor position and cursor moves forward.

Keys :

(a) **Alphabet Keys :** Alphabet keys have alphabet from A to Z, written on them. Generally, alphabet are print in lower core.

(b) **Number Keys :** Number keys have digits from 0 to 9 written on them.

(c) **Special Character Key :** Few keys of the keyboard have special character such as semicolon (;), comma (,), dot (.) etc.

(ii) **CD-ROM Drive :** CD stand for compact Disk it is used for storing the data or instruction for future or further used. It looks likes circular plastic disk. Huge amount of data can be stand on CD.

For Example : Single CD can hold personal data of all the student studying in a college. CD ROM is an input device that data from CD. CD-ROM stands for compact Disc Read Only memory can only read the data

from CD.

(iii) **Mouse :** Mouse is an input device. It looks like a real mouse with buttons placed on its back. When ever a mouse is connected to the computer an arrow appears on the monitor screen. This sign called mouse pointer.

(iv) **Light Pen :** Light pen looks like an ordinary pen and performs same function in computer world. You hold the pen and write on a special pad or directly on the screen.

(v) **Touch Screen :** Touch screens are special scenes, which one touch sensitive. When you touch a part of the screen, they are able to generate information that the screen has been touched and that too on this.

(vi) **Joystick :** Particular location. Joystick consists of a lever placed vertically in the middle of a small board. The lever can be moved in any desired direction.

(vii) **Bar Code Reader (BCR) :** Bar code reader is an input device. It is used for reading bar codes that appear on the wrapper of the objects like too the parts.

(viii) **Scanner :** Scanner is a device that scane the pictures, photos, documents etc. Scanner convert the electronic pulse into computer readable form.

(ix) **OCR (Optical Character Reader) :** If document is scanned and inputed into computer then using OCR, it can be converted into documents file.

(x) **MICR : (Megnetic Ink Character Reader) :** MICR is an input device which reads special character, written, using magnetic ink.

Q. 3. (a) Explain functions of operating system in detail.

Ans. Function of Operating System : An operating system performing following function :

(i)   Booting
(ii) Formatting
(iii) Process management function
(iv) Memory management function
(v)   Input/output management function
(vi) File management function
(vii) Task management function.

(i) **Booting :** Booting is a first step installation for operating system. In booting, copies BIOS (Basic Input/ Output System) program from ROM (Read Only Memory) chips to main memory.

(ii) **Formating :** Formating is a process in which data store at different places on the hard disk is deleted permanently.

(iii) **Processor Management :** Processor management function of a operation is related with allocation of process or jobs to processor.

The speed of processor or CPU is much faster than peripheral device. Since speed of CPU is much faster than input/output device, its a responsibility of operating system that CPU remains idle for minimum amount of time. During the waiting time, processor is switch to another process which is already is waiting for it.

(iv) **Memory Management :** The memory management function find all available free space in memory and allocates it to the processes. The operating system manage whole memory by dividing it into several blocks and parties. Every job under execution stage requires memory. If there exist two or more than two processes, then operating system allocates the memory segment as per requirement of the jobs. It finally frees the allocated memory space as and when the process is executed completely.

(v) **Input/Output Management :** This function of operating system deals with allocation of device to process. Operating system schedule the input/output devices, input/output channel to process whenever it require. In input/output scheduling after finding the status of devices, channels, a schedule is prepared to which state, which process, which device is allocated to the process.

(vi) **File Management :** File management of operating system is related with management of file stored on secondary storage. The file is considered as collection of logically related data item which are organized in records. These file may contain set of commands executable codes or reports. Purpose of file management is to hide the complex details of devices specific aspects from the user. It essentially does the task of mapping the request of file from process to physical disk, swaping the file content between primary and secondary memory.

7. Task Management : In case of multiple task, operating system should keep track of each task in a sequences.

Q. 3. (b) Differentiate between DOS, UNIX and window operating system.

Ans. DOS (Disk Operating System) : Disk operating system is a set of computer program. Disk operating system is command based operating system which is single user operating system. As name suggests, the operating system is used for operating the computer. The main function of DOS one to manage disk file, allocate system resources according to requirement. DOS provide feature essential to control hardware devices such as keyboard, screen, Disk devices, printers etc.

Basically DOS is a medium through which the user and external device attached to device communicating to the system. DOS translate the command issued by the user in the formate that is understandable by the computer & instruct computer to work accordingly.

Internal DOS Command :

   md, mkdir → Make or create directory

   cd, endir → Change the directory

    cls. → Clear the screen.

Unix : Unix was the first to exploit at the Internet. This was developed at AT & T's Bell laboratories, USA in 1969 as an operating system for minicomputer.

Unix is a multitasking system operating system for multiple users that has built-in networking capability.

It is used mostly on mainframes, workstation and servers rather than on PCs. Like MS-DOS, unix uses a command-line interface.

Window Operating System : Windows operating system is a GUI (Graphical User Interface) which make very attractive to the new users due to many feature availability such as GUI, menu-driven and command driven interface.

Window operating system are operated by pointing and clicking on a digital "desktop". There are three types of interface :

(i) Command driven

(ii) Graphical

(iii) Menu driven.

Q. 4. (a) Explain briefly categorisation of High level languages.

Ans. High Level Language : High level language is a third generation language. A programming language in which the program statement are not closely related to the computer is called a high level language. One statement in a high level language will be expand into several machine language instruction. High level language is easy to write a program because

Language like FORTRAN (Formula Translation) and COBOL (Common Business Oriented Language) made it possible for scientists and business people to write a program using familiar terms.

The first use of high level language in the early 1960's charged programming into some thing quite different from what it had been. Programs were written is an English like manner, making them more convenient to use and giving the programmer more time to address a client problem.

A few high-level language like Fortron (Formula translator), BASIC (Beginners All-Purpose symbolic instruction code) or COBOL (Common Business Oriented Language) all considered to be general purpose language.

A translator is need to translator High level programming language into computer-executable machine language. The programe that translate HLL into machine language is called compiler.

Q. 4. (b) Differentiate between compiler and Interpreter.

Ans. A compiler is a computer program (or set of programs) that translate text written in a computer language (some language) into another computer language (target language).
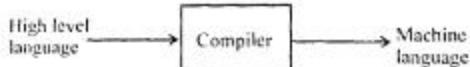
A compiler can translate only those source program which have been written in that language for which the compiler is meant for. For example FORTRAN compiler will not compile source code written in COBOL language.

Interpreter : An interpreter is another type of program translator used for translating Higher-Level Language (HLL) into Machine Level Language (MLL). It takes one statement of higher level language, translate it into machine language, translate it into machine language and immediately execute it.

Translation and execution is carried out for each statement while compiler translate whole program at a time.

Q. 4. (c) Describe the role of loaders.

Ans. Loader : An absolute loader can only load program with load origin = linked origin. This can be inconvenient if the load address of a program is likely to be different for different execution of a program.

A relocating loader perform reallocation while loading a program for execution. This permits a program to be executed in different parts of memory.

Q. 5. (a) What do you mean by network topology? Describe various network topologies in detail.

Ans. Network Topology : Network topology is the study of the arrangement of the element (links and nods) of a networks especially the physical (real) and logical (virtual) interconnection between nodes.

There one following types of topologies :

(a) Bus topology

(b) Star topology

(c) Ring topology

(d) Mess topology

(a) Bus Topology : In bus topology, signals one broadcast to all station. Each computer checks the address on the signal (data frames) as it passes along the bus.
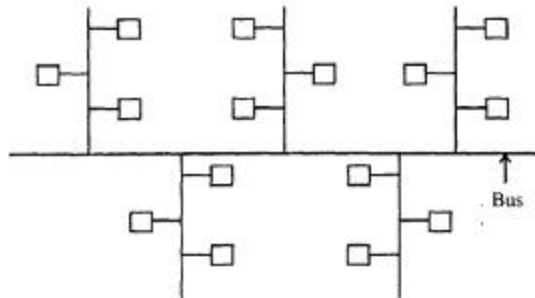
(a) Bus Topology

If signal address matches to the computer then computer process the signal other wire no action is takes place.

Advantage:

(i) Easy to connect to computer.

(ii) Require less cable length and no central failure.

Disadvantage:

(i) Entire network shutdown if main cable is break.

(ii) Maintenance cast is higher.

(b) Star Topology : In star topologies all the nodes such as printers, PC's, other peripheral devices are connected to central device.
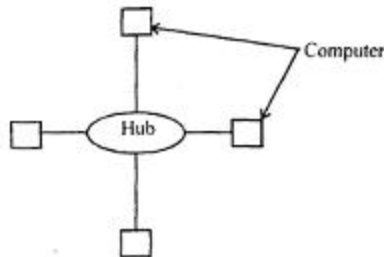
Advantage:

(i) Easily manageable.

(ii) Easy to install and wire.

(iii) Easy to address station.

Disadvantage:

(i) If central device hub is fail then whole network break

(ii) More expensive.

Ring Topology : In ring topology, the computer on the network form a logical ring. The ring represent a cable which connect all the computer.

Computer

Advantage :

(i)   All computer have equal access.

(ii)   Data packets at a greater speed.

Disadvantage :

(i)   Expensive topology

(ii)   Failure of one computer, break whole network.

Q. 5. (b) Explain briefly the layers of OSI model.
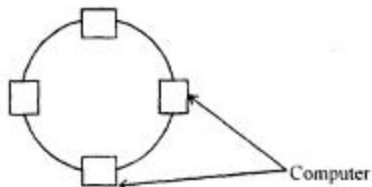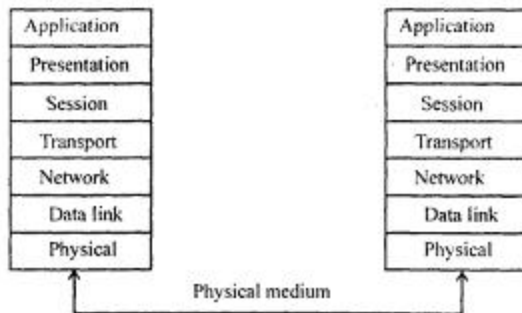
Ans. OSI Model (Open System Interconnection) : OSI reference model is designed in 1984. OSI (International Standard Organization) has proposed OSI reference model which provide a framework for data communication protocols.

OSI Model is Consists of Seven Layers :

Application
Presentation
Session
Transport
Network
Data link
Physical

Application
Presentation
Session
Transport
Network
Data link
Physical

Physical medium

Physical Layer : The main function of this layer is to send the databit stream over a communication network. This layer confirm that a bit is send sublaged sender.

Data Link Layer : Function of this layer is detecting and correcting errors. Data link layer take care of speed miss match between sender and receiver it also deals with framing.

Network Layer : Network layer is mainly concerning routing for data packets it is also decided optimal path for routing.

Transport Layer : Role of transport layer is to receive the data from session layer and dividing it into segment in order to pass it to the network layer.

Session Layer : This layer is establishes a session for communication between the two has. It is also involve in authentication.

Presentation Layer : This layer server as an interface between the application and communication network. This layer also deals with encoding and decoding.

Application Layer : The major role of application layer is to provide the facility to use the other layers. A user can interact with the system using telnet, FTP, SMTP etc. protocol.

Q. 6. (a) What do you mean by operator? Explain different types of operators used in 'C'.

Ans. Operators : Operators are the verbs of a language that help the user perform computation on values.

'C' language support following operators :

(a)   Arithmatic operators

(b)   Unary operators

(c)   Relational operators

(d)   Logical operators

(e)   Assignment operators

(f)   Bitwise operators.

(a) Arithmatic Operators : These operators are used for arithmatic calculation of numeric values.

Operators	Symbol	Purpose
Addition operator	+	Add two number
Subtraction operator	−	Subtact two number
Multiplication operator	*	Multiply two number
Division operator	/	Divide two number
Modulus operator	%	Provides remainder value that result from integer division.

(b) Unary Operators : Those operator which operate on a single operand are called unary operator.

(i) Unary minus (-)

(ii) Increment (+ +)

(iii) Decrement (- -)

(c) Relational Operators : Those operator operates upon expression and evaluate it to find whether it is true or false.

Operator	Symbol		Uses
<	less than		a < b
< =	Less than or equal to		a < = b
>	Greater than		a > b
> =	Greater than or equal to		a > = b

(d) Logical Operators : Logical operators operates upon two logical expression to build more complex logical expression which are either true or false.

(i) Logical AND operator (&&)

(ii) Logical OR operator (!!)

(iii) Logical)

(e) Assignment Operators : Assignment operators, operated upon operands and assign the resultant value to a variable.

For Example : $Y = 200$ **Ans.**

(f) Bitwise Operators :

(i) Bitwise AND operator (&)

(ii) Bitwise OR operator (!)

(iii) Bit wise NOR operator (!)

Example : Bitwise AND operator '&)

Operand–1	Operand–2	Operand–1 & Operand–2
0	0	0
0	1	0
1	0	0
1	1	1

Q. 6. (b) What is an Array? How arrays and Strings are related to each other? Explain with example.

Ans. Array : Arrays one a data structure which hold multiple variables of same data type.

Array One Two Types :

(i) One-D array.

(ii) Multi-dimensional array (2 or more)

One-Dimensional Array (1-D array) :

Declaration :

Type   arr_name   [size];
↑   ↑   ↑
Bow type   Law of   Size define
of array   array   number of
                       element
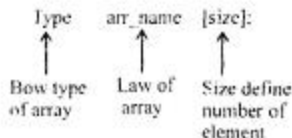
For Example : int a [5]

a [0]	a [1]	a [2]	a [3]	a [4]
Ist element	2nd element	3rd element	4th element	5st element
0	1	2	3	4

Array index from 0 to 4.

Because in 'C', the array index always begin with zero (0).

2-D Array : To store two dimensional data structure, we need 2-D Array.

For example Matrix.

Declaration of 2-D Array : data-type array-name [raw-size] [column-size]

For Example

int a [2] [2]

Array and String : Strings one defined as arrays of characters.

For Example : Char string [30];

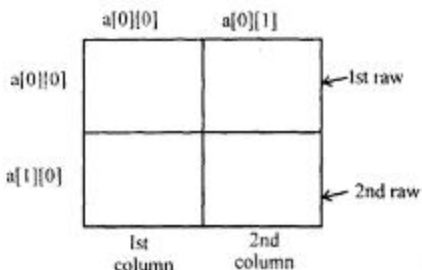Q. 7. (a) Write a program to swap two numbers using functions.

Ans.

```
# include <stdio.h>
void exchange (float *x. float * y)
{
        float t;
        t = *x;
        *x = *y;
        *y = t;
}
void exchange (float *, float *);
void main( )
{
        float x, y;
        printf ("\n Input value for x and y:");
        scanf ("%f", &x);
        scanf ("%f", & y);
        printf ("\n value before function call)
        printf ("\n value of x = %f". x);
        printf ("\n value of y = %f", y);
        printf ("\n value after functions call");
        exchange (&x. &y);
        printf ("\n value of x = % f', x);
        printf ("\n value of y = %f", y);
}
```

Q. 7. (b) Diff

Ans. Structure : Structure is basically a logical group of multiple items of different data types such as integer, float. character etc.

Conceptual diagram of a structure is given below :

```
Student                    Structure
   ┌──────→ Roll number    (int data type)        ┐
   │                                                │
   ├──────→ Age            (int data type)         │
   │                                                ├  Members of
   ├──────→ Religion       (character data type)   │   structure
   │                                                │
   └──────→ Fees           (float data type)       ┘
```
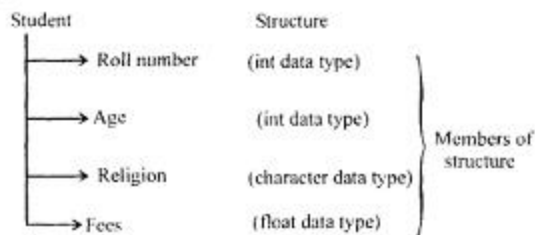
Declaring a Structure :

        Struct structure_name

        {

                data-type of member 1   member 1 name;
                data type of member 2   member 2 name;
                ---------------------------   -----------------------
                ---------------------------   -----------------------
                data type of member n   member n name;

        };

Here struct is keyword i.e., define structure. while structure name is a choice of student.

Thus, structure illustrated in above fig. can be defined as :

        struct student

        {

                int roll-no;

                int age;

                char religion:

                float fees;

        }

Union : Union is a special type of structure whose member share common memory space. Thus, at any given point of time, only one member will remain present in the memory.

General formate for defining the chain :

        Union union-name

        {

                member 1;

                member 2;

                ----

member n;
};

Where union is a keyword and rest of things like a structure.
Thus, union illustrated in above fig. can be define as

union item
{
                int size;
                char color [10];
                float amount;
}
shirt;

Example: # include <stdio.h>
            #include <stdlib.h>
            union item
            {
                int size;
                char color [10];
                float amount;
            }
            shirt;
            void main( )
{

Example : WAP to define a structure named employee having following element :

(i) Employee number (int type)
(ii) Name (character type)
(iii) Salary (float type)

This program create a variable alok of this data type accepting member's values from keyboard & print them.

                # include <stdio.h>
                struct employee
                {
                        int emp no.;
                        char name [30];
                        float salary;
                }
                alok;
                void main( )
                {

```
scanf ("%d", & alok.emp no);
printf("\n Enter name");
scanf ("%s", & alok name);
printf ("\n Enter salary");
scanf ("%f", & alok.salary);
printf ("\n Employee number %d", alok. emp no);
printf ("\n name %s", alok.name);
printf ("no salary %f", alok.salary);
}
shirt.size = 42;
shirt.color = "Blue"
shirt.amount = 500.55;
printf ("\n shirt size is % d", shirt.size);
printf ("\n short color is % s", shirt.amount);
}
```

Q. 8. Differentiate between Array of pointer and pointer to an Array with example.

Ans. Array of Pointer : An array is actually pointer to the 0th element of the array. Deferencing the array name will give 0th element.

For Example : arr is an array.

Array Access	Pointer Equivalent
arr [0]	*arr
arr [2]	* (arr + 2)
arr [n]	* (arr+n)

A function, which expects to be passed on array can declare that parameter in one of two ways

int arr [ ]; or int * arr;

Since a pointer variable always contain an address an array of pointer would be nothing but a collection of addresses. The address present in the array of pointer can be address of isolated variable or address of array elements or any other address. All rule that apply to an ordinary array apply into to the array of pointers as well.

Consider an example in which we show array of pointer in memory,

```
main ( )
{
        int * arr [4];
        int i = 31, j = 5, k = 19, 1 = 71 m;
        arr [0] = & i;
        arr [i] = &s;
        arr [2] = & k;
```
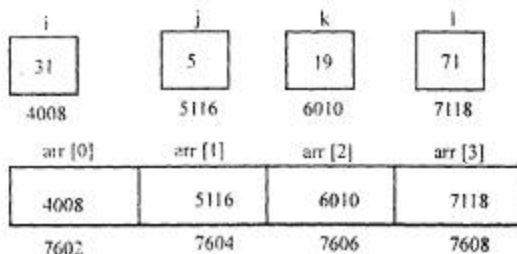
```
        arr [3] = &l;
        for (m = 0; m < = 3; m + +)
        printf ("%d", * (arr [m]));
}
```

Above figure shows the content and the arrangement of array of pointer in memory. As you can observe, arr contain address of isolated int variable i, j, k, l. The for loop in the program picks up address present in arr and print the value present at these address.

i	j	k	l
31	5	19	71
4008	5116	6010	7118

arr [0]	arr [1]	arr [2]	arr [3]
4008	5116	6010	7118
7602	7604	7606	7608

An array of pointers can even contain the addresses of the other arrays.

The following program would justify this :

```
main( )
{
        int a[ ] = {0, 1, 2, 3, 4};
        int * p [ ] = (a, a+1, a+2, a+3, a+4};
        printf ("\n %d", p, *p, *(*p));
}
```

Pointer to Array : An array is a actually very much like pointer. We can declare the array first element as a[0] or as int * a because a[0] is an address and * a is also an address the form of declaration is equivalent. The difference is pointer is a variable & can appear on the left the assignment operator that is l value.

The array name is constant and cannot appear as the left side of assignment operator.

Example : WAP to display the content of array using pointer.

```
main( )
{
        int a [100];
        int i, j, n;
        printf ("\n Enter the elements of array");
        scanf ("%d", & n);
        printf ("Enter the array element");
```

```
for (i = 0; i < n; i+ +)
scanf ("%d".& a[i]);
printf ("Array elements are");
for (ptr = a, ptr < (a+n); ptr + +)
printf ("Value of a [%d] = %d stored at address % u",j+ = x ptr, ptr);
```

Q. 9. (a) What are the different functions related to file handling in 'C'?

Ans. There are following function related to file handling in 'C'.

(a) Operating a data file or text file.

(b) Reading a text file.

(c) Writing a in a text file.

(d) Closing a data file.

While working with file in 'C' language you have to establish a buffer area first. General formate for establishing a buffer area is as follows :

FILE * pointer-variable; or FILE * fp

Here FILE (all upper case letter) is a special structure type that establishes the buffer area and the pointer-variable is the name of pointer variable that point to the beginning of the buffer area?

(a) Opening a Text File : After defining the buffer area, you have to open the text file, so that read/write operation could be performed.

The library function fopen opens the file. General formate for opening a file—

pointer-variable = fopen (filename, mode);

fp = fopen ("PRI.", "r");

where fp is pointer-variable, which contain address of the structure FILE.

fopen ( ) will open a file "PRI.C" in 'read' mode which tells 'C' compiler that are would be reading the contents of the file.

(b) Reading from a Text File : To read the file's contents from memory there exist a function called fget( ).

ch = fgetc (fp); or ch = fgete (pointer-variable)

fgetc( ) reads the character from current ptr position and return the character that is read and assigned it to a character variable ch.

(c) Writing to a text file :

ch = f putc (fp);

C provide fputc function to write a single character in the file.

(d) Closing a text file : After performing all the operation on the file such as (fopen, fgetc( ), fputc ()) all the opened file should be closed by using a function fclose. General formats for closing a file.

fclose (fp);

While opening a file there one more that one made which one given below :

Mode	Meaning
"r"	open the file for read operation only.
"r+"	opening on file for read and writing.
"w"	Open the file for writing only.
"wt"	Open a file for both reading and writing file.
"a"	Open the file for opending.
"a+"	Open the file for reading and appending.

Q. 9. (b) What do you mean by Dynamic memory allocation? Explain.

Ans. Dynamic Memory Allocation ; ANSIC provides five standard function that help you allocate memory on the heap.

1. Dynamic memory allocation : : size of ( ) size of ( ) return a size –t of the item passed in. So on the typical 32-bit machine, size of (int) return 4 bytes. Size-t is just an unsigned integer constant. Size of ( ) is helpful when using malloc or calloc called.

2. Dynamic Memory Allocation :: malloc (3), calloc(3); realloc(3), true (3)

    The prototype for malloc (3) is

    void * malloc (size – + size);

    malloc take in a size –t and return a void pointer because it does not matter to malloc to what type. This memory will be used for

Let's see an example of how malloc is used :

    int * ip;

    ip = malloc (5 * size of (int));

    or ip = malloc (5* size of (ip));

    size of (int) return the sum of an integer on the multiply by 5 and malloc that many p types.

The second malloc works became it sends what input is pointing to which is on int.

3. Dynamic Memory Allocation : : calloc (3)

In certain case, where malloc fail. calloc (3) works like malloc, but initialize the memory to zero if possible.

    The prototype for calloc (B) is

    Void * calloc (size-tnmemh, size-size);

4. Dynamic Memory Allocation : : realloc (3). If we run out of allocated memory during run-time of our program, then we need more memory so need readloc(3)

The prototype of realloc (3) is

    void * realloc (void * ptr, size-t six);

5. Dynamic Memory Allocation : : free (3)

We need to be able to free the memory space if we have no more use for it any more.

The prototye for free (3) –

    void free (void * ptr)

Free simply takes in a pointer to free.