Q.2 a. Write the basic characteristics of an algorithm. (5)

Answer:

2 (a) The basic characteristics of an algorithm.

1) An algorithm begins with instruction to accept inputs. These inputs are processed by subsequent instructions in the algorithm.

2. The processing rules specified in the algorithm must be precise and unambiguous. (unambiguous)

3. Each instruction must be sufficiently basic such that it can, in principle, be carried out by a person with paper & pencil.

4. The total time to carry out all the steps in the algorithm must be finite.

5. An algorithm must produce one or more outputs.

b. Convert the following: (6)

(i) $(36)_{10}$ to binary

(ii) $(23)_{10}$ to Hexadecimal

(iii) $(D6C1)_{16}$ to decimal

Answer:

(b) (i). $(36)_{10}$ to binary.

2	36	
2	18	– 0 → Least significant bit.
2	9	– 0
2	4	– 1
2	2	– 0
2	1	– 0.
	0	– 1 → most significant bit

$(36)_{10} = (100100)_2$.

(ii) $(23)_{10}$ to Hexadecimal.

16	23	
16	1	7 – LSB.
	0	1 – MSB

$(23)_{10} = (17)_{16}$.

(iii) $(D6C1)_{16}$ to decimal.

$= D \times 16^3 + 6 \times 16^2 + C \times 16^1 + 1 \times 16^0$

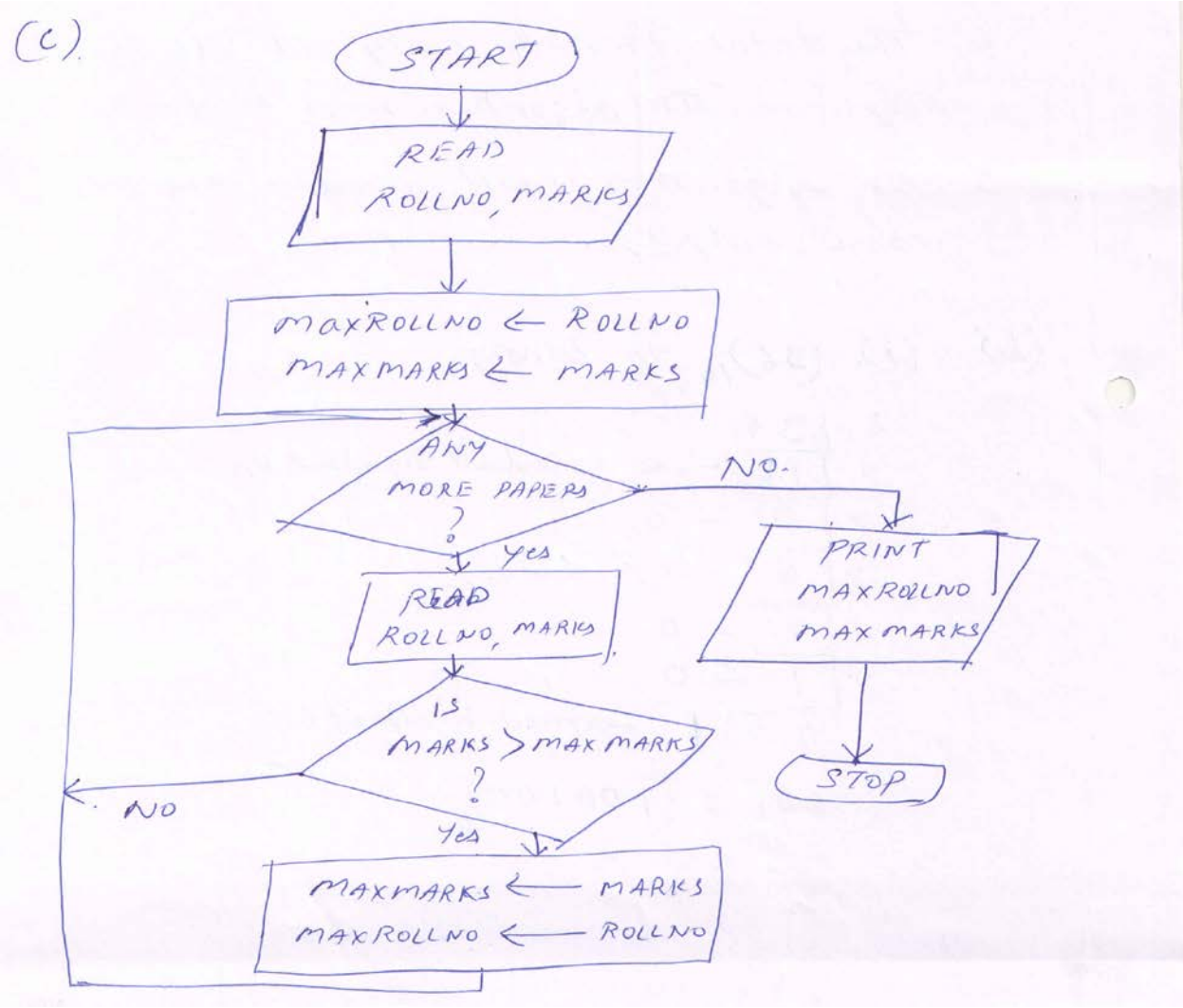
$= 13 \times 16^3 + 6 \times 16^2 + 12 \times 16 + 1 \times 16^0$

$= 53248 + 1536 + 192 + 1$

$= (54977)_{10}$.

c. Write the flow chart to pick the highest marks. (5)

Answer:



Q.3 a. Write a brief note on categories of keys present on a standard keyboard.(5)
Answer:

3(a) The most common input unit is a Keyboard used for manual data entry.

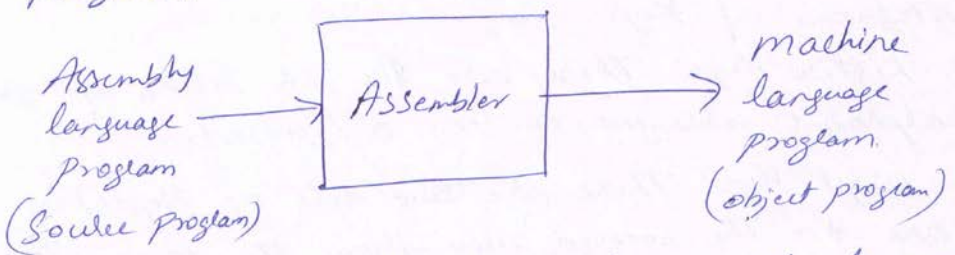
The Keyboard consists of the following major categories of Keys.

1. Letter Keys: These are the 26 letters of English alphabet arranged as in a typewriter.
2. Digit Keys: There are two sets of digits Keys. one on the second row from the top of the Keyboard just as in a typewriter, and the other is a numeric Key pad at the bottom right which allows quick entry of numbers with finger of one hand.
3. Special characters Keys: These are characters such as <, >, ?, /, £, }, [,], (,), ., ", ! @, #, $, %, ^, &, *, -, +, =, —, most of these are printed when the shift Key in the Keyboard is pressed down and the Key on which it is written is pressed.
4. Non printable control Keys: These are used for backspacing, going to next line, tabulation, moving cursor up or down, insert, delete characters, etc. There is also a Space bar at the bottom for leaving a Space.
5. Function Keys: These are labelled F1, F2, upto F15 and when pressed will invoke programs stored in the computer.

b. What is an assembler? Mention the main disadvantages of assembly language. (5)

Answer:

3(b) Assembler is a translator which translates Assembly language program to machine language program.



The main disadvantages of assembly language

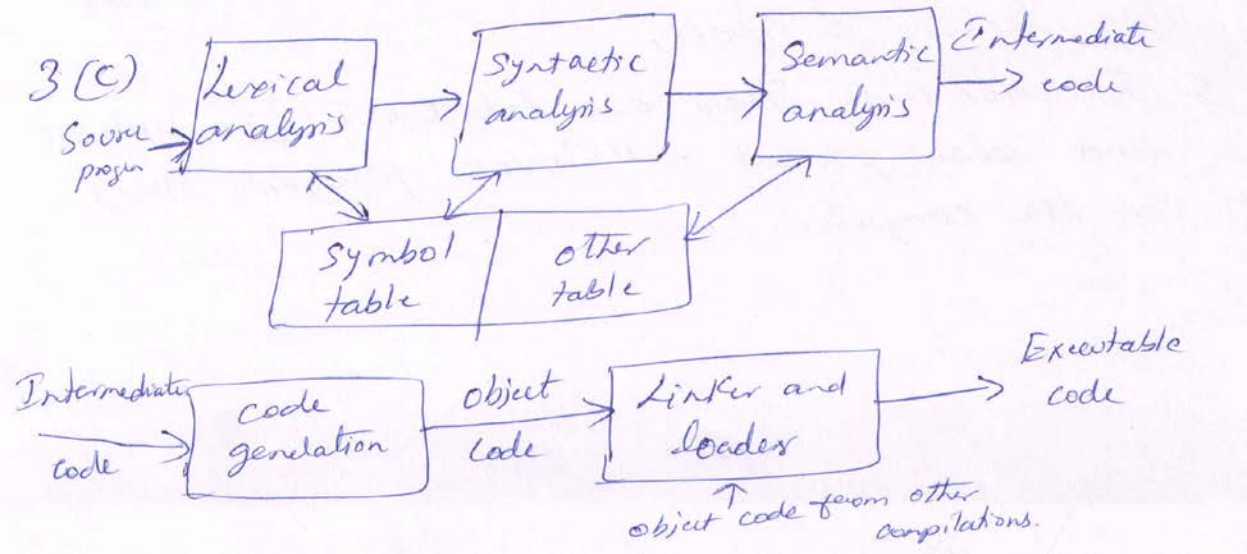
(i) It is machine-dependent. Thus programs written for one model of a computer cannot be executed on another model. In other words, it is not portable from one machine to another.

(ii) An assembly language programmer must be an expert who knows all about the logical structure of the computer for which the program is written.

(iii) Writing assembly language programs is difficult and time-consuming.

c. Write the steps involved in translation of high level language to machine language with the help of block diagram. (6)

Answer:



Q.4 a. Explain the internal structure of a microprocessor. (10)

Answer:

(11)

4(a) The central Processing Unit is a single-large scale Semiconductor integrated circuit chip and is known as the microprocessor.

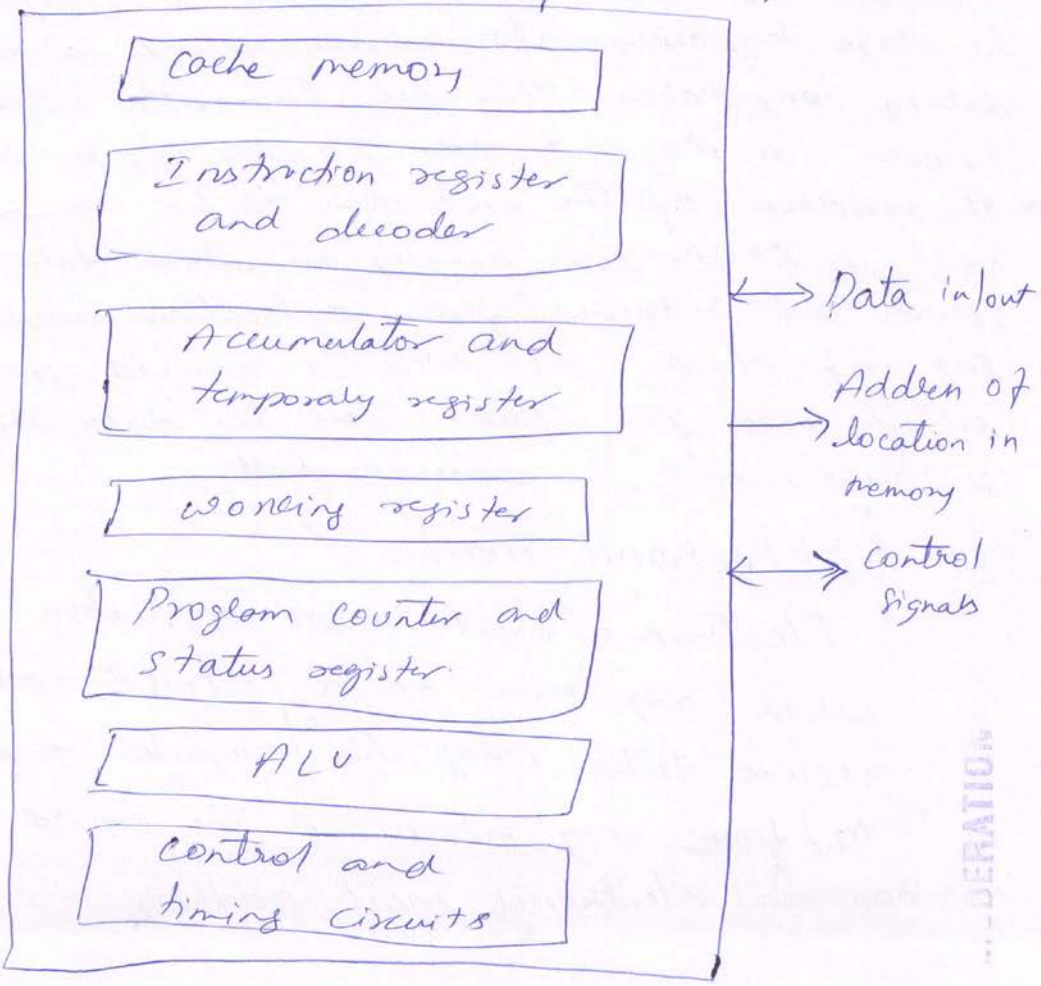
The chip contains:-

* An instruction register and a decoder.
* An arithmetic logic unit (ALU)
* A number of registers to store and manipulates data.
* control and timing circuits.
* Cache memory.

4(a)

The microprocessor is connected to the other units in the system by means of an address bus, a data bus and a control bus.

The internal structure of a microprocessor.



(12)

Instruction register and decoder:- The instruction register holds the instruction read from memory. The decoder sends the control signals appropriate to the decoded instruction to the ALU.

Arithmetic logic unit: ALU has circuits to perform the basic arithmetic operations, logical operations register operations, memory operations, program sequencing control operations and input/output operations.

Working Registers:- The time required to retrieve data from registers and to store results in them is at least ten times faster than that required to retrieve and store data in a RAM. Thus some working registers are provided in the CPU to store temporarily intermediate results obtained during computation. There are two other registers known as PC and SP. The PC register holds the address of the instruction to be executed next. A stack is a memory in which data is stored and retrieved from a location called the top of stack. As data is received it is pushed into the stack. Thus the data which is stored last is retrieved first.

b. Explain the following: (6)
(i) Electronic mail
(ii) File transfer
(iii) WWW

Answer:

(b) (i) Electronic mail.

Electronic mail is an application in which any user on a network can send/receive letters using his computer terminal to/from any person in the world who has an electronic mail address.

(13)

Internet provides a worldwide electronic mail facility. For example, any person in the world having access to the internet from his work place or home. can send me email to ~~any~~ email address rajaram@serc.iisc.ernet.in. The general format of internet email address is:

<name of addressee> @ <identity of his dept>. <institution>. <identity of Internet Service Provider>. <country code>.

mails can be sent not only to individuals but to groups, using group identity.

File transfer :- mail is intended for short messages. A file transfer program is available on the internet which allows transferring a large file containing programs or data from a computer in any part of the world to another. The files can be quite large (a few MB). The system provides authorization to persons allowed to copy the file. The file transfer is reliable. The rule used in Internet for file transfer is called file transfer protocol or FTP for short.

(14)

WWW :- Information is normally dispersed over many computers connected to the Internet. If a user wants to obtain information from any of the these computers they must be logically linked. The information stored in computers is not only text but also graphics, sound and video. This is called multimedia information.

Each document can be indexed using a number of keywords which can be used to link it through pointers to related multimedia items. A document annotated using keywords along with links linking it to other related documents is called a hypertext. A special notation is used to mark the keywords and links in hypertext. Every web page has a unique address called Universal Resource locator (URL). In order to transfer files on the web a special set of rules called hypertext transfer protocol (http) is used. TO obtain information from the web page a software called a browser is used which normally has a good graphical user interface (GUI).

Q.5 a. Write the rules which are to be following while defining an identifier. (5)
Answer:

5(a) The rules to be followed when defining Identifiers.

(i) First character must be an alphabet or underscore.
(ii) Must consist of only letters, digits or underscore
(iii) only first 31 characters are significant.
(iv) Cannot use a Keyword.
(v) must not contain white space.

b. With an example, explain different Relational operators, Logical operators & Bitwise operators used in C. (9)

Answer:

(b) Relational operators can be used for comparisons. Diffent relational operators used in C are

Operator	Meaning
<	is less than
<=	is less than or equal to.
>	is greater than
>=	is greater than or equal to.
==	is equal to.
!=	is not equal to.

A relational expression contains minimum of one relational operator and take the form.

exp1	relational operator	exp2.

Ex: 4.5 <= 10 True.

4.5 < -10 False

-35 >=0 False

(16)

Logical operators.

Logical operators are used when we want to test more than one condition and make decisions.

Operators	meaning
&&	Logical AND.
||	Logical OR.
!	Logical NOT.

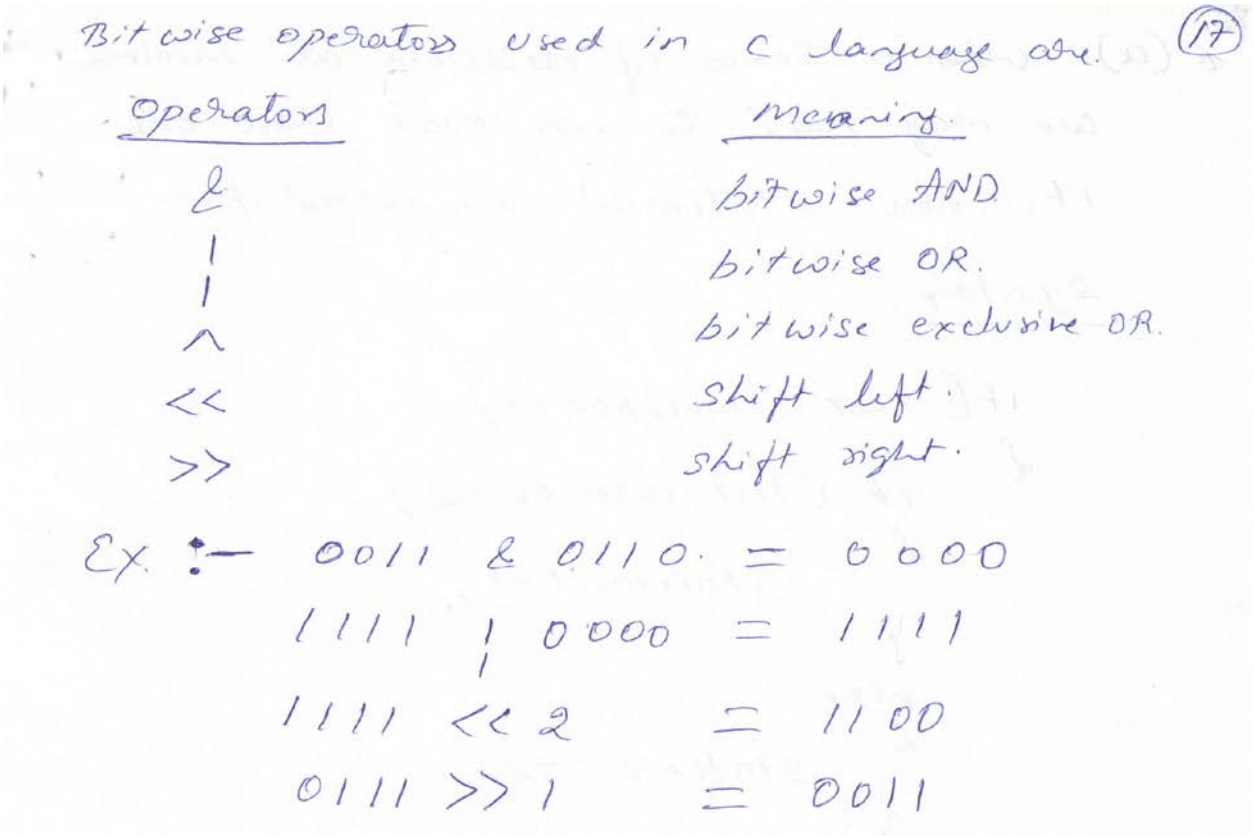
Logical expression also yields a value of one or zero. according to the truth table.

op1	op2	!op1	op1 && op2	op1 || op2	!op2
0	0	1	0	0	1
0	Non zero	1	0	1	0
Non zero	0	0	0	1	1
non zero	non zero	0	1	1	0

Ex: if (age > 55 && salary < 1000)

Bit wise operators.

Bit wise operators are used for manipulation of data at bit level. These operators are used for testing the bits, or shifting them right or left.



Bitwise operators used in C language are. (17)

Operators	Meaning
&	bitwise AND
|	bitwise OR.
^	bitwise exclusive OR.
<<	shift left.
>>	shift right.

Ex :- 0011 & 0110 = 0000

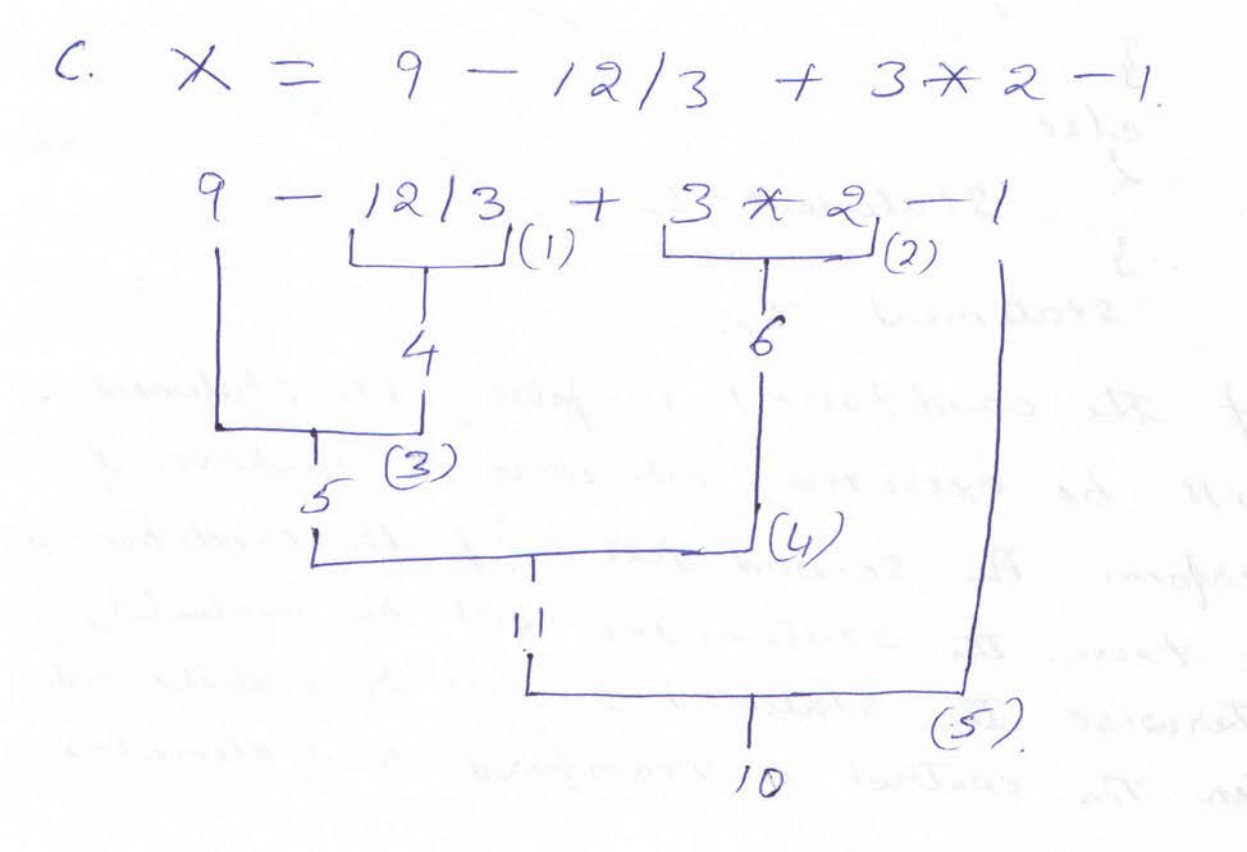
1111 | 0000 = 1111

1111 << 2 = 1100

0111 >> 1 = 0011

c. Evaluate the following expression using rules of evaluation.
X = 9 – 12/3 + 3 * 2 – 1 **(2)**

Answer:



Q.6 a. Explain nesting of IF...ELSE statement with an example. (10)

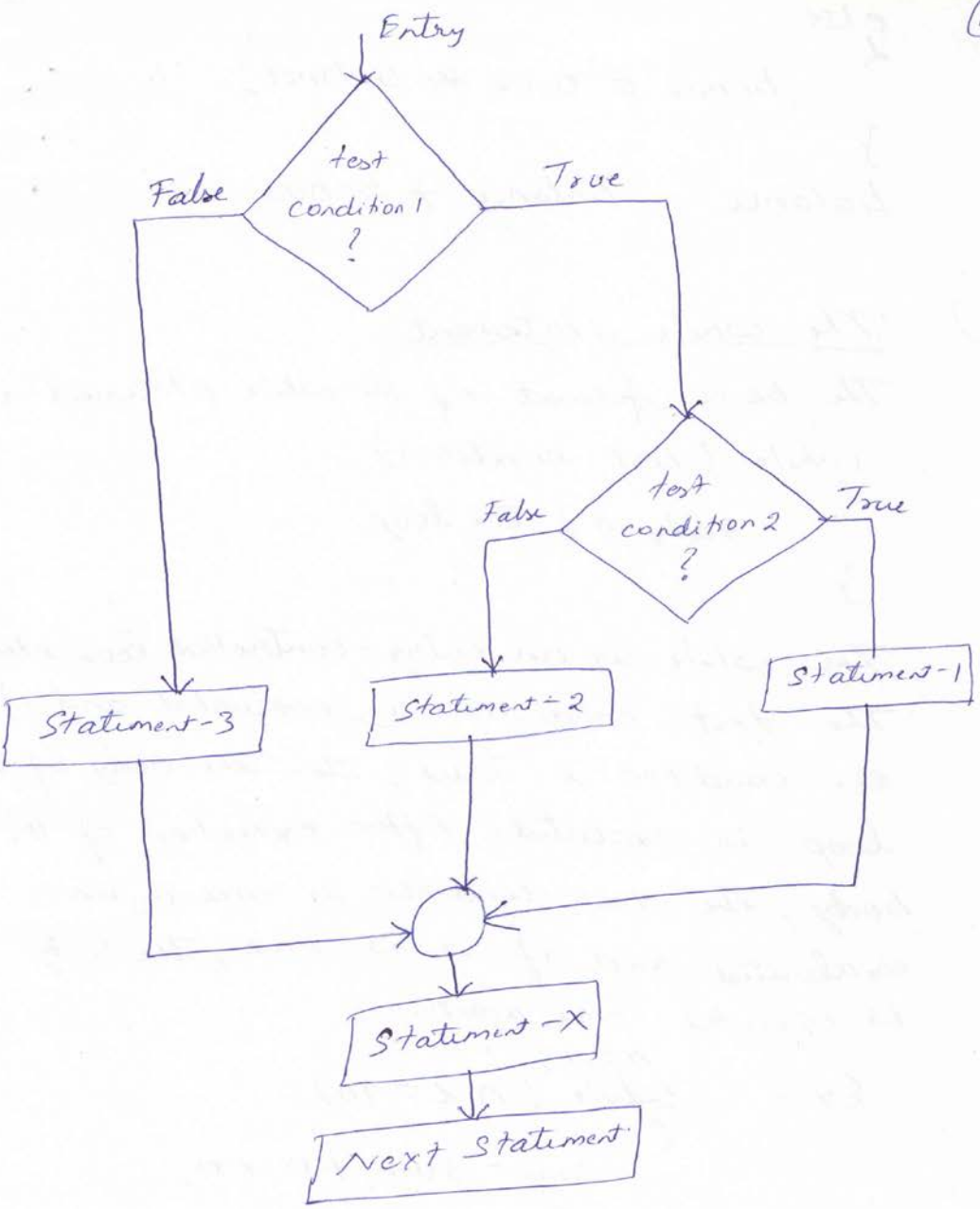
Answer:

18 6 (a). When a series of decisions are involved we may have to use more than one if...else statement in nested form

Syntax:-

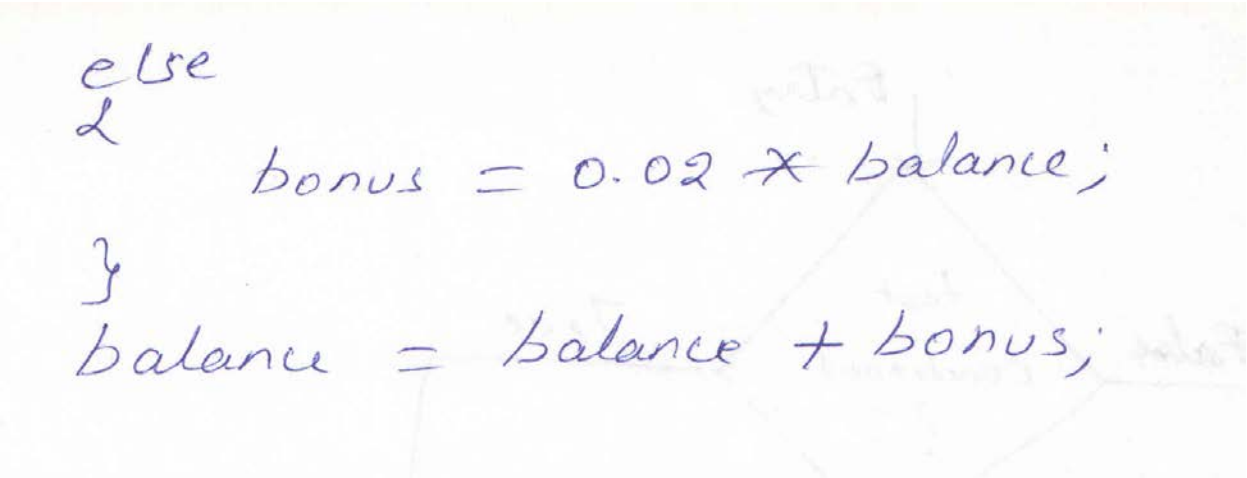
```
if( test condition-1)
{
    if (test condition-2)
    {
        statement-1;
    }
    else
    {
        statement-2;
    }
}
else
{
    statement-3;
}
statement-X;
```

If the condition-1 is false, the statement-3 will be executed; otherwise it continues to perform the second test. If the condition-2 is true, the statement-1 will be evaluated; otherwise the statement-2 will be evaluated and then the control is transferred to statement-X.



Example:-

```
if ( sex is female)
{
    if (balance > 5000)
        bonus = 0.05 * balance;
    else
        bonus = 0.02 * balance;
}
```



b. Differentiate between while statement and do-while statement. (6)

Answer:

(b) The while statement.

The basic format of the while statement is.

```
while (test condition)
{
    body of the loop.
}
```

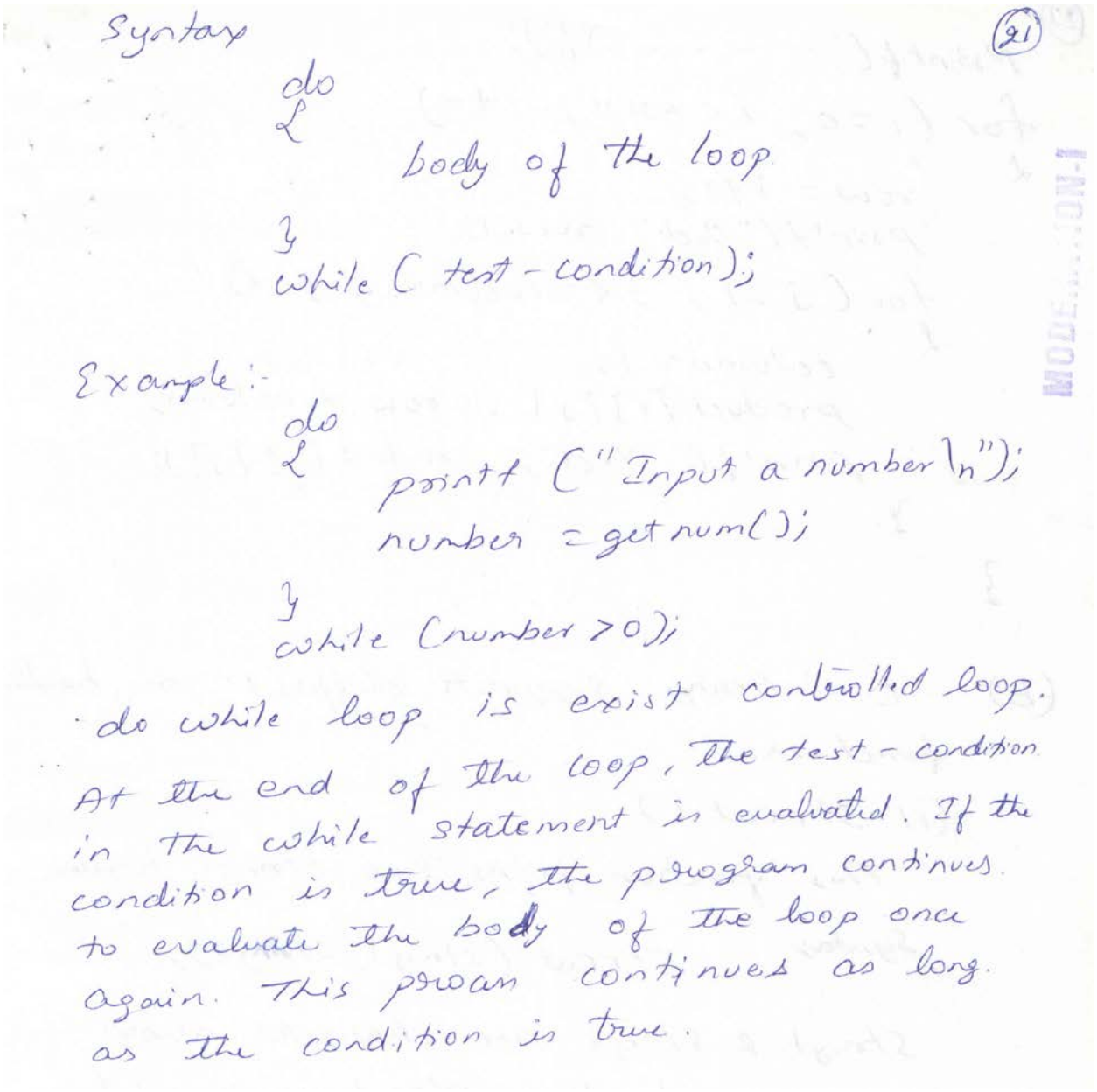
The while is an entry-controlled loop statement The test-condition is evaluated and if the condition is true, then the body of the loop is executed. After execution of the body, the test-condition is once again evaluated and if it is true, the body is executed once again.

Ex:-

```
n=1;
while (n<=10)
{
    sum = sum + n * n;
    n = n+1;
}
printf ("sum = %d \n", sum);
```

do while statement

In do while statement the test-condition is executed after the body of the loop is executed at least once.



Syntax (21)

```
do
{
    body of the loop
}
while ( test - condition );
```

Example:-

```
do
{
    printf ("Input a number\n");
    number = getnum();
}
while (number > 0);
```

do while loop is exist controlled loop. At the end of the loop, the test-condition in the while statement is evaluated. If the condition is true, the program continues to evaluate the body of the loop once again. This proam continues as long as the condition is true.

Q.7 a. Write a program to print multiplication table using two-dimensional array. (8)

Answer:

7 (a).

```
#define ROWS 5
#define COLUMNS 5
main()
{
    int row, column, product [ROWS] [columns];
    int i, j;
    Printf("MULTIPLICATION TABLE\n");
    Printf(" ");
    for(j=1; j<=COLUMNS; j++)
        Printf("%4d", j);
Printf("- - - - -\n");
for (i=0; i<ROWS; i++)
{
    row = i+1;
    printf("%d", row);
    for (j=1; j<=COLUMNS; j++)
    {
        column = j;
        product [i] [j] = row * column;
        printf(" %d", product [i] [j]);
    }
}
```

b. Explain different string handing functions. (8)

Answer:

(b) C library supports different string handling functions.

(i) strcat ()

This function joins two strings together

Syntax. strcat (string1, string2);

String1 & String2 are character arrays.

when the function strcat is executed String2 is appended to string1.

Ex: S3 = strcat (S1, S2);

where S1 = GOOD, S2 = MORNING.

S3 = GOODMORNING.

(ii) strcmp ().

This function compares two strings identified by arguments and has a value 0 if they are equal. If they are not, it has the numeric difference between the first nonmatching characters in the strings.

Syntax: strcmp(string1, string2);

String1 & string2 may be string variable or string constants.

Example :- strcmp ("Rom", "Ram"); return non zero.

strcmp ("Rom", "Rom"); return zero.

(iii) strcpy ()

The strcpy function works almost like a string assignment operator.

Syntax: strcpy (string1, string2);

String2 is assigned to string1.

Ex: strcpy (S1, "BOOK");

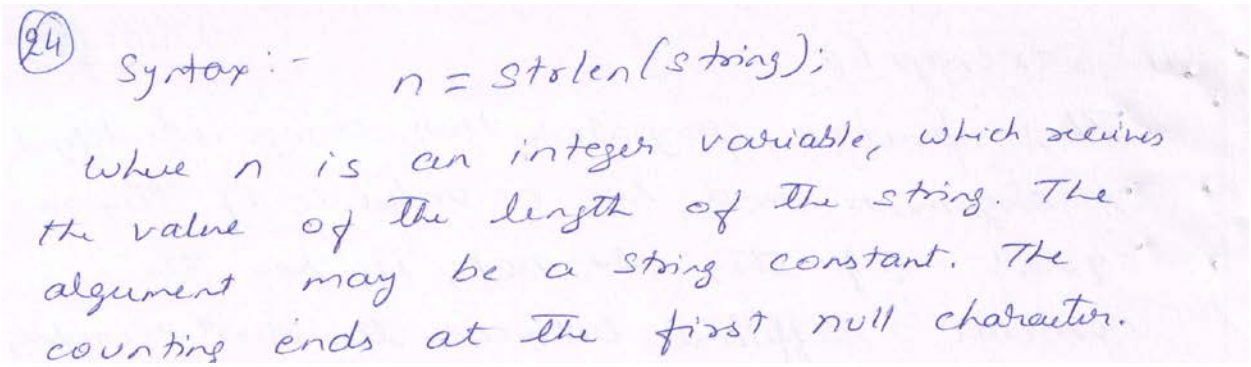
S1 is assigned value BOOK.

strcpy (SS1, SS2);

String in SS2 is assigned to variable SS1.

(iv) strlen ().

This function counts and returns the number of characters in string.



Q.8 a. What is function prototype? Explain with an example. (6)
Answer:

8 (a). Like variables, all functions in a C program must be declared, before they are invoked. A function declaration or function prototype consists of four parts.

Function type. (return type).

Function name.

Parameter list.

Terminating semicolon.

Format :-

Function-type function-name (parameter list);

A prototype declaration may be placed in two places in a program.

1. Above all the functions (including main).
2. Inside a function difination.

When we place the declaration above all the functions, the prototype is refered to as a global prototype. Such declarations are available for all the functions in the program.

When we place it in a function difinition the prototype is called a local prototype.

Such declarations are primarily used by the functions containing them.

Example:

```
#include<stdio.h>
int mul(int x, int y); /*Prototype*/
main()
{
    int y;
    y = mul(10, 5);
}
int mul(int x, int y)
{
    int p;
    p = x * y;
    return(p);
}
```

b. List out different categories of function, explain any two categories in detail. (10)

Answer:

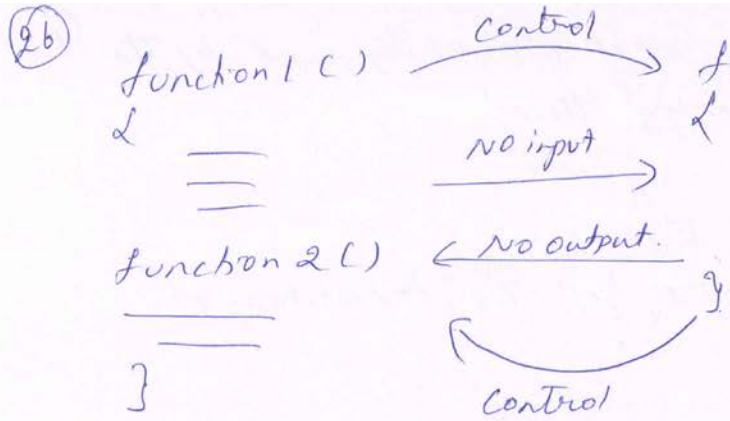
(b) A function can be classified depending on whether arguments are present or not and whether a value is returned or not.

Category 1: Functions with no arguments and no return value
Category 2: Functions with arguments and no return value
Category 3: Functions with arguments and one return value.
Category 4: Functions with no arguments but return a value.
Category 5: Functions that return multiple values.

Category 1:

When a function has no arguments, it does not receive any data from the calling function. When it does not return a value, the calling function does not receive any data from the called function.

(26)

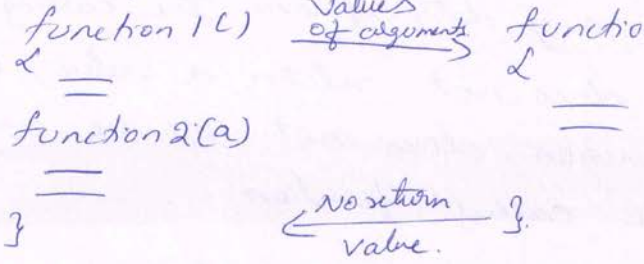


Example:

```
#include<stdio.h>
void print();
{
    Printf(" Hello ");
}.
Void main()
{
    print();
}.
```

Category 2

We could make the calling function to read data from the terminal and pass it on to the called function. In this approach calling function can check for the validity of data if necessary, before it is handed over to the called function.



(27)

example! -

```
#include <stdio.h>
void add (int x, int y)
{   int c;
    c = x+y;
    printf("value of c is %d", c);
}.
void main ()
{
    int x=1, y=2;
    add (x, y);
}.
```

<u>category 3.</u>

To assure a high degree of portability between programs, a function should generally be coded without involving any I/o operations, because different programs may require different output formats for display of results. These shortcomings can be overcome by handling over the result of a function to its calling function where the returned value can be used as required by the program.

```
function1()      Values of arguments →    function2(f)
{ ——                                      {
function2(a)                                 ==
  =                                          return (e);
}                ← Function result        }.
```

(20) Example:-

```
#include<stdio.h>
int add (int x, int y)
{
    int c;
    c = x+y;
    return c;
}
void main()
{
    int x=1, y=2, z=0;
    z=add(x, y);
    Printf("value of z is %d", z);
}
```

Category 4:

There could be occasions where we may need to design functions that may not take any arguments but returns a value to the calling function. The getchar function has no parameters but it returns an integer type data that represents a character.

Example:-

```
int get_number(void);
main()
{
    int m = get_number();
    printf("%d", m);
}
int get_number(void)
{
    int number;
```

```
    scanf("%d", &number);
    return (number);
}
```

Category 5:

Suppose we want to get more information from a function. we can achieve this in C using arguments not only to receive information but also to send back information to the calling function. The arguments that are used to "send out" information are called output parameters.

The mechanism of sending back information through arguments is achieved using what are known as the address operator (&) and indirection operator (*).

Example:

```
Void mathoperation (int x, int y, int *s, int *d)
main ()
{
    int x=20, y=10, s, d;
    mathoperation (x, y, &s, &d);
    printf("s = %d \n d = %d \n", s, d);
}
Void mathoperation (int a, int b, int *sum,
                                  int *diff)
{
    *sum = a+b;
    *diff = a-b;
}
```

Q.9 a. What is a pointer variable? Write a program using pointers to compute the sum of all elements stored in an array. (6)

Answer:

9 (a) A pointer variable is a variable that contains an address, which is a location of another variable in memory. Since a pointer is a variable, its value is also stored in the memory in another location.

```
#include<stdio.h>
Void main()
{
    int *p, sum, i;
    int x[5] = {5, 9, 6, 3, 7};
    i=0;
    p = x;
    printf("Element value Address\n");
    while (i<5)
    {
        printf(" x[%d] %d %u\n", i, *p, p);
        sum = sum + *p;
        i++, p++;
    }
    printf("\n sum = %d\n", sum);
    printf("\n &x[0] = %u\n", &x[0]);
    printf("\n p = %u\n", p);
}.
```

b. How to define a file? Explain how open operation is performed on a file. (10)

Answer:

9(b). If we want to store data in a file (31)
in the secondary memory, we must specify certain things about the file, to the operating system. They include.

1. File name.
2. Data structure.
3. Purpose.

Filename is a string of characters that make up a valid filename for the operating system. It may contain two parts, a primary name and an optional period with the extension.

Ex:- Input.C.
output.exe

Data structure of a file is defined as FILE in the library of standard I/O function definitions. Therefore, all files should be declared as type FILE before they are used.

FILE is a defined data type.

When we want to open a file, we must specify what we want to do with the file.

Format for declaring and opening a file.

```
FILE *fp;
fp = fopen("filename", "mode");
```

The first statement declares the variable fp as a "pointer to the data type FILE".

(32).

The second statement opens the file named filename and assigns an identifier to the FILE type pointer fp. This pointer, which contains all the information about the file is subsequently used as a communication link between the system and the program.

The second statement also specifies the purpose of opening this file. The mode does this job. mode can be one of the following.

r open the file for reading only.

w open the file for writing only.

a open the file for appending (or adding) data to it.

Example:-

```
FILE *p1, *P2;
P1 = fopen ("data", "r");
P2 = fopen ("results", "w");
```

Text Book

Fundamentals of Computers, V Rajaraman, Fourth Edition ,PHI2007.
Programming in ANSI C,E. Balagurusamy , Third Edition, Tata McGraw Hill

(33)

marks distribution

Q1 a to j $2 \times 10 = 20$.

Q2 (a). Each point 1 marks Total 5.
$1 \times 5 = 5M$.

(b) Each subdivision 2M.
$3 \times 2 = 6M$.

(c). flow chart – 5 M.

3 (a) Each category 1 M.
$5 \times 1 = 5M$.

(b) difination 1 M.
block diagram 1M.
3 disadvantages 3M (1M each)

(c). Diagram – 6M.

4 (a). Diagram 3 M.
Explanation 7M.

(b). Each subdivision – 2 M.
$3 \times 2 = 6M$.

5 (a) 5 points each 1M.
$5 \times 1 = 5M$.

(b). Each operator explanation 3M
$3 \times 3 = 9M$.

(c). complete evaluation – 2M.

6 (a) (34) Page no
Syntax – 3M.
Flow chart – 3M.
Example & Explanation – 4M.

(b). Explanation of while – 3M.
Explanation of do while – 3M.

7 (a) program – 8M.

(b) 4 functions each 2M.
4 × 2 = 8M.

8 (a). Syntax – 2M.
Explanation with example – 4M.

(b). Listing of categories – 2M.
Explanation any 2 catgories – 8M.
(4m each).

9 (a) Defination – 2M.
program – 4M.

(b). Defining a file – 4M.
Open operation – 6M.