Q.2 a. Find the hexadecimal equivalent of:
(i) $(0.5625)_{10}$
(ii) $(0.3)_{10}$

Answer: Page Number 26 of Text Book

b. Give the difference between micro, mini and mainframe & super computers

Answer:

Ans: 1. **Microcomputer:** Microcomputer is at the lowest end of the computer range in terms of speed and storage capacity. Its CPU is a microprocessor. The first microcomputers were built of 8-bit microprocessor chips. The most common application of personal computers (PC) is in this category. The PC supports a number of input and output devices. An improvement of 8-bit chip is 16-bit and 32-bit chips. Examples of microcomputer are IBM PC, PC-AT .

2. **Mini Computer**: This is designed to support more than one user at a time. It possesses large storage capacity and operates at a higher speed. The mini computer is used in multi-user system in which various users can work at the same time. This type of computer is generally used for processing large volume of data in an organisation. They are also used as servers in Local Area Networks (LAN).

3. **Mainframes:** These types of computers are generally 32-bit microprocessors. They operate at very high speed, have very large storage capacity and can handle the work load of many users. They are generally used in centralised databases. They are also used as controlling nodes in Wide Area Networks (WAN). Example of mainframes are DEC, ICL and IBM 3000 series.

4. **Supercomputer:** They are the fastest and most expensive machines. They have high processing speed compared to other computers. They have also multiprocessing technique. One of the ways in which supercomputers are built is by interconnecting hundreds of microprocessors. Supercomputers are mainly being used for whether forecasting, biomedical research, remote sensing, aircraft design and other areas of science and technology. Examples of supercomputers are CRAY YMP, CRAY2, NEC SX-3, CRAY XMP and PARAM from India.

c. Draw a block diagram to illustrate the basic organization of computer system and explain the function of various units.

Answer:

Ans: A computer as shown in Fig. 1 performs basically five major operations or functions irrespective of their size and make. These 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, and 5) it controls all operations inside a computer. We discuss below each of these operations.

1. Input: This is the process of entering data and programs in to the computer system. You should know that computer is an electronic machine like any other machine which takes as inputs raw data and performs some processing giving out processed data. Therefore, the input unit takes data from us to the computer in an organized manner for processing.

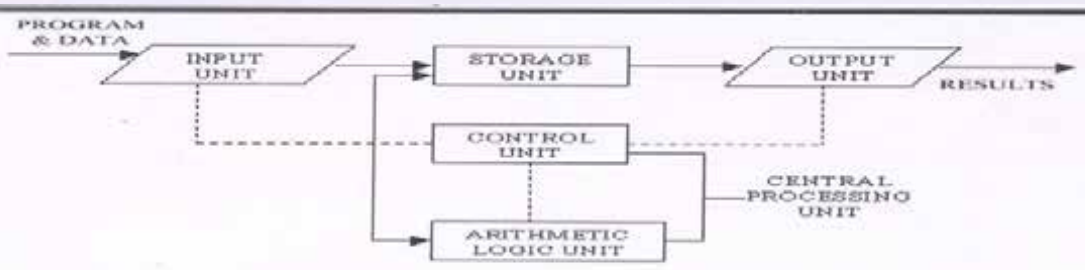

Fig. 1 Basic computer Operations

2. Storage: The process of saving data and instructions permanently is known as storag Data has to be fed into the system before the actual processing starts. It is because the processing speed of Central Processing Unit (CPU) is so fast that the data has to be provided to CPU with the same speed. Therefore the data is first stored in the storage unit for faster access and processing. This storage unit or the primary storage of the computer system is designed to do the above functionality. It provides space for storing data and instructions.
The storage unit performs the following major functions:

All data and instructions are stored here before and after processing.
Intermediate results of processing are also stored here.

3. Processing: The task of performing operations like arithmetic and logical operations is called processing. The Central Processing Unit (CPU) takes data and instructions from the storage unit and makes all sorts of calculations based on the instructions given and the type of data provided. It is then sent back to the storage unit.

4. Output: This is the process of producing results from the data for getting useful information. Similarly the output produced by the computer after processing must also be kept somewhere inside the computer before being given to you in human readable form. Again the output is also stored inside the computer for further processing.

5. Control: The manner how instructions are executed and the above operations are performed. Controlling of all operations like input, processing and output are performed by control unit. It takes care of step by step processing of all operations in side the computer.

FUNCTIONAL UNITS

In order to carry out the operations mentioned in the previous section the computer allocates the task between its various functional units. The computer system is divided into three separate units for its operation. They are 1) arithmetic logical unit, 2) control unit, and 3) central processing unit.

i) Arithmetic Logical Unit (ALU)

After you enter data through the input device it is stored in the primary storage unit. The actual processing of the data and instruction are performed by Arithmetic Logical Unit. The major operations performed by the ALU are addition, subtraction, multiplication, division, logic and comparison. Data is transferred to ALU from storage unit when required. After processing the output is returned back to storage unit for further processing or getting stored.

ii) Control Unit (CU)

The next component of computer is the Control Unit, which acts like the supervisor seeing that things are done in proper fashion. The control unit determines the sequence in which computer programs and instructions are executed. Things like processing of programs stored in the main memory, interpretation of the instructions and issuing of signals for other units of the computer to execute them. It also acts as a switch board operator when several users access the computer simultaneously. Thereby it coordinates the activities of computer's peripheral equipment as they perform the input and output. Therefore it is the manager of all operations mentioned in the previous section.

iii) Central Processing Unit (CPU)

The ALU and the CU of a computer system are jointly known as the central processing unit. You may call CPU as the brain of any computer system. It is just like brain that takes all major decisions, makes all sorts of calculations and directs different parts of the computer functions by activating and controlling the operations.

Q.3 a. Discuss various types of high level languages.

Answer:

Ans: Many languages have been developed for achieving different variety of tasks, some are fairly specialized others are quite general purpose.

These are categorized according to their use as

a) Algebraic Formula-Type Processing. These languages are oriented towards the computational procedures for solving mathematical and statistical problem

Examples are

- **BASIC (Beginners All Purpose Symbolic Instruction Code).**
- **FORTRAN (Formula Translation).**
- **PL/I (Programming Language, Version 1).**
- **ALGOL (Algorithmic Language).**
- **APL (A Programming Language).**

b) Business Data Processing:

- These languages emphasize their capabilities for maintaining data processing procedures and files handling problems. Examples are:
- **COBOL (Common Business Oriented Language).**
- **RPG (Report Program Generator**

(c) **b) String and List Processing:** These are used for string manipulation including search for patterns, inserting and deleting characters. Examples are:

- **LISP (List Processing).**
- **Prolog (Program in Logic).**

(d) **Object Oriented Programming Language**

In OOP, the computer program is divided into objects. Examples are:

- **C++**
- **Java**

(e) **e) Visual programming language:** these are designed for building Windows-based applications Examples are:

- **Visual Basic**
- **Visual Java**
- **Visual C**

b. Define the various types of common services provided by the operating system.

Answer:

Ans: The common service provided by the operating system is listed below.

1. Program execution

2. I/O operation

3. File system manipulation

4. Communications

5. Error detection

1. **Program execution**: Operating system loads a program into memory and executes the program. The program must be able to end its execution, either normally or abnormally.

2. **I/O Operation** : I/O means any file or any specific I/O device. Program may require any I/O device while running. So operating system must provide the required I/O.

3. **File system manipulation** : Program needs to read a file or write a file. The operating system gives the permission to the program for operation on file.

4. **Communication** : Data transfer between two processes is required for some time. The both processes are on the one computer or on different computer but connected through computer network. Communication may be implemented by two methods:

a. Shared memory

b. Message passing.

5. **Error detection** : error may occur in CPU, in I/O devices or in the memory hardware. The operating system constantly needs to be aware of possible errors. It should take the appropriate action to ensure correct and consistent computing.

Operating system with multiple users provides following services.

1. Resource Allocation

2. Accounting

3. Protection

c. How Multitasking is different from multiprogramming?

Answer:

Ans: Time sharing, or multitasking, is a logical extension of multiprogramming. Multiple jobs are executed by the CPU switching between them, but the switches occur so frequently that the users may interact with each program while it is running.
When two or more programs are in memory at the same time, sharing the processor is referred to the **multiprogramming operating system.**

Q.4 a. How LAN, WAN and MAN are differ from each other?

Answer:

Ans: A **LAN** (local area network) is a group of computers and network devices connected together, usually within the same building. By definition, the connections must be high speed and relatively inexpensive (e.g., token ring or Ethernet). Most Indiana University Bloomington departments are on LANs.

A LAN connection is a high-speed connection to a LAN. On the IUB campus, most connections are either Ethernet (10Mbps) or Fast Ethernet (100Mbps), and a few locations have Gigabit Ethernet (1000Mbps) connections.

A **MAN** (metropolitan area network) is a larger network that usually spans several buildings in the same city or town. The IUB network is an example of a MAN.

A **WAN** (wide area network), in comparison to a MAN, is not restricted to a geographical location, although it might be confined within the bounds of a state or country. A WAN connects several LANs, and may be limited to an enterprise (a corporation or an organization) or accessible to the public. The technology is high speed and relatively expensive. The Internet is an example of a worldwide public WAN.

b. What is the difference between internet and World Wide Web (WWW)?

Answer:

Ans: The Internet is a massive network of networks, a networking infrastructure. It connects millions of computers together globally, forming a network in which any computer can communicate with any other computer as long as they are both connected to the Internet. Information that travels over the Internet does so via a variety of languages known as protocols.

World Wide Web, or simply Web, is a way of accessing information over the medium of the Internet. It is an information-sharing model that is built on top of the Internet. The Web uses the HTTP protocol, only one of the languages spoken over the Internet, to transmit data. Web services, which use HTTP to allow applications to communicate in order to exchange business logic, use the the Web to share information. The Web also utilizes browsers, such as Internet Explorer or Firefox, to access Web documents called Web pages that are linked to each other via hyperlinks. Web documents also contain graphics, sounds, text and video.

The Web is just one of the ways that information can be disseminated over the Internet. The Internet, not the Web, is also used for e-mail, which relies on SMTP, Usenet news groups, instant messaging and FTP. So the Web is just a portion of the Internet, albeit a large portion, but the two terms are not synonymous and should not be confused.

Q.5 a. What is keyword and identifier? Give the rules for identifier.

Answer:

Ans. : Keywords : Keywords have fixed meanings and these meanings can not be changed. There are 32 keywords. Some compiler may use additional keywords that must be identified from the C manual. Keywords serve as basic building block for a program statement.

Such as, **auto**, **break**, **double** etc. All keyword must be written in lowercase.

Identifiers : The names of variables, functions and arrays are **identifiers**. These are user-defined names and consist of a sequence of letters and digits. Such as, my_num, _ton etc.

Rules for identifier :

(i) Must consist of only letters, digits and underscores.
(ii) First character must be an alphabet or underscore.
(iii) Only first 31 characters are significant.
(iv) Can not use a keyword.
(v) Must not contain white spaces.

b. Write about various data types of C.

Answer:

Ans. : Data types : C language is rich in its data types. ANSI supports three classes of data types.

(i) Primary or fundamental data types.
(ii) Derived data types.
(iii) User-defined data types.

All C compilers support five fundamental data types. Namely, integer (**int**), character (**char**), floating point (**float**), double-precision floating point (**double**) and **void.** They are described below.

Integer types : Integer are whole numbers with a range of values supported by a particular machine. If we use a 16 bit word length, the size of the integer value is limited to the range -32768 to +32767.

C has three classes of integer storage, namely **short int**, **int** and **long int. Short int** represents fairly small value than **int** and **int** represents fairly small value than **long int.**

Floating point type : Floating point numbers are stored in 32 bits, with 6 digits of precision. Floating point numbers are defined in C by the keyword **float**. When **float** is not sufficient **double** can be used and when **double** is not sufficient **long double** can be used.

Void types : The **void** types has no values. This is usually used to specify the type of functions. The type of a function is said to be **void** when it does not return any values to the calling function.

Character types : A single character can be defined as a character (**char**) type data. Characters are usually stored in 8 bits of internal storage. While **unsigned char** have values between 0 to 255, **sign char** have values from -128 to 127.

c. What are the rules for + + and - - operators? Discuss briefly.

Answer: Page Number 60 of Text Book

Q.6 a. Describe nesting of IF.... ELSE statements with the help of example.

Answer: Page Number 122-123 of Text Book

b. Describe the switch statement with the help of an example.

Answer:

Ans: The **switch** and **case** statements help control complex conditional and branching operations. The **switch** statement transfers control to a statement within its body.

Syntax

selection-statement:

switch (*expression*)*statement*

labeled-statement:

case *constant-expression* : *statement*

default : *statement*

Control passes to the statement whose **case** *constant-expression* matches the value of **switch** (*expression*). The **switch** statement can include any number of **case** instances, but no two case constants within the same **switch** statement can have the same value. Execution of the statement body begins at the selected statement and proceeds until the end of the body or until a **break** statement transfers control out of the body.

Use of the **switch** statement usually looks something like this:

```
switch ( expression )
{
  declarations
  .
  .
  .
  case constant-expression :
    statements executed if the expression equals the
    value of this constant-expression
    .
    .
    .
    break;
  default :
    statements executed if expression does not equal
    any case constant-expression
}
```

For example:

```
switch( c )
{
   case 'A':
      capa++;
   case 'a':
      lettera++;
   default :
      total++;
}
```

c. Give the difference between do-while and while loop.

Answer:

Ans: The difference between a "do ...while" loop and a "while {} " loop is that the while loop tests its condition before execution of the contents of the loop begins; the "do" loop tests its condition after it's been executed at least once. As noted above, if the test condition is false as the while loop is entered the block of code is never executed. Since the condition is tested at the bottom of a do loop, its block of code is always executed at least once

Q.7 a. Write a Program to reverse ANY given number in C.

Answer:

Ans:

```
#include <stdio.h>
main()
{
        int num,mod,rev=0;
        printf("Enter a number:");
scanf("%d", &num);
while(num>0)
{
mod=num%10;
rev=(rev*10)+mod;
num=num/10;
}
printf("Reverse of the given number: %d", rev);
getchar();
}
```

b. Do array subscripts always start with zero?

Answer:

Ans: Yes. If you have an array a[MAX] (in which MAX is some value known at compile time), the first element is a[0], and the last element is a[MAX-1]. This arrangement is different from what you would find in some other languages. In some languages, such as some versions of BASIC, the elements would be a[1] through a[MAX], and in other languages, such as Pascal, you can have it either way.

This variance can lead to some confusion. The "first element" in non-technical terms is the "zero'th" element according to its array index. If you're using spoken words, use "first" as the opposite of "last." If that's not precise enough, use pseudo-C. You might say, "The elements a sub one through a sub eight," or, "The second through ninth elements of a."

There's something you can do to try to fake array subscripts that start with one. Don't do it. The technique is described here only so that you'll know why not to use it.

Because pointers and arrays are almost identical, you might consider creating a pointer that would refer to the same elements as an array but would use indices that start with one.

c. How to compare two strings without using strcmp?

Answer:

```
Ans:  #include<stdio.h>
void main()
{
int a=compare("forgetcode","FORGETCODE");
if(a===0)
{
}else
{
}
}
int compare(char a[], char b[])
{
  int c = 0;

  while( a[c] == b[c] )
  {
    if( a[c] == '\0' || b[c] == '\0' )
      break;
    c++;
  }
  if( a[c] == '\0' && b[c] == '\0' )
    return 0;
  else
    return -1;
}
```

Q.8 a. What is the need for user-defined functions?

Answer: Page Number 262-263 of Text Book II

b. Write down the advantages for using user-defined functions in 'C' language.

Answer: Page Number 263 of Text Book

c. Write a C program to add two integers. Make a function add to add integers and display sum in main() function.

Answer:

```
Ans:  #include <stdio.h>
int add(int a, int b);          //function prototype(declaration)
int main(){
    int num1,num2,sum;
    printf("Enters two number to add\n");
    scanf("%d %d",&num1,&num2);
    sum=add(num1,num2);         //function call
    printf("sum=%d",sum);
    return 0;
}
int add(int a,int b)            //function declarator
{
/* Start of function definition. */
    int add;
    add=a+b;
    return add;                 //return statement of function
/* End of function definition. */
}
```

Q.9 a. What is pointer and how to use pointers in C? Also discuss about nu pointers.

Answer:

Ans: A **pointer** is a variable whose value is the address of another variable ie. direct address of the memory location. Like any variable or constant, you must declare a pointer before you can use it to store any variable address. The general form of a pointer variable declaration is:

```
type *var-name
```

There are few important operations which we will do with the help of pointers very frequently. (a) we define a pointer variables (b) assign the address of a variable to a pointer and (c) finally access the value at the address available in the pointer variable. This is done by using unary operator * that returns the value of the variable located at the address specified by its operand.

It is always a good practice to assign a NULL value to a pointer variable in case you do not have exact address to be assigned. This is done at the time of variable declaration. A pointer that is assigned NULL is called a **null** pointer.

The NULL pointer is a constant with a value of zero defined in several standard libraries. Consider the following program:

```
#include <stdio.h>

int main ()
{
  int  *ptr = NULL;

  printf("The value of ptr is : %x\n", ptr  );

  return 0;
}
```

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

```
The value of ptr is 0
```

b. Discuss the following functions:-
(i) getc (ii) fscanf (iii) put w (iv) fprintf

Answer: Page Number 392 of Text Book II

TEXT BOOKS

1. Fundamentals of Computers, V. Rajaraman, Fourth Edition, PHI, 2007

2. Programming in ANSI C, E. Balagurusamy, Third Edition, Tata McGraw Hill