

**Q.2 a. With the help of a block diagram, explain the basic parts and structure of a computer system.**

**Answer:**

In Computer System, different parts of a computer are organized in such a way that, it helps to perform various operations to complete the given specific task. There are five basic operations. They are:

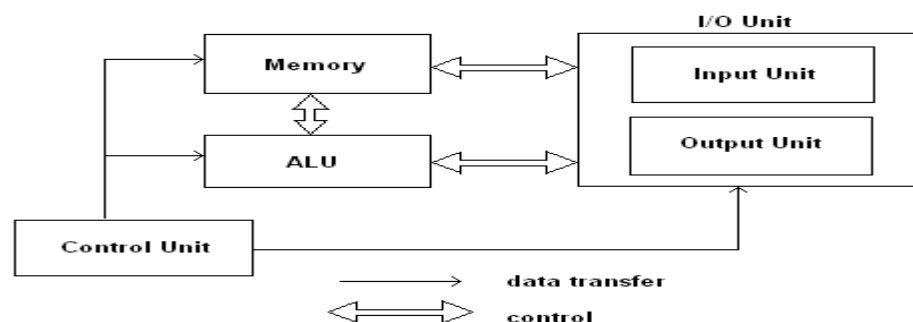
1. Inputting
2. Outputting
3. Control unit
4. Storage unit
5. Processing

1. **INPUT:** This operation is used to feed the information in the computer. The standard devices are keyboard, Mouse-a pointing device, and card redirect. The input devices must accept the data from the outside world and the computer to process it must accept the same data. The data or information feuded through the keyboard are stored in the storage device.

2. **OUTPUT:** This operation is used to display the fed data or the processed data. Some standard output devices are monitor or screen, printer, etc. These output devices must accept the data, which was processed by the processor. The processing is done binary format and it must be converted to understandable form.

3. **CONTROL UNIT:** This unit is used to control all the devices, which is helpful for processing. It controls the inflow and outflow of data. It works like a traffic cop, which controls the movement of data from memory to processing unit. We can also say it as central nervous system of the computer.

4. **STORAGE UNIT:** Storage unit is to store any kind of information. Whatever the data inserted or feeded through keyboard is first stored in the memory for further processing. It must store the intermediate results and also final result. The memory in the storage unit is divided in the form of cells. Each and every cell has its address. Memory is divided in two types: RAM & ROM



**Q.2 b. What is the purpose of an output device? Explain any two most commonly used output devices.**

**Answer:**

The data, processed by the CPU is made available to the end user by the output devices. Explain briefly any two of the following:  
Monitor, Printer, Speaker, Plotter

**Q.3 a. Distinguish between single-user and multi-user operating system. Give an example of each.**

**Answer:**

Single-user operating system is an operating system, where only one user can use the computer at a time. They do not maintain log of the system usage. They also do not support security aspects, such as login and password procedures. Ex: MS-DOS operating system is a user friendly, hierarchical directory structure for organizing and storing files.

Multi-user operating system is an operating system, where more than one user can use the computer simultaneously. They maintain an extensive log of the system usage. They also support login and password procedures to restrict the access and to provide security to the system. Ex: UNIX operating system is a complex operating system containing more than 200 utility programs that can be used by the programmers to perform useful functions.

**Q.3 b. What is a computer network? Describe different types of computer networks in use today.**

**Answer:**

By computer network we mean an interconnected set of autonomous computers. The term autonomous implies that the computers can function independent of others. However, these computers can exchange information with each other through the communication network system.

Different type os computer networks are as follows:

#### **Local Area Network (LAN)**

LAN is usually privately owned and links the devices in a single office, building or campus of up to few kilometres 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.

LANs are restricted in size, which means that their worst-case transmission time is bounded and known in advance. Hence this is more reliable as compared to MAN and WAN. Knowing this bound makes it possible to use

certain kinds of design that would not otherwise be possible. It also simplifies network management.

### **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*.

### **The Internet**

Internet is a collection of networks or network of networks. Various networks such as LAN and WAN connected through suitable hardware and software to work in a seamless manner. It allows various applications such as e-mail; file transfer, remote log-in, World Wide Web, Multimedia, etc run across the internet. The basic difference between WAN and Internet is that WAN is owned by a single organization while internet is not so. But with the time the line between WAN and Internet is shrinking, and these terms are sometimes used interchangeably.

**Q.4 a. Describe the basic structure of a C program. Why do we need to use comments in programs?**

**Answer:**

Structure of C program is defined by set of rules called protocol, to be followed by programmer while writing C program. All C programs are having sections/parts which are mentioned below.

- Documentation section
- Link Section
- Definition Section
- Global declaration section
- Function prototype declaration section
- Main function
- User defined function definition section

(briefly explain each part)

Comments in a C Program are used to mark:

1. Code Logic
2. They are necessary when some other programmer later work on it. They will also be helpful when you want to work again on the program after one or two years when you forgot that what code perform what task.
3. They are very helpful in understanding large programmes and program flow.

#### Q.4 b. How do we implement multiple source C program files in UNIX?

##### Answer:

There are two possible ways to compile a multi-source C program. The first is to use a single command line to compile all the files. Suppose that we have a program whose source is found in files main.c, a.c and b.c. We could compile it this way:

```
% gcc -o hello_world main.c a.c b.c
```

This will cause the compiler to compile each of the given files separately, and then link them all together to one executable file named "hello\_world". Two comments about this program:

- If we define a function (or a variable) in one file, and try to access them from a second file, we need to declare them as external symbols in that second file. This is done using the C "extern" keyword.
- The order of presenting the source files on the command line may be altered. The compiler (actually, the linker) will know how to take the relevant code from each file into the final program, even if the first source file tries to use a function defined in the second or third source file.

The problem with this way of compilation is that even if we only make a change in one of the source files, all of them will be re-compiled when we run the compiler again. In order to overcome this limitation, we could divide the compilation process into two phases - compiling, and linking. Let's first see how this is done, and then explain:

```
% gcc -c main.c
% gcc -c a.c
% gcc -c b.c
% gcc -o hello_world main.o a.o b.o
```

The first 3 commands have each taken one source file, and compiled it into something called "object file", with the same names, but with a '.o' suffix. It is the '-c' flag that tells the compiler only to create an object file, and not to generate a final executable file just yet. The object file contains the code for the source file in machine language,

but with some unresolved symbols. After creating the 3 object files, we use the 4th command to link the 3 object files into one program. The linker (which is invoked by the compiler now) takes all the symbols from the 3 object files, and links them together. Furthermore, the linker also links the standard C library into the program, in this case, to resolve the "printf" symbol properly.

To see why this complexity actually helps us, we should note that normally the linking phase is much faster than the compilation phase. This is especially true when doing optimizations, since that step is done before linking. Now, let's assume we change the source file "a.c", and we want to re-compile the program. We'll only need now two commands:

```
% gcc -c a.c
% gcc -o hello_world main.o a.o b.o
```

**Q.5 a. What do you mean by identifiers? How are these different from keywords?**

**Answer:**

Each C word can be classified as either a keyword or an identifier. Identifiers refer to the names of variables, functions and arrays. These are user-defined names and consist of a sequence of letters and digits, with a letter as a first character. Both uppercase and lowercase can be used to form identifiers. Maximum length of an identifier is 8 characters. Some compiler allows length upto 40 characters. Comma and blanks are not allowed. No special symbol except underscore is allowed in identifiers name.

All keywords have fixed meanings and these cannot be changed. Keywords are the basic building blocks for program statement. The keywords also known as reserved words cannot be used as variable names. There are 32 keywords available in C.

**Q.5 b. List some of the general points to keep in mind while writing a scanf statement.**

**Answer:**

Some of the general points to keep in mind while writing a scanf statement

1. All function arguments, except the control string, must be pointers to variables.
2. Format specifications contained in the control string should match the arguments in order.
3. Input data items must be separated by spaces and must match the variables receiving the input in the same order.
4. The reading will be terminated, when scanf encounter an invalid mismatch of data or a character that is not valid for the value being read.
5. When searching for a value, scanf ignores line boundaries and simply looks for the next appropriate character.

6. Any unread data items in a line will be considered as a part of the data input line to the next scanf call.
7. When the field width specifier w is used, it should be large enough to contain the input data size.

- Q.5 c. Write a program to read three integers from the keyboard using one scanf statement and output them on one line using.**
- (i) three printf with statements**
  - (ii) only one printf without conversion specifiers, and**
  - (iii) only on printf without conversion specifiers**

**Answer:**

Refer section 5.5, Page Number 5.13, Computer Concepts & Programming in C by E Balagurusamy-3<sup>rd</sup> Edition. TATA McGRAW- HILL

- Q.6 a. Explain any four bitwise operators available in 'C' language using an example of each category.**

**Answer:**

Four different bitwise operators available in C language are as follows:

(i) **One's complement:** denoted by symbol ~ operates on a single variable. On applying this operator on a number, all 1's present in the binary equivalent of that number is changed to 0 and

all 0's are changed to 1's. For example:

x=1001 0110 1100 1011

~x=0110 1001 0011 0100

(ii) **Right Shift Operator:** represented by symbol >>. It shifts each bits in the operand to the right. The number of places the bits are shifted to right depends on the number following the operand. For example x>>3 would shift all bits three places to the right. For example suppose

x=0100 1001 1100 1011

x>>3=0000 1001 0011 1001

(iii) **Left Shift Operator:** represented by symbol <<. It shifts each bits in the operand to the left. The number of places the bits are shifted to left depends on the number following the operand. For example x<<3 would shift all bits three places to the left.

x<<3 for above x would be 0100 1110 0101 1000

(iv) **Exclusive OR:** It is a binary operator represented by ^. The result of exclusive OR is 1 if

only one of the bits is 1; otherwise it is 0. For example:

X=0000 0000 0000 1101

Y=0000 0000 0001 1001

100

X^Y=0000 0000 0001 0100

**Q.6 b. (i) What is the output of the following?**

```
int main()
{ int a=1,b=2,c=3,d=4,e;
  e=c+d=b*a;
  printf(“%d,%d\n”,e,d);
}
```

**(ii) What is the output of the following:**

```
Int a b,c,d;
main(){
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);
}
```

**Answer:**

(i) syntax error

(ii) a =16, b=10, c=6

A=16, b=11, d=26

Page No 6.9 Computer Concepts & Programming in C by E Balagurusamy -  
TATA McGRRAW- HILL

**Q.7 a. Write a function power that computes x raised to the power y for integers x and y and returns double-type value.**

**Answer:**

```
main()
{ int x,y; /*input data */
  double power(int, int); /* prototype declaration*/
  printf(“Enter x,y:”);
  scanf(“%d %d” , &x,&y);
  printf(“%d to power %d is %f\n”, x,y,power (x,y));
}
double power (int x, int y);
{
  double p;
  p = 1.0 ; /* x to power zero */
  if(y >=0)
```

```
        while(y--) /* computes positive powers */
            p *= x;
    else
        while (y++) /* computes negative powers */
            p /= x;
    return(p);
}
```

**Q.7 b. What do you mean by global variable. Write a multifunction program to illustrate the properties of global variables.**

**Answer:**

**Program**

```
int fun1(void);
int fun2(void);
int fun3(void);
int x ;                               /* global */
main()
{
    x = 10 ; /* global x */
    printf("x = %d\n", x);
    printf("x = %d\n", fun1());
    printf("x = %d\n", fun2());
    printf("x = %d\n", fun3());
}
fun1(void)
{
    x = x + 10 ;
}
int fun2(void)
{
    int x ;          /* local */
    x = 1 ;
    return (x);
}
fun3(void)
{
    x = x + 10 ;    /* global x */
}
```



**Q.8 a. Explain condition operator and its syntax using an example.**

**Answer:**

The conditional operator in C 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.

**condition? result1: result2;**

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

**Q.8 b. An electric power distribution company charges its domestic consumers as follows:**

<i>Consumption Units</i>	<i>Rate of Charge</i>
<b>0 - 200</b>	<b>Rs. 0.50 per unit</b>
<b>201 - 400</b>	<b>Rs. 100 plus Rs.0.65 per unit excess of 200</b>
<b>401 - 600</b>	<b>Rs. 230 plus Rs.0.80 per unit excess of 400</b>
<b>601 and above</b>	<b>Rs. 390 plus Rs.1.00 per unit excess of 600</b>

**Write a C program that reads the customer number and power consumed and prints the amount to be paid by the customer**

**Answer:**

```
main()
{
    int units, custnum;
    float charges;

    printf("Enter CUSTOMER NO. and UNITS consumed\n");
    scanf("%d %d", &custnum, &units);

    if (units <= 200)
        charges = 0.5 * units;
    else if (units <= 400)
        charges = 100 + 0.65 * (units - 200);
    else if (units <= 600)
        charges = 230 + 0.8 * (units - 400);
    else
        charges = 390 + (units - 600);

    printf("\n\nCustomer No: %d: Charges = %.2f\n",
        custnum, charges);
}
```

**Q.9 a. Write a C program that reads a string from keyboard and determines whether the string is palindrome or not. (A string is palindrome if it is read from left or right gives you the same string).**

**Answer:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j,k,flag=1;
char a[20];
clrscr();
printf("enter any word:\n");
scanf("%s",a);
i=0;
while(a[i]!='\0')
i++;
k=i;
for(j=0;j<k/2;)
{
if(a[j]!=a[i-1])
{
flag=0;
break;
}
if(a[j]==a[i-1])
{
j++;
i--;
}
}
if(flag==0)
printf("it is a not palindrome");
else
printf("it is a palindrome");
getch();
}
```

**Q.9 b. Write a C program to compute and print a multiplication table for numbers 1 to 5 as shown below:**

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>1</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>2</b>	<b>2</b>	<b>4</b>	<b>6</b>	<b>8</b>	<b>10</b>
<b>3</b>	<b>3</b>	<b>6</b>	.	.	.
<b>4</b>	<b>4</b>	<b>8</b>	.	.	.
<b>5</b>	<b>5</b>	<b>10</b>	.	.	<b>25</b>

**Answer:**

**Program:**

```
#define ROWS 5
#define COLUMNS 5

main()
{
    int row, column, product[ROWS][COLUMNS];
    int i, j;

    printf(" MULTIPLICATION TABLE\n\n");
    printf(" ");

    for( j = 1 ; j <= COLUMNS ; j++ )
        printf("%4d" , j );
    printf("\n");
    printf("-----\n");
    for( i = 0 ; i < ROWS ; i++ )
    {
        row = i + 1 ;
        printf("%2d |", row);
        for( j = 1 ; j <= COLUMNS ; j++ )
        {
            column = j ;
            product[i][j] = row * column ;
            printf("%4d", product[i][j] );
        }
        printf("\n");
    }
}
```

**Text Book**

**Computer Concepts & Programming in C, E.Balagurusamy, Tata McGraw-Hill, 2010**