Q.2 a. Find the decimal equivalent of:
(i) $(2B.C4)_{16}$ (ii) $(110.101)_2$

Answer:

i) $(2B.C4)_{16}$ $= 2 \times 16^1 + B \times 16^0 + C \times 16^{-1} + 2 \times 16^{-2}$

$= 32 + B + C/16 + 4/256$

$=32 + 12 + 11/16 + 4/256$

$=43 + 0.75 + 0.015625$

$=(43.765652)_{10}$

ii) $(110.101)_2$ $= 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$

$= 4 + 2 + 0 + 0.5 + 0 + 0.125$

$=6 + 0.5 + 0.125$

$=(6.625)_{10}$

b. What is a Computer? Why is it known as data processor? List some important characteristics of a Computer.

Answer:
Computer: A computer is a programmable machine or device that performs pre-defined or programmed computations or controls operations that are expressible in numerical or logical terms at high speed and with great accuracy.
Computer is a fast operating electronic device, which automatically accepts and store input data, processes them and produces results under the direction of step by step program. Any Process that uses a computer program will enter data and summarize, analyze or otherwise convert data into usable information. The process may be automated and run on a computer. It involves recording, analyzing, sorting, summarizing, calculating, disseminating and storing data. Thus Computer is known as data processing system.
Its various operations are:
1) It accepts data or instructions by way of input.

2) It stores data.

3) It can process data as required by the user.

4) It gives results in the form of output.

Characteristics of computers:

· Speed

· Accuracy.

· Automation.

· Endurance.

· Versatility.

· Storage.

· Cost Reduction.

c. **Differentiate between the characteristics of primary and secondary storage of a computer system.**

Answer:

Primary storage. The primary storage, also known as main memory, is used to hold pieces of program instructions and data, intermediate result of processing, and recently produced results of processing, of the job(s), which the computer system is currently working on. These pieces of information are represented electronically in the memory chip's circuitry, and while it remains in the main memory, the central processing unit can access it directly at a very fast speed. However, the primary storage can hold information only while the computer system is on. As soon as the computer system is switched off or reset, the information held in the primary storage disappears. Moreover, The primary storage normally has limited storage capacity, because it is very expensive. The primary storage of modern computer systems is made up of semiconductor devices.

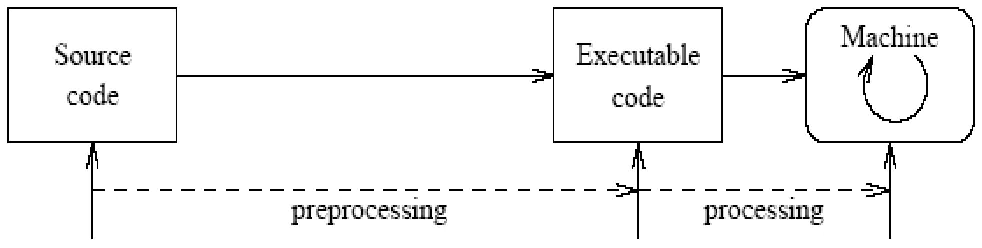
Secondary Storage. The Secondary storage, also known as auxiliary memory, is used to take care of the limitations of the primary storage. That is, it is used to supplement the limited storage capacity and the volatile characteristic of primary storage. This is because secondary storage is much cheaper than primary storage, and it can retain information even when the computer is switched off or reset. The secondary storage is normally used to hold the program instructions data, and information of those jobs, on which the computer system is not working on currently, but needs to hold them for processing later. The most commonly used secondary medium is the magnetic disk.

Q.3 a. Differentiate between Compiler and Interpreter.

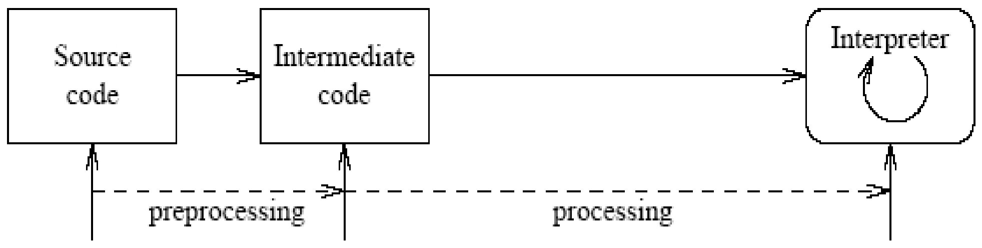
Answer:

Compiler vs. Interpreter

An interpreter translates some form of source code into a target representation that it can immediately execute and evaluate. The structure of the interpreter is similar to that of a compiler, but the amount of time it takes to produce the executable representation will vary as will the amount of optimization. The following diagram shows one representation of the differences.



Compilation



Interpretation

Compiler characteristics:

- spends a lot of time analyzing and processing the program
- the resulting executable is some form of machine- specific binary code
- the computer hardware interprets (executes) the resulting code
- program execution is fast

Interpreter characteristics:

- relatively little time is spent analyzing and processing the program
- the resulting code is some sort of intermediate code
- the resulting code is interpreted by another program
- program execution is relatively slow

Difference between compiler and interpreter

• A complier converts the high level instruction into machine language while an interpreter converts the high level instruction into an intermediate form.

• Before execution, entire program is executed by the compiler whereas after translating the first line, an interpreter then executes it and so on.

• List of errors is created by the compiler after the compilation process while an interpreter stops translating after the first error.

• An independent executable file is created by the compiler whereas interpreter is required by an interpreted program each time.

b. What is an Operating System? Why it is necessary for a Computer System?

Answer:
An operating system is an integrated set of programs that control the resources (CPU, memory, I/O devices etc.) of a computer system and provides its users with an interface that is more convenient to use than the bare machine.

Operating Systems are resource managers. The main resource is computer hardware in the form of processors, storage, input/output devices, communication devices, and data.

Operating system functions are: implementing the user interface, sharing hardware among users, allowing users to share data among themselves, preventing users from interfering with one another, scheduling resources among users, facilitating input/output, recovering from errors, accounting for resource usage, facilitating parallel operations, organizing data for secure and rapid access, and handling network communications.

Objectives of Operating Systems

Modern Operating systems generally have following three major goals.

Operating systems generally accomplish these goals by running processes in low privilege and providing service calls that invoke the operating system kernel in high-privilege state.

- **To hide details of hardware by creating abstraction**
  An abstraction is software that hides lower level details and provides a set of higher-level functions. An operating system transforms the physical world of devices, instructions, memory, and time into virtual world that is the result of abstractions built by the operating system. There are several reasons for abstraction.
  First, the code needed to control peripheral devices is not standardized. Operating systems provide subroutines called device drivers that perform operations on behalf of programs for example, input/output operations.
  Second, the operating system introduces new functions as it abstracts the hardware. For instance, operating system introduces the file abstraction so that programs do not have to deal with disks.
  Third, the operating system transforms the computer hardware into multiple virtual computers, each belonging to a different program. Each program that is running is called a process. Each process views the hardware through the lens of abstraction.
  Fourth, the operating system can enforce security through abstraction.

- **To allocate resources to processes (Manage resources)**
  An operating system controls how processes (the active agents) may access resources (passive entities).

- **Provide a pleasant and effective user interface**

The user interacts with the operating systems through the user interface and usually interested in the “look and feel” of the operating system. The most important components of the user interface are the command interpreter, the file system, on-line help, and application integration. The recent trend has been toward increasingly integrated graphical user interfaces that encompass the activities of multiple processes on networks of computers.

c. List various advantages and limitations of High-level languages.

Answer:
Advantages of High Level Language (HLL):

1. **Machine Independence:** High level languages are machine independent in the sense that a program written with HLL can be used on different computer systems.
2. **Ease of Learning :** Since HLL are English like , it is very easy to learn them. At the same time, the programmer needs not to learn about the hardware details of the computer system.
3. **Understandable program:** The HLL programs are more understandable as compared to the programs written in the low level languages.
4. **Low programming cost :** since it is easier to write a program in HLL , the productivity of a programmer increases manifolds. The increase in productivity reduces the overall programming cost.
5. **Easier to maintain :** As compared to low level languages , the program written in HLL can be easily modified because such programs are easier to understand. Therefore software maintenance becomes easy.
6. **Less errors :** As compared to low level languages, the program written in HLL is less error prone.

Limitations of High-level Languages:

1. **Lower efficiency :** A program written in HLL has lower efficiency than the one written in an assembly language or a machine language to do

the same job. i.e. programs written in HLL take more time to execute and require more memory space.

2. **Less flexibility :** HLLs are less flexible than assembly languages because they do not normally have instructions or mechanisms to control the computer's CPU , memory and registers.

The advantages of HLL far outweigh its disadvantages.

Q.4 a. What do you mean by Computer Network? List various objectives of using a Computer Network.

Answer:

A **computer network**, or simply a **network**, is a collection of computers and other hardware components interconnected by communication channels that allow sharing of resources and information. Where at least one process in one device is able to send/receive data to/from at least one process residing in a remote device, then the two devices are said to be in a network. Simply, more than one computer interconnected through a communication medium for information interchange is called a computer network.

Networks may be classified according to a wide variety of characteristics, such as the medium used to transport the data, communications protocol used, scale, topology, and organizational scope.

Properties :

Computer networks:

Facilitate communications Using a network, people can communicate efficiently and easily via email, instant messaging, chat rooms, telephone, video telephone calls, and video conferencing.

Permit sharing of files, data, and other types of information In a network environment, authorized users may access data and information stored on other computers on the network. The capability of providing access to data and information on shared storage devices is an important feature of many networks.

Share network and computing resources In a networked environment, each computer on a network may access and use resources provided by devices on the network, such as printing a document on a shared network printer. Distributed computing uses computing resources across a network to accomplish tasks.

May be insecure A computer network may be used by computer hackers to deploy computer viruses or computer worms on devices connected to the network, or to prevent these devices from normally accessing the network (denial of service).

May interfere with other technologies Power line communication strongly disturbs certain forms of radio communication, e.g., amateur radio. It may also interfere with last mile access technologies such as ADSL and VDSL.

May be difficult to set up A complex computer network may be difficult to set up. It may also be very costly to set up an effective computer network in a large organization or company.

b. What is Internet? List various applications areas where Internet plays an important role.

Answer:

Internet is interconnection of large number of heterogeneous computer networks all over the world that can share information back and forth. These interconnected network exchange information by using same standards and protocols.

Applications of Internet :

The internet is treated as one of the biggest invention. It has a large number of uses.

1. Communication: it is used for sending and receiving message from one and other through internet by using electronic mail. Some of the web sites providing this service are yahoomail.com Hotmail.com rediffmail.com etc

2. Job searches: getting information regarding availability of job in different sectors and areas. You can publish your resume in online for prospective job. Some of the web sites providing this service are naukri.com, monster.com, summerjob.com, recuritmentindia.com etc.
3. Finding books and study material : books and other study material stored around the world can be easily located through internet. Latest encyclopaedias are available online.
4. Health and medicine: internet provide information and knowledge about field of health medicine people can have information about various disease and can receive help .patient can be taken to virtual check room where they can meet doctors.
5. Travel: one can use internet to gather information about various tourist place . it can be used for booking Holiday tours , hotels, train and flights. Some of the web sites providing this service areindiatravelog.com, rajtravel.com, makemytrip.com.
6. Entertainment one can doun lode jokes, songs muvis, latest sports updates through internet Some of the web sites providing this service arecricinfo.com, movies.com espn.com
7. Shopping : internet is also used for online shopping. By just giving accounts details you can perform the transaction. You can even pay your bills and perform bank related transaction.
8. Stock market updates : you can sell or buy shares while sitting on computer through internet. Several websites like ndtvprofit.com, moneypore.com, provide information regarding investment
9. Research : a large number of people are using internet for research purposes you can download any kind information by using internet
10. Business use of internet: different ways by which intenet can be used for business are:
• Information about the product can be provided can be provided online to the the customer .

• Provide market information to the business

• It help business to recruit talented people

• Help in locating suppliers of the product .

• Fast information regarding customers view about companies product

• Eliminate middle men and have a direct contact with contact with customer
.

• Providing information to the investor by providing companies back ground and financial information on web site

Q.5 a. Define variable. List various rules to define variable in C.

Answer:
A variable is nothing but a name given to a storage area that our programs can manipulate. Each variable in C has a specific type, which determines the size and layout of the variable's memory; the range of values that can be stored within that memory; and the set of operations that can be applied to the variable.

The name of a variable can be composed of letters, digits, and the underscore character. It must begin with either a letter or an underscore. Upper and lowercase letters are distinct because C is case-sensitive.

Basic variable types:

Type	**Description**
Char	Typically a single octet(one byte). This is an integer type.
Int	The most natural size of integer for the machine.
Float	A single-precision floating point value.
double	A double-precision floating point value.

Void	Represents the absence of type.

C programming language also allows to define various other types of variables, like Enumeration, Pointer, Array, Structure, Union, etc.

Variable Definition in C:

A variable definition means to tell the compiler where and how much to create the storage for the variable. A variable definition specifies a data type and contains a list of one or more variables of that type as follows:

```
type variable_list;
```

Here, **type** must be a valid C data type including char, w_char, int, float, double, bool or any user-defined object, etc., and **variable_list** may consist of one or more identifier names separated by commas. Some valid declarations are shown here:

```
int    i, j, k;

char   c, ch;

float  f, salary;

double d;
```

The line **int i, j, k;** both declares and defines the variables i, j and k; which instructs the compiler to create variables named i, j and k of type int.

Variables can be initialized (assigned an initial value) in their declaration. The initializer consists of an equal sign followed by a constant expression as follows:

```
type variable_name = value;
```

Rules for writing variable name in C

1. Variable name can be composed of letters (both uppercase and lowercase letters), digits and underscore '_' only.
2. The first letter of a variable should be either a letter or an

underscore. But, it is discouraged to start variable name with an underscore though it is legal. It is because, variable name that starts with underscore can conflict with system names and compiler may complain.

3. There is no rule for the length of length of a variable. However, the first 31 characters of a variable are discriminated by the compiler. So, the first 31 letters of two variables in a program should be different.

b. What is storage class specifier? Why is it used? List various storage class specifiers used in C.

Answer:

Storage Class :

'Storage' refers to the scope of a variable and memory allocated by compiler to store that variable. Scope of a variable is the boundary within which a varible can be used. Storage class defines the the scope and lifetime of a variable.

From the point view of C compiler, a variable name identifies physical location from a computer where varaible is stored. There are two memory locations in a computer system where variables are stored as : Memory and CPU Registers.

Functions of storage class :

To detemine the location of a variable where it is stored ?

Set initial value of a variable or if not specified then setting it to default value.

Defining scope of a variable.

To determine the life of a variable.

Types of Storage Classes :

Storage classes are categorised in 4 (four) types as,

- Automatic Storage Class

- Register Storage Class
- Static Storage Class
- External Storage Class

Storage type	**Created**	**Initialized**	**Scope**	**purpose**
auto	Each time the function or block is called	Can be initialised at the time of declaration	With in the block or function	As variable with in a block
static	First time when the function is called	Initialised at the time of declaration	With in the block or function	As variables which retain value even after the termination of function.
register	same as auto	same as auto	same as auto	As most frequently used variables
extern	created in the file where it has been declared as global	initialised in the file where it has been declared as global	Both the files where declared as global and where declared as extern	as variables used by multiple files.

c. Write a C code that accept a character from keyboard and displays it in reverse case on screen.

Answer:

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

```
main()
{
        char c;
        printf("Enter an alphabet");
        putchar('\n');
        c=getchar();
        if (islower(c))
        putchar(toupper(c));
        else
        putchar(tolower(c));
}
```

Q.6 a. Describe IF...ELSE statement with the help of flowchart and an example.

Answer:

if...else statement

The if...else statement is used if the programmer wants to execute some statement/s when the test expression is true and execute some other statement/s if the test expression is false.

Syntax of if...else

```
if (test expression)
{
    statements to be executed if test expression is true;
}
else
 {
    statements to be executed if test expression is false;
}
```

Flowchart of if...else

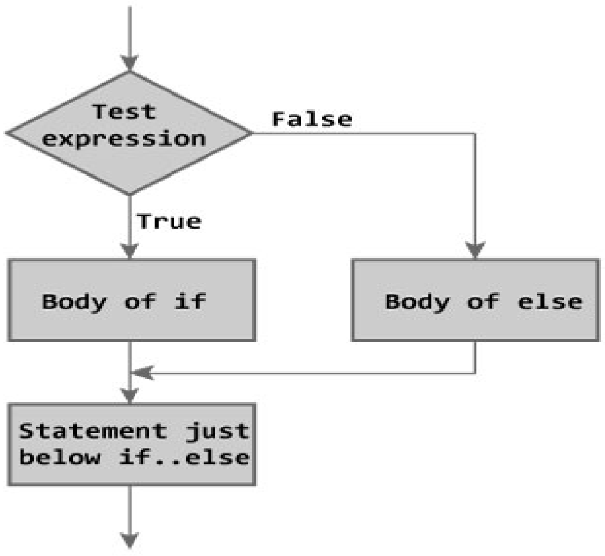


Figure: Flowchart of if...else Statement

statement:

Example program to check whether a number entered by user is even or odd

```
#include <stdio.h>

int main(){

      int num;

      printf("Enter a number you want to check.\n");

      scanf("%d",&num);

      if((num%2)==0)          //checking whether remainder is 0
or not.

           printf("%d is even.",num);

      else

           printf("%d is odd.",num);

      return 0;

}
```

b. Describe the conditional operator with the help of an example.

Answer:
The conditional operator is also known as ternary operator. It is called ternary operator because it takes three arguments. The conditional operator evaluates an expression returning a value if that expression is true and different one if the expression is evaluated as false.

Syntax:
condition ? result1 : result2

If the condition is true, result1 is returned else result2 is returned.

<u>Examples:</u>

10==5 ? 11: 12 // returns 12, since 10 not equal to 5.

10!=5 ? 4 : 3 // returns 4, since 10 not equal to 5.

12>8 ? a : b // returns the value of a, since 12 is greater than 8.

```
/*Conditional operator*/
#include<stdio.h>
void main(){
int a = 10, b = 11;
int c;
c = (a < b)? a : b
printf("%d", c);
}
```

Output:
```
10
```

c. Explain the use of switch statement in C with the help of flowchart.

Answer:
A **switch** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each **switch case**.

Syntax:
The syntax for a **switch** statement in C programming language is as follows:

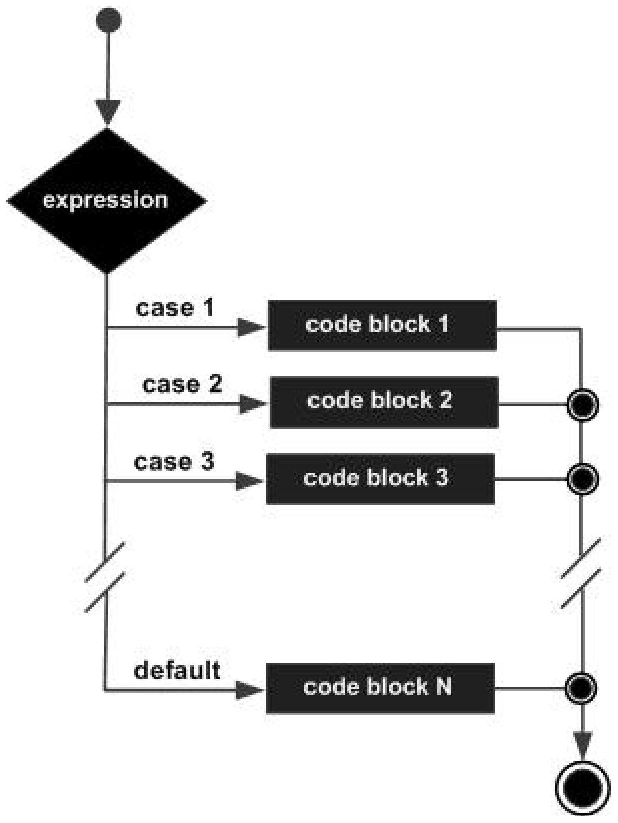
```
switch(expression){
   case constant-expression  :
      statement(s);
      break; /* optional */
   case constant-expression  :
      statement(s);
      break; /* optional */


   /* We may have any number of case statements */
   default : /* Optional */
      statement(s);
}
```

The following rules apply to a **switch** statement:

- The **expression** used in a **switch** statement must have an integral or enumerated type, or be of a class type in which the class has a single conversion function to an integral or enumerated type.
- You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.
- The **constant-expression** for a case must be the same data type as the variable in the switch, and it must be a constant or a literal.
- When the variable being switched on is equal to a case, the statements following that case will execute until a **break** statement is reached.
- When a **break** statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.
- Not every case needs to contain a **break**. If no **break** appears, the flow of control will *fall through* to subsequent cases until a break is reached.

- A **switch** statement can have an optional **default** case, which must appear at the end of the switch. The default case can be used for performing a task when none of the cases is true. No **break** is needed in the default case.



Example:

```
#include <stdio.h>

 int main ()

{

   /* local variable definition */

   char grade = 'B';


   switch(grade)

   {

   case 'A' :

      printf("Excellent!\n" );
```

```
    break;
  case 'B' :
  case 'C' :
    printf("Well done\n" );
    break;
  case 'D' :
    printf("You passed\n" );
    break;
  case 'F' :
    printf("Better try again\n" );
    break;
  default :
    printf("Invalid grade\n" );
  }
  printf("Your grade is  %c\n", grade );


  return 0;
}
```

When the above code is compiled and executed, it produces the following result:

```
Well done
Your grade is B
```

Q.7 a. Write a Program to compare two matrices for equality in C.

Answer:

Two matrices A and B are equal if the elements A[I][J] and B[I][J] are equal for all values of I and J within the specified range.

```
# include <stdio.h>
 main()
{
int A[10][10],B[10][10];
int i,j;
int m1,n1,m2,n2;
int flag;
printf("\n Enter the order of Ist matrix: ");
scanf("%d %d" , & m1,&n1);
printf("\n Enter the order of 2nd matrix: ");
scanf("%d %d" , & m2,&n2);
if ((m1 != m2) || (n1 != n2))
printf("\n Matrices can't be compared");
else
{
printf("\n Enter the Ist Matrix");
for(i=0;i<m1;i++)
for(j=0;j<n1;j++)
{
printf("\n Enter an element:");
scanf("%d",&A[i] [j]);
}
printf("\n Enter the 2nd Matrix");
for(i=0;i<m2;i++)
for(j=0;j<n2;j++)
{
printf("\n Enter an element:");
scanf("%d",&B[i] [j]);
}
flag=1;
for(i=0;i<m2;i++)
for(j=0;j<n2;j++)
{
if ( A[i][j] != B[i][j])
{
```

```
flag =0;
break;
}
}
If (flag)
printf("\n The matrices are equal");
else
 printf("\n The matrices are not equal");
}
}
```

b. Define string. Explain various string functions along with their syntax.

Answer:

String:

- A string is combination of characters.
- Any set or sequence of characters defined within double quotation symbols is a constant string.

String handling Functions:

- **strlen() function:**

This function counts and returns the number of characters in a particular string. The length always does not include a null character. The syntax of strlen() is as follows:

```
n=strlen(string);
```

Where n is the integer variable which receives the value of length of the string.

strcmp function:

In c,we cannot directly compare the value of 2 strings in a condition like if(string1==string2) Most libraries however contain the function called strcmp(),which returns a zero if 2 strings are equal, or a non zero number if the strings are not the same. The syntax of strcmp() is given below:

```
strcmp(string1,string2)
```

strcmpi() function

This function is same as strcmp() which compares 2 strings but not case sensitive.

```
strcmpi(string1,string2)
```

strcpy() function:

To assign the characters to a string,C does not allow you directly as in the statement name=Robert; Instead use the strcpy() function found in most compilers the syntax of the function is illustrated below.

```
strcpy(string1,string2);
```

strlwr () function:

This function converts all characters in a string from uppercase to lowercase The syntax of the function strlwr is illustrated below

```
strlwr(string);
```

strrev() function:

This function reverses the characters in a particular string. The syntax of the function strrev is illustrated below

```
strrev(string);
```

The following program illustrate the use of string functions:

```
/* Example program to use string functions*/
#include < stdio.h >
#include < string.h >
void main()
{
char s1[20],s2[20],s3[20];
int x,l1,l2,l3;
printf(”Enter the strings”);
scanf(”%s%s”,s1,s2);
x=strcmp(s1,s2);
if(x!=0)
{printf(”\nStrings are not equal\n”);
strcat(s1,s2);
}
else
printf(”\nStrings are equal”);
strcpy(s3,s1);
l1=strlen(s1);
l2=strlen(s2);
l3=strlen(s3);
printf(”\ns1=%s\t length=%d characters\n”,s1,l1);
printf(”\ns2=%s\t length=%d characters\n”,s2,l2);
printf(”\ns3=%s\t length=%d characters\n”,s3,l3);
}
```

Q.8 a. What is a function? Explain various types of functions used in C.

Answer:
Functions are the building blocks where every program activity occurs.

• Functions are self-contained program blocks that carry out some specific, well- defined task.

• User defined functions are subordinate to main() and also to one another.

• If facilitates top-down modular programming.

• Length of program can be reduced using functions.

• It makes debugging easier.

• It provides the facility of reusability of module.

There are five types of functions and they are:

1. Functions with no arguments and no return values.
2. Functions with arguments and no return values.
3. Functions with arguments and return values.
4. Functions that return multiple values.
5. Functions with no arguments and return values.

Functions with no arguments and no return value.

A C function without any arguments means you cannot pass data (values like int, char etc) to the called function. Similarly, function with no return type does not pass back data to the calling function. It is one of the simplest types of function in C. This type of function which does not return any value cannot be used in an expression it can be used only as independent statement.

Functions with arguments and no return value.

A C function with arguments can perform much better than previous function type. This type of function can accept data from calling function. In other words, you send data to the called function from calling function but you cannot send result data back to the calling function. Rather, it displays the result on the terminal. But we can control the output of function by providing various values as arguments.

Functions with arguments and return value.

This type of function can send arguments (data) from the calling function to the called function and wait for the result to be returned back from the called function back to the calling function. And this type of function is mostly used in programming world because it can do two way communications; it can accept data as arguments as well as can send back data as return value. The data returned by the function can be used later in our program for further calculations.

Functions with no arguments but returns value.

We may need a function which does not take any argument but only returns values to the calling function then this type of function is useful. The best example of this type of function is "getchar()" library function which is declared in the header file "stdio.h". We can declare a similar library function of own.

Functions that return multiple values.

In a function, return statement was able to return only single value. That is because; a return statement can return only one value. But if we want to send back more than one value then how we could do this?

As arguments are used to send values to the called function, in the same way we can also use arguments to send back information to the calling function. The arguments that are used to send back data are called **Output Parameters**.

b. **Differentiate between the various parameter passing techniques used in a C function.**

Answer:

Call by Value	Call by Reference
This is the usual method to call a function in which only the value of the variable is passed as an argument	In this method, the address of the variable is passed as an argument
Any alternation in the value of the argument passed is local to the function	Any alternation in the value of the argument passed is accepted in

and is not accepted in the calling program	the calling program(since alternation is made indirectly in the memory location using the pointer)
Memory location occupied by formal and actual arguments is different	Memory location occupied by formal and actual arguments is same and there is a saving of memory location
Since a new location is created, this method is slow	Since the existing memory location is used through its address, this method is fast
There is no possibility of wrong data manipulation since the arguments are directly used in an application	There is a possibility of wrong data manipulation since the addresses are used in an expression. A good skill of programming is required here

Q.9 a. What is dangling pointer in C? Explain with the help of example. How it is differentiated with null pointer.

Answer:
A dangling pointer is that pointer which has been allocated but does not point to any entity. Such an uninitialized pointer is dangerous in the sense that a deference operation on it will result in an unpredictable operation or a runtime error.

e.g.
```
int * ptr;
* ptr = 50;
```

The first statement allocates the pointer called ptr. Currently ptr is a dangling pointer. The second statement is trying to load a value 50 to a location which is pointed by ptr. This is a dangerous situation and consequently the results will be unpredictable.

Therefore , if pointers are being used , the following steps must be followed:

1. Allocate the pointe.
2. Allocate the variable to which the pointer is to be pointed.
3. Point the pointer to the variable.

Thus the correct code is

```
int * ptr;

int val;
ptr = &val
* ptr = 50;
```

b. Explain the concept of file in C. What are different methods of opening a file?

Answer:
The console oriented I/O functions such as scanf() and printf() which are used for reading and writing data always uses the terminal(keyboard and screen) as the target place. This works fine as long as the data is small. However many real life problems involves large volumes of data and in such situations , the console oriented I/O operations results in following problems:

1. It becomes cumbersome and time consuming to handle large volumes of data through terminals.

2. The entire data is lost when either the program is terminated or the computer is turned off.

It is therefore necessary to have a more flexible approach where data can be stored on the disks and read whenever necessary without destroying the data. This method employs the concept of files to store data. Thus a file is a place on the disk where a group of related data is stored.

Modes of opening a file:

Mode	Purpose
“r”	Open a file for reading. If file does not exist, NULL is returned.
“w”	Open a file for writing. If the file does not exist a new file is created . if the file exists, the new contents overwrites the previous contents.
“r+”	Open the file for both reading and writing. If file does not

	exist, NULL is returned.
“w+”	Open the file for both reading and writing. If file does not exist, the new contents overwrites the previous contents
“a”	Open a file for appending. If the file exists, the new data is written at the end of the file else the new file is created.
“a+”	Open the file for reading and appending. If the file does not exist, a new file is created.

TEXT BOOKS

I. Fundamentals of Computers, V. Rajaraman, Fourth Edition, PHI, 2007
II. Programmeming in ANSI C, E. Balagurusamy, Third Edition, Tata McGraw Hill