Q.2 a. What is an algorithm? State and explain basic characteristics of algorithm.

Answer:
An **algorithm** is a set of instructions that a **computer** follows, generally to accomplish one specific task.
Finiteness:
An algorithm must terminate after a finite number of steps and further each step must be executable in finite amount of time. In order to establish a sequence of steps as an algorithm, it should be established
that it terminates (in finite number of steps) on all allowed inputs.
Definiteness (no ambiguity):
Each steps of an algorithm must be precisely defined; the action to be carried out must be rigorously and unambiguously specified for each case.
Inputs:
An algorithm has zero or more but only finite, number of inputs.
Output:
An algorithm has one or more outputs. The requirement of at least one output is obviously essential, because, otherwise we cannot know the answer/ solution provided by the algorithm. The outputs have specific relation to the inputs, where the relation is defined by the algorithm.
Effectiveness:
An algorithm should be effective. This means that each of the operation to be performed in an algorithm must be sufficiently basic that it can, in principle, be done exactly and in a finite length of time, by person using pencil and paper. It may be noted that the 'FINITENESS' condition is a special case of 'EFFECTIVENESS'. If a sequence of steps is not finite, then it cannot be effective also.

b. Find (i) Decimal equivalent of $(11001)_2$
(ii) Decimal equivalent of $(1C7)_{16}$
(iii) Binary equivalent of $(18)_{10}$
(iv) Decimal equivalent of $(1071)_8$

Answer:
(Show steps to get the following answers)
(11001)2=(25)10
(1C7)16=((455)10
(18)10=(10010)2
(1071)8=(569)10

Q.3 a. Why do we need an Operating System? List basic functions of an operating system.

Answer:

Basic functions of an operating system:

- controlling the user interface
- controlling tasks in progress
- controlling access to data
- allocating resources
- memory management
- process scheduling

b. Compare the characteristics of impact and non-impact printers with examples. What are digitizers?

Answer:

Impact printers print the document character by character, with a writing head striking on an inkcoated ribbon to get the character printed. Non-impact prints do not print by any such physical impact, rather it prints block by block or line by line with ink or laser jet. Examples of impact printers are daisy wheel and dot matrix printers, whereas non-impact printers are laser or ink-jet printers. A digitizer is an input device used for drawing free-hand images and graphics. This is a pad with a grid of sensor's wire. A pen is moved on to the grid and every movement of the pen on the gird is captured. This pen is also known as stylus.

Q.4 a. Differentiate between LAN, MAN and WAN. Show with the help of a diagram how will you connect five computers on a LAN.

Answer:

Local Area Network (LAN)

LAN is usually privately owned and links the devices in a single office, building or campus of up to few kilometers in size. These are used to share resources (may be hardware or software resources) and to exchange information. LANs are distinguished from other kinds of networks by three categories: their size, transmission technology and topology.

LAN typically used transmission technology consisting of single cable to which all machines are connected. Traditional LANs run at speeds of 10 to 100 Mbps (but now much higher speeds can be achieved). The most common LAN topologies are bus, ring and star.

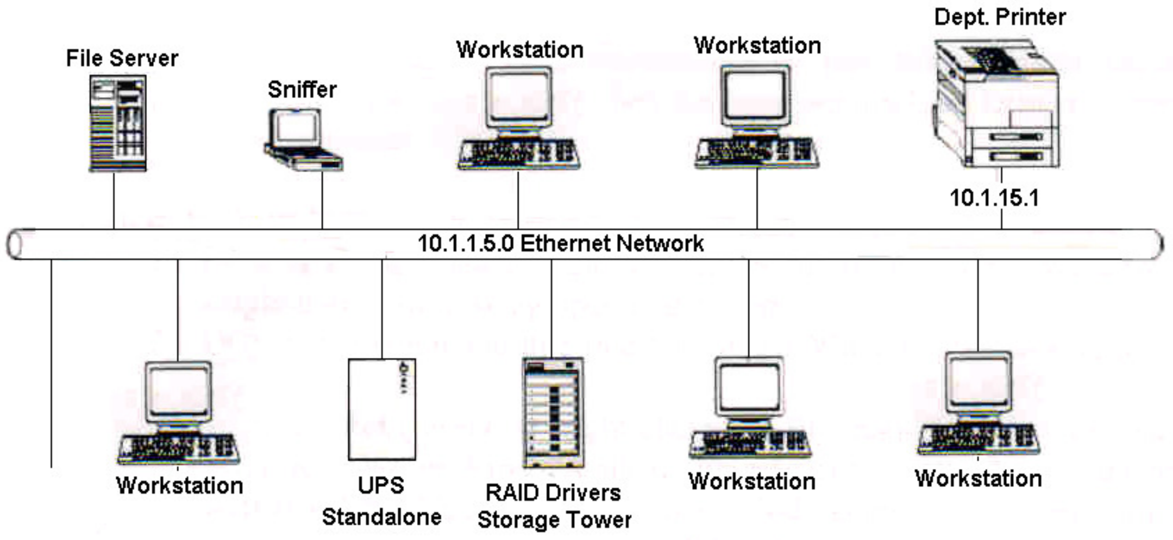
Metropolitan Area Networks (MAN)

MAN is designed to extend over the entire city. It may be a single network as a cable TV network or it may be means of connecting a number of LANs into a larger network so that resources may be shared. For example, a company can use a MAN to connect the LANs in all its offices in a city. MAN is wholly owned and operated by a private company or may be a service provided by a public company.

Wide Area Network (WAN)

WAN provides long-distance transmission of data, voice, image and information over large geographical areas that may comprise a country, continent or even the whole world. In contrast to LANs, WANs may utilize public, leased or private communication devices, usually in combinations, and can therefore span an unlimited number of miles. A WAN

that is wholly owned and used by a single company is often referred to as *enterprise network.*
We can connect five computers on a LAN as shown below:



b. Distinguish between minicomputer, microcomputer and mainframe computer.

Answer:
Mini Computer: Smaller than mainframe. Not portable. It integrates commercial and technical operations better than the more powerful computers.
Micro computer: Smallest in size and Portable
Mainframe Computer: Great power and storage capacity. Large computers. Not portable.

Q.5 a. What is the output of the following program? Explain.

```
 main()
{
int a, b, c, d;

a = 15;
b = 10;
c = ++a - b;

printf("a = %d  b = %d  c = %d\n",a, b, c);

d = b++ +a;
            printf("a = %d  b = %d  d = %d\n",a, b, d);
```

```
                    printf("a/b = %d\n", a/b);
                            printf("a%%b = %d\n", a%b);
          printf("a *= b = %d\n", a*=b);                \
          printf("%d\n", (c>d) ? 1 : 0);
          printf("%d\n", (c<d) ? 1 : 0);
          }
```

Answer:

Output:

```
a = 16  b = 10  c = 6
a = 16  b = 11  d = 26
a/b = 1
a%b = 5
a *= b = 176
0
  1
```

b. Write a C program that requests the user to enter a character and displays a message on the screen telling the user whether the character is an alphabet or digit, or any other special character.

Answer:

```
#include  <stdio.h>
  #include  <ctype.h>
  main()
  {
     char character;
     printf("Press any key\n");
     character = getchar();
     if (isalpha(character) > 0)
        printf("The character is a letter.");
     else
        if (isdigit (character) > 0)
           printf("The character is a digit.");
        else
           printf("The character is not alphanumeric.");
  }
```

Q.6 a. Write a program using while loop to evaluate the equation $y = x^n$ where n is a non-negative integer.

Answer:

```
 main()
   {
      int count, n;
      float x, y;

      printf("Enter the values of x and n : ");
      scanf("%f %d", &x, &n);
      y = 1.0;
      count = 1;                 /* Initialisation */

      /* LOOP BEGINs */

      while ( count <= n)      /* Testing */
      {
          y = y*x;
          count++;           /* Incrementing */
      }
      /* END OF LOOP */
      printf("\nx = %f; n = %d; x to power n = %f\n",x,n,y);
   }
```

---

b. Illustrates the use of the goto statement by implementing a C program that evaluates the square root for five numbers. The variable count keeps the count of numbers read. When count is less than or equal to 5, goto read; directs the control to the label read; otherwise, the program prints a message and stops.

Answer:

```
#include  <math.h>
   main()
   {
       double x, y;
       int count;

       count = 1;

       printf("Enter FIVE real values in a LINE \n");
   read:
       scanf("%lf", &x);
```

```
     printf("\n");
     if (x < 0)
       printf("Value - %d is negative\n",count);
     else
     {
       y = sqrt(x);
       printf("%lf\t %lf\n", x, y);
     0}
     count = count + 1;

     if (count <= 5)
  goto read;
     printf("\nEnd of computation");
}
```

Q.7 a. The names of employees of an organization are stored in three arrays, namely first_name, second_name, and last_name. Write a program to concatenate the three parts into one string to be called name without using standard string concatenation function.

Answer:

```
 main()
  {
     int  i, j, k ;
     char   first_name[10] = {"VISWANATH"}  ;
     char  second_name[10] = {"PRATAP"} ;
     char    last_name[10] = {"SINGH"} ;
     char   name[30] ;

   /* Copy first_name into name */

     for( i = 0 ; first_name[i] != '\0' ; i++ )
        name[i] = first_name[i] ;

   /* End first_name with a space */

     name[i] = ' ' ;

   /* Copy second_name into name */

     for( j = 0 ; second_name[j] != '\0' ; j++ )
        name[i+j+1] = second_name[j] ;

   /* End second_name with a space */

     name[i+j+1] = ' ' ;

   /* Copy last_name into name */
```

```
    for( k = 0 ; last_name[k] != '\0'; k++ )
      name[i+j+k+2] = last_name[k] ;

  /* End name with a null character */

    name[i+j+k+2] = '\0' ;

    printf("\n\n") ;
    printf("%s\n", name) ;
}
```

Q.8 a. Write a C program using functions to calculate the standard deviation and mean of an array of values. The array elements are read from the terminal.

Answer:

```
#include     <math.h>
  #define SIZE    5
  float std_dev(float a[], int n);
  float mean (float a[], int n);
  main( )
  {
      float value[SIZE];
      int i;

      printf("Enter %d float values\n", SIZE);
      for (i=0 ;i < SIZE ; i++)
          scanf("%f", &value[i]);
      printf("Std.deviation is %f\n", std_dev(value,SIZE));
  }

  float std_dev(float a[], int n)

  {   int i;
      float x, sum = 0.0;
      x = mean (a,n);
      for(i=0; i < n; i++)
          sum += (x-a[i])*(x-a[i]);
      return(sqrt(sum/(float)n));
  }

  float mean(float a[],int n)

  {
      int i ;
      float sum = 0.0;
      for(i=0 ; i < n ; i++)
         sum = sum + a[i];
         return(sum/(float)n);
}
```

b. Explain function and function prototype. State advantages of using functions in a C program.

Answer:
A **function** is a named, independent section of C code that performs a specific task and optionally returns a value to the calling program. A function is named, each have a unique name. By using that name in another part of the program, one can execute the statements contained in the function.
The prototype for a function is identical to the function header, with a semicolon added at the end. Like the function header, the function prototype includes information about the function's return type, name, and parameters. The prototype's job is to tell the compiler about the function's return type, name, and parameters. With this information, the compiler can check every time when the function is called to verify that programmer is passing the correct number and type of arguments to the function and using the return value correctly. If there's a mismatch, the compiler generates an error message.

Advantages of using functions in C code:
1. A programmer may have a block of code that he has repeated forty times throughout the program. A function to execute that code would save a great deal of space, and it would also make the program more readable.

2. It is easy to locate and isolate a faulty function. Having only one copy of the code makes it easier to make changes.

3. Another reason for functions is to break down a complex program into logical parts. For example, take a menu program that runs complex code when a menu choice is selected. The program would probably best be served by making functions for each of the actual menu choices, and then breaking down the complex tasks into smaller, more manageable tasks, which could be in their own functions. In this way, a program can be designed that makes sense when read. And has a structure that is easier to understand quickly. The worst programs usually only have the required function, main, and fill it with pages of jumbled code.

4. A function may be used by many other programs. A programmer can use already compiled function instead of starting over from scratch.

Q.9 a. Write a program using pointers to compute the sum of all elements stored in an array.

Answer:

```
main()
  {
     int *p, sum, i;
     int x[5] = {5,9,6,3,7};
```

```
    i  = 0;
    p  = x;              /* initializing with base address of x */
    printf("Element   Value   Address\n\n");
    while(i < 5)
    {
      printf(" x[%d] %d %u\n", i, *p, p);
      sum = sum + *p;   /* accessing array element  */
      i++, p++;        /* incrementing pointer     */
    }
    printf("\n  Sum    =  %d\n", sum);
    printf("\n  &x[0]  =  %u\n", &x[0]);
    printf("\n  p      =  %u\n", p);
 }
```

b. Write a program to read data from the keyboard, write it to a file called INPUT, again read the same data from the INPUT file, and display it on the screen.

Answer:

```
#include  <stdio.h>

  main()
  {
    FILE *f1;
    char c;
    printf("Data Input\n\n");
        /* Open the file INPUT */
    f1 = fopen("INPUT", "w");

        /* Get a character from keyboard   */
    while((c=getchar()) != EOF)

          /* Write a character to INPUT  */
      putc(c,f1);
    /* Close the file INPUT   */
    fclose(f1);
    printf("\nData Output\n\n");
    /* Reopen the file INPUT    */
    f1 = fopen ("INPUT", "r");
    /* Read a character from INPUT*/
     while((c=getc(f1)) != EOF)

            /* Display a character on screen */
         printf("%c",c);

         /* Close the file INPUT       */
     fclose(f1);
}
```

Text Books

(1) **Fundamentals of computers, V. Rajaraman , 5th Edition, P H I, 2011**

(2) **Programming in ANSI C , E, Balagurusamy , 5th Edition, Tata Mc Graw hill, 2011**