TYPICAL QUESTIONS & ANSWERS

PART - I,

OBJECTIVE TYPE QUESTIONS

Each Question carries 2 marks.
Choose correct or the best alternative in the following:

Q.1 Literal means

(A) a string. **(B)** a string constant.
(C) a character. **(D)** an alphabet.

Ans:B

Q.2 Choose the correct answer

(A) Casting refers to implicit type conversion.
(B) Coercion refers to implicit type conversion.
(C) Casting means coercion.
(D) Coercion refers to explicit type conversion.

Ans:B

Q.3
```
printf ("%d", printf ("tim"));
```

(A) results in a syntax error **(B)** outputs tim3
(C) outputs garbage **(D)** outputs tim and terminates abruptly

Ans:B
printf statement will print tim3, "tim" due to inner printf statement and 3 as length due to outer printf statement.

Q.4 Output of the following program fragment is

```
x = 5;
y = x++;
printf("%d%d", x, y);
```

(A) 5, 6 **(B)** 5, 5
(C) 6, 5 **(D)** 6, 6

Ans:C
x is incremented by 1 and before that increment the value is assigned to y so value of x=6 and y=5.

Q.5 The value of an automatic variable that is declared but not initialised will be

(A) 0 **(B)** -1
(C) unpredictable **(D)** none of these

Ans:C

Q.6 Consider the following program

```
main ( )
{        float a = 0.5, b = 0.7;
                  if (b < 0.8)
             if (a < 0.5) printf ("ABCD");
                           else printf ("PQR");
                  else printf ("JKLF);
    }
```

The output is

(A) ABCD **(B)** PQR
(C) JKLF **(D)** None of these

Ans:B

Since b=0.7<0.8, the control goes to second "if" statement where (a<0.5) is false to printf statement in else part executed printing "PQR"

Q.7 The following program fragment

```
int *a;
*a = 7;
```

(A) assigns 7 to a **(B)** results in compilation error
(C) assigns address of a as 7 **(D)** segmentation fault

Ans:D

Q.8 A pointer variable can be

(A) passed to a function as argument.
(B) changed within function.
(C) returned by a function.
(D) assigned an integer value.

Ans:C

Q.9 'C' is often called a

(A) Object oriented language **(B)** High level language
(C) Assembly language **(D)** Machine level language

Ans:B

Q.10 The loop in which the statements within the loop are executed at least once is called

(A) do-while **(B)** while
(C) for **(D)** goto

Ans:A

Q.11 The control automatically passes to the first statement after the loop in

(A) continue statement **(B)** break statement
(C) switch statement **(D)** if statement

Ans:B

Q.12 A self contained block of statements that perform a coherent task of some kind is called a
(A) Monitor **(B)** Function
(C) Program **(D)** Structure

Ans:B

Q.13 Recursion is sometimes called
(A) Circular definition **(B)** Complex definition
(C) Procedure **(D)** Union

Ans:A

Q.14 Unsigned integer occupies
(A) Two bytes **(B)** Four bytes
(C) One byte **(D)** Eight bytes

Ans:B

Q.15 Each C preprocessor directive begins with
(A) # **(B)** include
(C) main() **(D)** {

Ans:A

Q.16
```
main() {
   long i = 30000;
   printf("%d", i); }
```
the output is
(A) 3000 **(B)** 30000
(C) 0 **(D)** -1

Ans:B

Q.17 The directive that can be used to test whether an expression evaluates to a nonzero value or not is
(A) #if **(B)** #elif
(C) #endif **(D)** #exit

Ans:A

Q.18
```
main() {
   printf("%p\n", main());
   }
```
(A) Prints the address of main function.

(B) Prints 0.
(C) Is an error.
(D) Is an infinite loop.

Ans:A

Q.19 The << operator is used for
(A) Right shifting **(B)** Left shifting
(C) Bitwise shifting **(D)** Bitwise complement

Ans:B

Q.20 The C language includes the header file standard input & output in
(A) stdlib.h library **(B)** stdio.h library
(C) conio.h library **(D)** #include library

Ans:B

Q.21 The value that follows the keyword CASE may only be
(A) constants **(B)** variable
(C) number **(D)** semicolon

Ans:A

Q.22 The statement which is used to terminate the control from the loop is
(A) break **(B)** continue
(C) goto **(D)** exit

Ans:A

Q.23 The machine registers are sometimes called
(A) local variables **(B)** global variables
(C) accumulators **(D)** static variables

Ans:A

Q.24 Set of values of the same type, which have a single name followed by an index is called
(A) function **(B)** structure
(C) array **(D)** union

Ans:C

Q.25 An array of pointers is same as
(A) pointer to array **(B)** pointers to pointers
(C) pointer to function **(D)** pointer to structure

Ans:B

Q.26 What is the output of the following program segment?

```
main()
{
  long i = 65536;
  printf("%d\n", i);
}
```

(A) 0 **(B)** 65536
(C) -1 **(D)** 65

Ans:A

Q.27 What is the output of the following program segment?

```
main()
{
  int i = 1;
do
  { printf("%d..", i);
  } while(i--);
}
```

(A) 0..1.. **(B)** 1..0..
(C) 0 **(D)** -1

Ans:B

Q.28 What is the output of the following program segment?

```
main()
{
  int i = ++2;
  printf("%d\n", i);
}
```

(A) 3 **(B)** 2
(C) 0 **(D)** -1

Ans:A
It is a compilation error. However if we write "i=2; ++i;" then value of i is printed as 3.

Q.29 The name of all functions end with a
(A) pair of parenthesis **(B)** semicolon
(C) braces **(D)** colon

Ans:A

Q.30 A float variable can store any variable within the range of
(A) -1.7×10^{38} to 1.7×10^{38} **(B)** -3.4×10^{38} to 3.4×10^{38}

(C) -7.2×10^{38} to 7.2×10^{38} **(D)** -1.2×10^{38} to 1.2×10^{38}

Ans:B

Q.31 scanf() can be used for reading
(A) double character **(B)** single character
(C) multiple characters **(D)** no character

Ans:C

Q.32 'C' allows a three-way transfer of control with the help of
(A) unary operator **(B)** relational operator
(C) ternary operator **(D)** comparison operator

Ans:C

Q.33 The statement that transfers control to the beginning of the loop is called
(A) break statement **(B)** exit statement
(C) continue statement **(D)** goto statement

Ans:C

Q.34 A variable which is visible only in the function in which it is defined, is called
(A) static variable **(B)** auto variable
(C) external variable **(D)** local variable

Ans:D

Q.35 The number of arguments supplied from the command line, by convention, is known as
(A) arg c **(B)** arg v
(C) #define **(D)** #include

Ans:A

Q.36 Output of the program given below is

```
int i;
main()
{
  printf("%d", i);
}
```

(A) 1 **(B)** 0
(C) -1 **(D)** Null

Ans:B

Q.37 What will be the output of the following program?

```
main()
{
  char *p = "ayqm";
  printf ("%c", ++*(p++));}
```

(A) b **(B)** z
(C) q **(D)** n

Ans:A

Q.38 What will be the output of the following program?

```
main()
{
  int i = 5;
  printf("%d", i=++i==6);
}
```

(A) 0 **(B)** 1
(C) 7 **(D)** 6

Ans:B

Q.39 Determine which of the following is a valid character constant

(A) '\\' **(B)** '\0'
(C) 'xyz' **(D)** '\052'

Ans:A

Q.40 The maximum value that an integer constant can have is

(A) .32767 **(B)** 32767
(C) 1.7014e+38 **(D)** –1.7014e+38

Ans:B

Q.41 The expression X=4+2%-8 evaluates

(A) –6 **(B)** 6
(C) 4 **(D)** None

Ans:B

Q.42 What will be the output of following program?

```
main()
{
   int x=15;
   printf("\n%d%d%d", x!=15, x=20, x<30);
}
```

(A) 0, 20, 1 **(B)** 15, 20, 30
(C) 0, 0, 0 **(D)** Error

Ans:A

Q.43 How many times the following program would print ("abc")?

```
main()
{
   printf("\nabc");
   main();
}
```

(A) Infinite number of times **(B)** 32767 times
(C) 65535 times **(D)** Till the stack does not overflow

Ans:A

Q.44 What would be output of the following program?

```
# define SQR(X) (X*X)
main()
{
   int a, b=3;
   a = SQR(b+2);
   printf("\n%d", a);
}
```

(A) 25 **(B)** 11
(C) Error **(D)** Garbage value

Ans:B

Q.45 What would be output of the following program?

```
#include "stdio.h"
main()
{
   printf("%d%d", size of (NULL!), size of (" "));
}
```

(A) 2 1 **(B)** 1 2
(C) 2 2 **(D)** 1 1

Ans:C

Q.46 What would be output of the following program, if the array begins at 65486?

```
main()
{
   int arr[ ] = {12, 14, 15, 23, 45};
   printf("%u%u", arr+1, &arr+1);
}
```

(A) 65486, 65486 **(B)** 65488, 65488
(C) 65488, 65496 **(D)** None of the above

Ans:C

Array begins at address 65486 so arr+1=65488 as size of int is 2 bytes and &arr+1=65486+10=65496 as there are 5 elements in array.

Q.47 Given the statement, maruti.engine.bolts=25;
which of the following is true?
(A) Structure bolts is nested within structure engine
(B) Structure engine is nested within structure maruti
(C) Structure maruti is nested within structure engine
(D) Structure maruti nested within structure bolts

Ans:B

Q.48 Which amongst the following is not a keyword?
(A) external **(B)** int
(C) float **(D)** double

Ans:A

Q.49 If a = 5 and b = 7 then the statement p = (a > b) : a ? b
(A) assigns a value 5 to p **(B)** assigns a value 7 to p
(C) assigns a value 8 to p **(D)** gives an error message

Ans:D

Q.50 The expression P >> 6 shifts all bits of P six places to right. What is the value of P >> 6 if P = 0×6db7 ?
(A) 0×1234 **(B)** 0×0001
(C) 0×0000 **(D)** 0×1B6

Ans:D

Q.51 If an integer occupies 4 bytes and a character occupies 1 bytes of memory, each element of the following structure would occupy how many bytes?

```
struct name
       {
       int age;
                                          char name [20];
       }
```

(A) 5 **(B)** 24
(C) 21 **(D)** 22

Ans:B

Q.52 If an array is used as function argument, the array is passed
(A) by value. **(B)** by reference

(C) by name. **(D)** the array cannot be used as a function argument.

Ans:B

Q.53 To access a structure element using a pointer, _____ operator is used

(A) dot (.) **(B)** pointer (&)
(C) pointer (*) **(D)** arrow ($\rightarrow$)

Ans:D

Q.54 The library function sqrt( ) operates on a double precision argument. If, i is an integer variable, which one of the following calls would correctly compute sqrt(i)?

(A) sqrt((double)i) **(B)** (double) sqrt(i)
(C) (double) (sqrt(i)) **(D)** sqrt(i)

Ans:A

Q.55 What will happen if the following loop is executed?

```
int num = 0;
do
{
   --num;
      printf("%d", num);
   num++;
   }while (num >= 0);
}
```

(A) The loop will run infinite number of times.
(B) The program will not enter the loop.
(C) There will be a compilation error.
(D) There will be runtime error.

Ans:C

Q.56 The break statement causes an exit

(A) Only from the innermost loop.
(B) Only from the innermost switch.
(C) From the innermost loop or switch.
(D) From the program.

Ans:C

Q.57 It is necessary to declare the type of function in the calling program if

(A) Function returns an integer.
(B) Function returns a non-integer value.
(C) Function is not defined in the same file.
(D) Function is called number of times.

Ans:B

Q.58 The function fprintf is used in a program

(A) When too many printf calls have been already used in the program.
(B) In place of printf, since printf uses more memory.
(C) When the output is to be printed on to a file.
(D) When the type of variables to be printed are not known before.

Ans:C

Q.59 The following statement displays

```
float x = 2000.53;
printf ("%e", x);
```

(A) 2.00053e+04 **(B)** 2.00053e+03
(C) 2.00053+e04 **(D)** 2.0005e+03

Ans:B

Q.60 The output of the following is

```
int a = 75;
printf ("%d%%", a);
```

(A) 75 **(B)** 75%%
(C) 75% **(D)** None of the above

Ans:D

Q.61 C language was invented by

(A) Abacus **(B)** Charles babage
(C) Thomson **(D)** Dennis Ritchie

Ans:D

Q.62 The given FOR loop is

```
for ( ; ; )
{
    printf(" ");
}
```

(A) valid **(B)** indefinite
(C) invalid **(D)** displays runtime errors

Ans:D
The given for loop displays runtime errors because no test condition is given. Test condition is must inside for loop.

Q.63 The following code displays

```
main( )
```

```
{
   int *p;
   p = (int*) malloc(sizeof(int));
   *p = 10;
   printf("p = %d\n", *p);
}
```

(A) 10 **(B)** 1542 (address of p)
(C) 20 **(D)** None of the above

Ans:A

Q.64 The _____ operator is a technique to forcefully convert one data type to the others

(A) Cast **(B)** Conversion
(C) Type **(D)** Uniary

Ans:A

Q.65 The output of the following will be

```
for (x=1, y=5; x+y<=10; x++)
{
   printf("%d%d", x,y);
   y++;
}
```

(A)
1 5
2 6
3 7

(B)
1 5
2 6
3 7
4 8

(C)
1 5
1 6
1 7
1 8
1 9

(D)
1 5
2 5
3 5
4 5
5 5

Ans:A

Q.66 The _____ statement causes immediate exit from the loop overriding the condition test

(A) Exit **(B)** Break
(C) Goto **(D)** None of the above

Ans:B

Q.67 The output of the following code is

```
a = 5;
a << 1;
printf("%d", a);
```

(A) 5 **(B)** 6
(C) 2 **(D)** 3

Ans:A

Q.68 The purpose for mode "w+b" in file operation is
(A) create a binary file for write
(B) create a binary file for read/write
(C) open a binary file for writing
(D) open a binary file for reading/writing

Ans:B

Q.69 The operators << and >> are
(A) assignment operator **(B)** relational operator
(C) logical operator **(D)** bitwise shift operator

Ans D

Q.70 Which of the following numerical value is invalid constant
(A) .75 **(B)** 9.3e2
(C) 27,512 **(D)** 123456

Ans C

Q.71 A C program contains the following declaration int i=8, j=5 what would be the value of following expression?

abs(i-2*j)

(A) 2 **(B)** 4
(C) 6 **(D)** 8

Ans A

Q.72 What will be the output of the following program

```
main( )
{
        int k,  num=30;
        k=(num>5 ? (num<= 10 ? 100:200):500);
        printf("\n%d", num);
}
```

(A) 100 **(B)** 5
(C) 30 **(D)** 500
Ans C

Q.73 What is the output of the following code

```
int n = 0, m=1;
```

```
do
{
printf("%d", m);
m++;
}
while(m<=n);
```

(A) 0 **(B)** 2
(C) 1 **(D)** 4

Ans C

Q.74 If a=8 and b=15 then the statement

```
x= (a>b) ? a:b;
```

(A) assigns a value 8 to x **(B)** gives an error message
(C) assigns a value 15 to x **(D)** assigns a value 7 to x

Ans C

Q.75 What is the output of the following code

```
int n=0, m;
for (m=1;  m<=n+1;  m++)
printf("%d", m);
```

(A) 2 **(B)** 1
(C) 0 **(D)** 6

Ans B

Q.76 How many times the following loop be executed

```
{
…
ch = 'b';
while(ch >= 'a' && ch <= 'z')
ch++;    }
```

(A) 0 **(B)** 25
(C) 26 **(D)** 1

Ans B

Q.77 Which of the following is FALSE in C

(A) Keywords can be used as variable names
(B) Variable names can contain a digit
(C) Variable names do not contain a blank space
(D) Capital letters can be used in variable names

Ans A

Q.78 int **ptr; is
(A) Invalid declaration **(B)** Pointer to pointer
(C) Pointer to integer **(D)** none of the above

Ans B

Q.79 A Pixel is _____.
(A) a computer program that draws picture
(B) a picture stored in secondary memory
(C) the smallest resolvable part of a picture
(D) None of these

Ans C

Q.80 Which number system is usually followed in a typical 32-bit computer?
(A) 2 **(B)** 10
(C) 16 **(D)** 32

Ans A

Q.81 Which technology is used in optical disks?
(A) Mechanical **(B)** Electrical
(C) Electro Magnetic **(D)** Laser

Ans D

Q.82 Which of the following storage devices can store maximum amount of data?
(A) Floppy Disk **(B)** Hard Disk
(C) Compact Disk **(D)** Magneto Optic Disk

Ans B

Q.83 EPROM can be used for
(A) Erasing the contents of ROM
(B) Reconstructing the contents of ROM
(C) Erasing and reconstructing the contents of ROM
(D) Duplicating ROM

Ans C

Q.84 Memory unit is one part of
(A) Input device **(B)** Control unit
(C) Output device **(D)** Central Processing Unit

Ans D

Q.85 MS-WORD is a
(A) system software **(B)** high level language
(C) spread sheet application **(D)** word processing package

Ans D

Q.86 The grammar dialog box can be involved by choosing grammar from _____ menu.
(A) insert **(B)** file
(C) tools **(D)** view

Ans C

Q.87 A template stores
(A) styles, macros
(B) Auto Text entires, Customized word command Settings
(C) graphics, text
(D) All of the above

Ans A

Q.88 To return the remainder after a number is divided by a divisor in EXCEL we use the function
(A) ROUND( ) **(B)** FACT( )
(C) MOD( ) **(D)** DIV( )

Ans C

Q.89 _____ unit controls the flow and manipulation of data and information.
(A) Arithmetic logic **(B)** Central
(C) Middle **(D)** Control

Ans D

Q.90 Usually, an algorithm will contain a number of procedural steps which are dependent on results of previous steps and is called _____.
(A) Flowchart **(B)** Chart
(C) Drawing Chart **(D)** Food Chart

Ans A

Q.91 The performance of cache memory is frequently measured in terms of a quantity called:
(A) bit ratio **(B)** nor ratio
(C) no ratio **(D)** hit ratio

Ans D

Q.92 In addition to communicating with I/O, the processor must communicate with the _____ unit.
(A) control **(B)** memory
(C) Arithmetic **(D)** process

Ans B

Q.93 Which of the following was not associated with second generation computers?
(A) high level procedural language **(B)** operating system
(C) magnetic core and transistor **(D)** All of the above were associated

Ans D

Q.94 Which of the following number system/code uses only 0s and 1s
(A) decimal **(B)** octal
(C) hexadecimal **(D)** none of the above

Ans D

Q.95 A group of characters that has some meaning as a unit is known as a
(A) field **(B)** file
(C) record **(D)** word

Ans A

Q.96 The process of production of customer list in alphabetical order falls under the category of
(A) editing **(B)** sorting
(C) updating **(D)** calculating

Ans B

Q.97 State in case of following statement whether it is true or false?
Magnetic drums and disks are quiet similar in operation.
(A) True **(B)** False

Ans A

Q.98 State in case of following statement whether it is true or false?
Ink-jet printers are classified as impact printers.
(A) True **(B)** False

Ans B

Q.100 A Compiler is _____.
(A) a combination of computer hardware
(B) a program which translates from one high-level language to another

(C) a program which translates from one high-level to a machine level
(D) None of these

Ans C

Q.101 When a key is pressed on the keyboard, which standard is used for converting the keystroke into the corresponding bits
(A) ANSI **(B)** ASCII
(C) EBCDIC **(D)** ISO

Ans B

Q.102 Which of the following is not an output device?
(A) Scanner **(B)** Printer
(C) Flat Screen **(D)** Touch Screen

Ans A

Q.103 The memory location address are limited to
(A) 00000 to 9ffff(16) **(B)** 00001 to 9ffff(16)
(C) 00010 to 9ffff(16) **(D)** 10000 to 9ffff(16)

Ans A

Q.104 The programs which are as permanent as hardware and stored in ROM is known as
(A) Hardware **(B)** Software
(C) Firmware **(D)** ROMware

Ans C

Q.105 Memory is made up of
(A) Set of wires **(B)** Set of circuits
(C) Large number of cells **(D)** All of these

Ans C

Q.106 Using Find command in Word, we can search
(A) characters **(B)** formats
(C) symbols **(D)** All of the above
Ans D

Q.107 MS-Word automatically moves the text to the next line when it reaches the right edge of the screen and is called
(A) Carriage Return **(B)** Enter
(C) Word wrap **(D)** None of the above

Ans C

Q.108 MS-EXCEL can be used to automate
(A) Financial statements, Business forecasting
(B) Transaction registers, inventory control
(C) Accounts receivable, accounts payable
(D) Any of the above

Ans D

Q.109 Command to delete all files and folders is _____.
(A) Deltree **(B)** Del
(C) Remove **(D)** CD

Ans A

Q.110 Machine language is a language
(A) Directly understood by a computer.
(B) Which needs to be translated.
(C) Which uses mnemonics.
(D) In which programs are written first.

Ans A

Q.111 Which of the following device have a limitation that one can only add information to it but cannot erase or modify.
(A) Floppy Disk **(B)** Hard Disk
(C) Tape Drive **(D)** CDROM

Ans D

Q.112 Which amongst the following devices can store the maximum amount of data
(A) Floppy Disk **(B)** Hard Disk
(C) Compact Disk **(D)** Magnetic Optic Disk

Ans B

Q.113 EPROM can be used for
(A) Erasing the contents of ROM.
(B) Reconstructing the contents of ROM.
(C) Erasing and reconstructing the contents of ROM.
(D) Duplicating ROM.

Ans A

Q.114 The earliest calculating devices are

(A) Abacus. **(B)** Clock.
(C) Difference . **(D)** None of these.

Ans A

Q.115 The language that the computer can understand and execute is called
(A) Machine Language. **(B)** Application Software.
(C) System Program. **(D)** Assembly language.

Ans A

Q.116 The three sequential function of CPU operation are
(A) Decode, fetch, execute. **(B)** Execute, decode, fetch.
(C) Fetch, execute, decode. **(D)** Fetch, decode, execute.

Ans D

Q.117 Moving process from a main memory to a disk is called
(A) Scheduling **(B)** Caching
(C) Swapping **(D)** Spooling

Ans C

Q.118 While running DOS on a PC, the command that is used to duplicate the entire diskette?

(A) COPY **(B)** DISK COPY
(C) CHK DISK **(D)** TYPE

Ans B

Q.119 If data is processed as it arrives, this type of data processing is called
(A) Real time processing. **(B)** Batch processing.
(C) Off line processing. **(D)** Distributed processing.

Ans A

Q.120 Client/Server architecture can be used in
(A) LAN **(B)** WAN
(C) MAN **(D)** All of these.

Ans D

Q.121 Which of the following is not a programming language?
(A) UNIX **(B)** LISP
(C) BASIC **(D)** ADA

Ans A

Q.122 The process of retaining data for future use is called
(A) Reading **(B)** Writing
(C) Storing **(D)** Coding

Ans C

Q.123 The process of manipulating the data to achieve some meaningful results is called
(A) Information handling. (B)Data sharing.
(C) Data distribution. (D)Data processing.

Ans D

Q.124 Windows is a(n)
(A) Operating system. **(B)** User interface
(C) Operating environment. **(D)** Programming platform.

Ans C

Q.125 The register that keeps track of the program during execution
(A) Address register **(B)** Program counter.
(C) Data register **(D)** Accumulator

Ans B

Q.126 Typical processing speed of Super Computer is of the order of
(A) 100 MIPS **(B)** 200 MIPS
(C) 300 MIPS **(D)** 400 MIPS and above

Ans D

Q.127 While using DIR command,_____switch can not be used with it
(A) /P. **(B)** /W.
(C) /S. **(D)** /T.

Ans D

Q.128 _____ is not a utility program
(A) Debugger **(B)** Editor
(C) Spooler **(D)** Defragmenter

Ans C

Q.129 The access method used for magnetic tape is
(A) Direct **(B)** Random

(C) Sequential **(D)** Indexed

Ans C

Q.130 _____ unit coordinates the sequencing of events within the central processor of a computer.
(A) Logic unit. **(B)** Arithmetic unit.
(C) Storage unit. **(D)** Control unit.

Ans D

Q.131 Which of the following is not a direct entry input device?
(A) Optical scanner. **(B)** Digitizer.
(C) Keyboard. **(D)** Light pen.

Ans C

Q.132 A network, which is used for sharing data, software and hardware among several users of microcomputers, is called
(A) Wide Area Network. **(B)** Metropolitan Area Network.
(C) Local Area Network. **(D)** Value Added Network.

Ans C

Q.133 The instructions / programs that are loaded into main memory when a computer is booted are
(A) Internal commands. **(B)** External commands.
(C) Utility Programs. **(D)** Loader.

Ans A

Q.134 Which of the following types of memory loses data when power is switched off?
(A) Magnetic tape **(B)** Static Random Access Memory
(C) Magnetic disk **(D)** CD-ROM

Ans B

Q.135 Which is the latest write-once optical storage media
(A) Digital Page. **(B)** CD - ROM disk.
(C) WORM disk. **(D)** Magneto- optical disk.

Ans B

Q.136 The factor that does not affect the storage capacity of a hard disk
(A) Track density. **(B)** Height of the hard disk drive.
(C) Recording density. **(D)** Number of plates.

Ans B

Q.137 The operating system that reads and reacts in terms of actual time is
(A) Batch system. **(B)** Quick response system.
(C) Real time system. **(D)** Time sharing system.

Ans C

Q.138 Which of the following is a spreadsheet package?
(A) Corel Draw. **(B)** Wordstar
(C) EXCEL. **(D)** MS-WORD.

Ans C

Q.139 Find and Replace option is placed under the_____ menu.
(A) Edit. **(B)** Insert.
(C) View. **(D)** File.

Ans A

Q.140 Ink-jet printers are
(A) Impact printers **(B)** Laser printers
(C) Non-impact printers **(D)** Optical printers

Ans C

Q.141 A refreshing circuit is required in
(A) SRAM **(B)** EPROM
(C) DRAM **(D)** EEPROM

Ans C

Q.142 The first component of the machine language statement is called
(A) Lines of code. **(B)** Op-code.
(C) Pseudocode. **(D)** Operators and operands.

Ans B

Q.143 10001 x 101 =
(A) 101101. **(B)** 1010101.
(C) 100101. **(D)** 101010.

Ans A

Q.144 In heap files, records are placed in
(A) Random order. **(B)** Sequential order.
(C) Indexed order **(D)** Printer-referenced order.

Ans A

Q.145 In immediate addressing mode the immediate data is

(A) Constant data. **(B)** Variable data
(C) Symbolic data. **(D)** **(A)** & **(C)** both

Ans A

Q.146 Conditional jump instructions in 8086 are always
(A) Short-jumps ranging from –128 bytes to +127 bytes.
(B) Jumps ranging from –256 bytes to +255 bytes.
(C) Short-jumps ranging from –64 bytes to +63 bytes.
(D) Jumps ranging from –256 to +256 bytes.

Ans A

Q.147 Lotus notes is a
(A) Spreadsheet package **(B)** Programming tool.
(C) Notepad **(D)** Groupware.

Ans D

PART – II

DESCRIPTIVES

Q.1 Write a C program that reads a fixed point real number in a character array and then displays rightmost digit of the integral part of number. **(6)**

Ans: A C program to display rightmost digit of the integral part of a read real number:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   char a[10];
   int i;
   clrscr();
   printf("enter a fixed point real no. : ");
   scanf("%s",a);
   i=0;
   while(a[i]!='.')
   i++;
   printf("%c",a[--i]);
   getch();
}
```

Q.2 Output the number x = 56.1624 under the following format specification

(i) printf ("%7.2f", x)
(ii) printf (%f", x)
(iii) printf("8.2e", x)
(iv) printf("%e", x) **(4)**

Ans: The output of number 56.1624 under various format specification is:

(i) 56.16
(ii) 56.162399
(iii) 5.62e+01
(iv) 5.616240e+01

Q.3 Write C statements to read the values of three variables of the type int, float and string and print them if correct data is entered otherwise print "error in input". **(4)**

Ans: A C program that read three values and print these if data input is correct otherwise print "error in input":

```
#include<stdio.h>
#include<conio.h>
void main()
{
```

```
   int n;
   float m;
   char p;
   printf("enter the values of n,m and p");
   ((scanf("%d %f %c",&n,&m,&p)==3)?printf("correct
   input"):printf("error in input"));
   getch();
}
```

Q.4 Write a C program that uses 'for' construct to find the sum of the following harmonic series for a given value of n and display the sum.

1 + 1/2 +1/3 +… + 1/n **(6)**

Ans: A C program to find and display sum of given harmonic series is listed below:

```
 #include<stdio.h>
#include<conio.h>
void main()
{
   int i,n;
   float sum=0.0;
   clrscr();
   printf("enter the value of n: ");
   scanf("%d",&n);
   for(i=1;i<=n;i++)
   sum=sum+(float)1/i;
   printf("sum of the series upto %d = %f",n,sum);
   getch();
}
```

Q.5 Write conditional operators to evaluate the following function

y = 2.4x + 3, for x <= 2
y = 3x –5, for x > 2 **(4)**

Ans: The conditional operator for given problem can be written as y = ((x < = 2)? (2.4 * x + 3): (3 * x + 5)). A program to evaluate this conditional operator is given below:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int x;
   float y;
   clrscr();
   printf("enter the values of x\n");
   scanf("%d",&x);
   y=((x<=2)?(2.4*x+3):(y=3*x+5));
   printf("y=%f",y);
   getch();}
```

Q.6 Write a C program fragment using "do…while" construct to print out even numbers between 10 to 100 making sure that two numbers are written per line. **(4)**

Ans: A C program to print even numbers between 10 to 100 using "do-while" construct:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int i,k;
   clrscr();
   k=10;
   i=0;
   do
   {
      if(k>10)
      {
         printf("%d",k);
         printf(" ");
         if(i%2==0)
         printf("\n");
      }
      i++;
      k=k+2;
   }while(k<100);
   getch();
}
```

Q.7 Given an integer, write a C program that displays the number as follows:

first line : All digits of integer
second line : all except first rightmost digit
third line : all except two rightmost digits .
. .
last line : leftmost digit **(8)**

Ans: A C program to display a number as follows:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int n,k,l,i,r,a[100];
   clrscr();
   printf("enter any integer value");
   scanf("%d",&n);
   while(n>0)
   {
      printf("%d",n);
      printf("\n");
```

```
      n=n/10;
    }
    //to have space between the digits of the integer
    k=n;
    while(k>0)
    {
      l=0;
      n=k;
      while(n>0)
      {
        r=n%10;
        a[l]=r;
        l++;
        n=n/10;
      }
      l--;
      for(i=l;i>=0;i--)
      {
        printf("%d",a[i]);
        printf(" ");
      }
      printf("\n");
      k=k/10;
    }
    getch();
  }
```

Q.8 Describe the output of the following C program fragment

```
void main ()
{  int k = 0, x = 0;
  while (k < 25)
  {          if (k % 5 == 0)
              {    x += k;
                                printf("%d  ", x);
               }
  ++k;
  }
  printf("\nx=%d ", x +k);
  }
```

(6)

Ans: Output of the given program segment is:
05153050
x=75
k=0; test condition of while loop is true; If condition false till k=5
At k=5, (k%5==0) is true so x=x+k so printf prints x=0+5=5;
At k=10, (k%5==0) is true so x=x+k so printf prints x=5+10=15;
At k=15, (k%5==0) is true so x=x+k so printf prints x=15+15=30;

At k=20, (k%5==0) is true so x=x+k so printf prints x=30+20=50; k is incremented till 25 where we are out of while loop.
Outer printf statement prints x+k=50+25=75.

Q.9 Shown below is a Floyd's triangle. Write a C program to print this triangle **(8)**

```
1
2     3
4     5     6
7     8     9     10
11    …                 15

79    …                       91
```

Ans: A C program to print Floyd's triangle upto 91 is given below:

```
#include<conio.h>
void main()
{
        int num=1,k=1,j;
        clrscr();
        while(num<=91)
        {
              for(j=1;j<=k;j++)
                    printf(" %d",num++);
              printf("\n\n");
              k++;
        }
        getch();
}
```

Q.10 Given the string "DATA PROCESSING", write a C program to read the string from terminal and display the same in the following formats.
(i) DATA PROCESSING
(ii) DATA
PROCESSING **(6)**

Ans: A C program that read string "DATA PROCESSING" from the terminal and display that in two formats:
(i)DATA PROCESSING
(ii)DATA

PROCESSING

```
 #include<stdio.h>
#include<conio.h>
void main()
{
```

```
    int i;
    char a[50];
    clrscr();
    printf("enter the string");
    gets(a);
    puts(a);
    i=0;
    while(a[i]!=' ')
    {
       printf("%c",a[i]);
       i++;
    }
    i++;
    printf("\n");
    while(a[i]!='\0')
    {
       printf("%c",a[i]);
       i++;
    }
    getch();
}
```

Q.11 Write a C program to compute the value of sin function. The loop must terminate when the absolute value of the term is less than 0.00001.

$$\sin x = x\ x - x^3/3! + x^5/5! - x^7/7! + \ldots$$ **(8)**

Ans: A C program to compute the sum of sin function is listed below:

```
#include<conio.h>
#include<math.h>
void main()
{
 float x,dig,sum;
 int i,n;
 clrscr();
 printf("\n Enter the value of x : ");
 scanf("%f",&x);
 printf("\n Enter the value of n : ");
 scanf("%d",&n);
 /*convert x into radians*/
 x=x*3.1412/180;
 sum=x;
 dig=x;
 for(i=1;i<=n;i++)
 {
     dig=(dig*pow((double)(-1),(double)(2*i-
1))*x*x)/(2*i*(2*i+1));
```

```
        sum+=dig;
    }
    printf("\n The sum is : %6.2f",sum);
    getch();
}
```

Q.12 Write a switch statement that will examine the value of an integer variable flag and print the following messages: **(6)**

It is hot weather; if flag has value 1
It is a stormy weather; if flag has value 2
It is sticky weather; if flag has value 3
It is a pleasant weather; otherwise

Ans: A Program to demonstrate switch statement:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int flag;
    printf( "Enter any value\n" );
    scanf( "%d", &flag );
    switch ( flag ) {
        case 1:
            printf( "It is hot weather!\n" );
            break;
        case 2:
            printf( "It is a stormy weather!\n" );
            break;
        case 3:
            printf( "It is a sticky weather!\n" );
            break;
        default:
            printf( "It is a pleasant weather!\n" );
            break;
    }
    getch();
}
```

Q.13 Define a structure named 'student' containing two fields 'name' and 'marks'.

Ans: A structure student:

```
struct student
{
   char name[10];
   int marks[6];
   int total;
};
```

Q.14 Declare an array of structure having 50 elements of student type. **(2)**

Ans: An array of structure:

```
struct student stu[50];
```

Q.15 Write an input statement for inputting the marks and the names of 50 students defined as above. **(3)**

Ans: Statements for inputting marks of 50 students:

```
for(i=0;i<50;i++)
{
    printf("%d. name : ",i+1);
    scanf("%s",&stu[i].name);
    printf("\nenter marks :");
    scanf("%d",&stu[i].marks);
    printf("\n");
}
```

Q.16 Write a complete C program to compute and print the names of those students who have got more than 80 marks. Also print their marks along with their names. **(7)**

Ans: A C program to compute and print names of students who scored more than 80 marks

```
#include<stdio.h>
#include<conio.h>
struct student
{
   char name[10];
   int marks;
};
void main()
{
   int i,n;
   struct student stu[50];
   clrscr();
   printf("enter no. of students");
   scanf("%d",&n);
   printf("ENTER NAME AND MARKS\n");
   for(i=0;i<n;i++)
   {
      printf("%d. name : ",i+1);
      scanf("%s",&stu[i].name);
      printf("\nenter marks :");
      scanf("%d",&stu[i].marks);
      printf("\n");
   }
   for(i=0;i<n;i++)
   {
```

```
        if(stu[i].marks>80)
        {
           printf("%s",stu[i].name);
           printf("\t");
           printf("%d",stu[i].marks);
           printf("\n");
        }
     }
     getch();
}
```

Q.17 Write a recursive function in C to compute the value of x^n where n is a positive integer and x has a real value. **(7)**

Ans: A recursive function to compute the value of x^n, power(x,n) is given below:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   float x,y;
   int n;
   clrscr();
   printf("enter the value of x : ");
   scanf("%f",&a);
   printf("enter the value of n : ");
   scanf("%d",&n);
   y=power(x,n);
   printf("%f raise to power %d = %f",x,n,y);
   getch();
}
float power(float a, int b)
{
   float  k;
   if(b==1)
   return(a);
   else
   k=a*power(a,b-1);
   return(k);}
```

Q.18 Write a function 'exchange' to interchange the values of two variables say x and y. Illustrate the use of this function in calling program. Assume that x and y are defined as global variables. **(7)**

Ans: A C function exchange to interchange the values of x and y:

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

```
#include<math.h>
int i,j;
int x,y;
void main()
{
    i=10,j=20;
   clrscr();
   printf("The values before exchange is i: %d,
   j:%d\n",i,j);
   exchange();
   printf("The values after exchange is i: %d,
   j:%d\n",i,j);
   printf("\n");
   getch();
}
void exchange()
{ int temp;
   temp = x;
   x = y;
   y = temp;
 }
```

Q.19 Write a C function that returns 1 if the argument is a prime number and returns 0 otherwise. **(7)**

Ans: A C function prime(n) described below returns 1 if the argument is a prime number otherwise 0:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int n,flag;
   clrscr();
   printf("enter any number");
   scanf("%d",&n);
   flag=prime(n);
   if(flag==1)
   {
      printf("no. is prime");
      else
      printf("%d",flag);
      getch();
   }
   int prime(n)
   { int i;
      for(i=2;i<n;i++)
      {
         if(n%i==0)
```

```
            return 0;
        }
    }
     return 1;
}
```

Q.20 Write a C function for searching an element in an array of size N. Assume that elements in array are stored in ascending order. **(7)**

Ans: A C function to search an element in an array of size n:

```
rec(int a[],int low,int high,int m)
{
   int mid;
   mid=(low+high)/2;
   if(low<=high)
   {
      if(a[mid]==m)
         return 1;
      if(a[mid]>m)
         rec(a,low,(mid-1),m);
      if(a[mid]<m)
         rec(a,(mid+1),high,m);
   else
      printf("element is not present");
     }
}
```

Q.21 Explain the declaration int (*p (char *a))[10]; **(3)**

Ans: *int (*p (char *a))[10];*
interpretation of the above statement is:
*int (*p (char *a))[10];*
char *a :- pointer to character (Character Array)
(*p (Character Array))
int (*p (Character Array))
Integer pointer to character array
*int (*p (Character Array))[10]*
Array of integer pointer to character array
The given statement represent array of int pointer to char array.

Q.22 Suppose a member of a structure is a pointer variable. How can the object of the pointer be accessed in terms of structure variable name and the member name? **(3)**

Ans:

```
struct rec
{
```

```
        int *a;}obj;
```

to use pointer member with structure variable following operator is used.

obj->a;

that is ordinary variables are used with dot (.) operator and pointer variables are used with arrow (->) operator.

Q.23 What happens when a pointer to structure is incremented? What danger is associated with this type of operation? **(2)**

Ans: When pointer to structure is incremented then the pointer points to the next block of memory. As for example

```
struct rec
{
        int a;
        char b[10], *c;
};
main()
{
        struct rec *obj;
        obj=(struct rec *)malloc(sizeof(struct rec));
        obj->a=10;
        strcpy(obj->b, "XYZ");
        strcpy(obj->c, "ABC");
        printf ("\n%d\t%s\t%s",obj->a, obj->b, obj->c);
        //Incrementing pointer to structure
           obj++;
        obj->a=15;
        strcpy(obj->b, "PQR");
        strcpy(obj->c, "MNO");
        printf ("\n%d\t%s\t%s",obj->a, obj->b, obj->c);
}
```

In the above program, dynamic memory allocation is used for assigning one block of structure in obj. Later the pointer to structure (obj) is incremented, so that pointer points to the next block of memory.

Danger associated with increment of pointer to structure:

As in the above program structure member is a char pointer. When pointer to structure is incremented then it may results in memory overwrite, because the size for the pointer member is not defined.

Q.24 Distinguish between the following:

(i) Syntactic error and semantic error
(ii) Run time error and logical error
(iii) Compiler and Interpreter **(6)**

Ans:(i)Syntactic error and semantic error

Syntactic errors also known as compilation errors are caused by violation of the grammar rules of the language. The compiler detects, isolate these errors and give terminate the source program after listing the errors. Some of the common syntactic errors are:

- missing or misplaced ; or }
- missing return type for a procedure
- missing or duplicate variable declaration

Semantic errors are logical errors. If there is a semantic error in a program, it will run successfully, in the sense that the computer will not generate any error messages, but it will not do the right thing. The problem is that the meaning of the program (its semantics) is wrong. Identifying semantic errors can be tricky because it requires working backward by looking at the output of the program and trying to figure out what it is doing.

(ii) Run time error and logical error

Run-time errors: Errors such as mismatch of data types or array out of bound error are known as runtime errors. These errors are generally go undetected by the compiler so programs with run-time error will run but produce erroneous results.

Logical errors: These are the errors related with the logic of the program execution. These errors are not detected by the compiler and are primarily due to a poor understanding of the problem or a lack of clarity of hierarchy of operators. Such errors cause incorrect result.

(iii) Compiler and Interpreter

These are two types of language translators.

A **compiler** converts the source program (user-written program) into an object code (machine language by checking the entire program before execution. If the program is error free, object program is created and loaded into memory for execution. A compiler produces an error list of the program in one go and all have to be taken care even before the execution of first statement begin. It takes less time for execution.

An **interpreter** is also a language translator that translates and executes statements in the program one by one. It work on one statement at a time and if error free, executes the instruction before going to second instruction. Debugging is simpler in interpreter as it is done in stages. An interpreter takes more time for execution of a program as compared to a compiler.

Q.25 The skeletal outline of a C program is shown below:

```
    main()
{ FILE *p;
   int a;
   float b;
   char c;
   p = fopen("sample.dat", "r");
   …
   fclose(p);
}
Read the values of a, b and c from the data file and
display them on the screen
```

(3)

Ans:
```
#include<stdio.h>
#include<conio.h>
void main()
{
     FILE *p;
     int a;
     float b;
     char c;
     p=fopen("sample.dat","r");
      fscanf(p,"%d %f %c",&a,&b,&c);
      printf("a=%d b=%8.2f    c=%c", a,b,c);
      fclose(p);
       getch();
   }
```

Q.26 How the program design and program efficiency related to each other. **(2)**

Ans: Design is an important phase in software engineering. Two critical resources used in development process are execution time and memory. The efficiency of a program is measured in terms of these two resources. Inefficiencies in software design can result in complexity that is costly in software maintenance. Program efficiency can be improved with good program design and error free coding.

Q.27 Given the following program

```
main( )
{           static int a[5] = {10,20,30,40,50};
  void find (int *p);
  ….
  find (a);
  ….
}
void find (int *p)
{int j, sum = 0;
  for (j=3;j<5;++j) sum +=*(p+j);
  printf("sum = %d", sum);
  return;
}
```

(i) What kind of argument is passed to find?
(ii) What kind of value is returned?
(iii) What value is displayed by print statement within find? **(9)**

Ans :
```
#include<conio.h>
main()
{       static int a[5]={10,20,30,40,50};
       void find(int *p);
       clrscr();
```

```
        find(a);
        getch();
}
void find(int *p)
{
      int j, sum=0;
      for(j=3;j<5;++j)
      sum+=*(p+j);
      printf("sum=%d",sum);
}
```

(i) a pointer type variable is passed to function find .
(ii) The value returned is void.
(iii)The "printf" statement within find will display 90, which is sum of subscript 3 and 4, that is value 40 and 50 in the array a.

Q.28 What is top down design? Write down the steps to breakdown a problem into sub problems? **(8)**

Ans: A **top-down** design is essentially breaking down a system to gain insight into its compositional sub-systems. In a top-down approach an overview of the system is first formulated, specifying but not detailing any first-level subsystems. Each subsystem is then refined in greater detail, sometimes in many additional subsystem levels, until the entire specification is reduced to base elements. In short the top down approach means decomposing of the solution procedure into subtasks. This approach produces a readable and modular code that can be easily understood and maintained.

Steps for breaking a problem into sub-problems involve the following steps:
1. Start with a simple and short statement of the problem. This is top level in the design.
2. In the next level, describe the program as sequence, selection or repetition of main tasks. These tasks are called modules and these are complete in themselves. However these are described in greater details in next level. A module should have just one entry point and exit point.
3. This process is repeated step by step till all modules are refined. This step-wise refinement stops when there are sufficient details to convert that procedure into program.
4. Pseudo code or flowcharts are used to represent the procedure at intermediate steps.
5. Individual modules are tested at each level to ensure that they are working as per requirement.

Q.29 What are the qualities and capabilities of good algorithms? **(8)**

Ans: Every algorithm should have the following five capabilities and qualities:

1.Input: The algorithm should take zero or more input.
2. Output: The algorithm should produce one or more outputs.

3. Definiteness: Each and every step of algorithm should be defined unambiguously.
4. Effectiveness: A human should be able to calculate the values involved in the procedure of the algorithm using paper and pencil.
5. Termination: An algorithm must terminate after a finite number of steps.

Q.30 Design an algorithm that accepts a positive integer and reverses the order of its digits. **(8)**

Ans: An algorithm to find reverse of an integer is:

```
void reversein()
{
   int n,r;
   printf("enter an integer");
   scanf("%d",&n);
   printf("\nreverse of %d : ",n);
   while(n>0)
   {
      r=n%10;
      printf("%d",r);
      n=n/10;
   }
}
```

Q.31 Given a set of n numbers design an algorithm that adds these numbers and returns the resultant sum. **(8)**

Ans: An algorithm to add n numbers:

```
void main()
{
   int a[10],n,i,sum=0;
   printf("how many nos. u want to enter");
   scanf("%d",&n);
   printf("enter the nos.");
   for(i=0;i<n;i++)
   {
        scanf("%d",&a[i]);
        sum=sum+a[i];
   }
   printf("sum of these numbers = %d",sum);
}
```

Q.32 Determine the hierarchy of operations and evaluate the following expression:
kk = 3 / 2 * 4 + 3 / 8 + 3 **(4)**

Ans: kk=7 because 3/2*4+3/8+3 =1*4+0+3=7; 3/2 evaluates to 1, int/int=int and similarly 3/8 evaluates to 0.

Q.33 While purchasing certain items, a discount of 10% is offered if the quantity purchased is more than 1000. If the quantity and price per item are input through keyboard, write a program to calculate the total expenses. **(6)**

Ans:A C program to calculate the total expenses if the quantity and price per item are input through keyboard is:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int n,i,price,dis,p,q,sum=0;
   clrscr();
   printf("how many items u purchased");
   scanf("%d",&n);
   i=1;
   while(i!=n+1)
   {
      printf("enter the quantity of %d item",i);
      scanf("%d",&q);
      printf("enter price per item");
      scanf("%d",&p);
      if(q>1000)
      {
         dis=.1*p*q;
         price=(p*q)-dis;
      }
      else
      price=p*q;
      sum=sum+price;
      i++;
   }
   printf("total bill = %d",sum);
   getch();
}
```

Q.34 Write a program to copy input to output, replacing each string of one or more blanks by a single blank. **(6)**

Ans: A C program to copy input to output replacing each string of one or more blanks by a single blank is:

```
#include<stdio.h>
#include<conio.h>
void main()
{
 int i,j;
 char a[50],b[50];
 printf("enter the string");
```

```
    gets(a);
    i=0;
    j=0;
    while(a[i]!=' ')
    {
        b[j]=a[i];
        j++;i++;
    }
    b[j]=' ';
    j++;
    i++;
    while(a[i]!='\0')
    {
     while(a[i]!=' '&&a[i]!='\0')
     {
      b[j]=a[i];
      j++;
      i++;
     }
     while(a[i]==' ')
       i++;
    }
    b[j]='\0';
    printf("%s",b);
    getch();
}
```

Q.35 How many types of logical operators are there in 'C' programming language. **(4)**

Ans: C allows usage of three logical operators:

Operator	Symbol	Example	It evaluates to
AND	&&	exp1 && exp2	True (1) only if both exp1 and exp2 are true; false (0) otherwise
OR	||	exp1 || exp2	True (1) if either exp1 or exp2 is true; false (0) only if both are false
NOT	!	!exp1	False (0) if exp1 is true; true (1) if exp1 is false

For example:
(4 == 4) && (5!= 1) evaluates to True (1), because both operands are true.
(4 > 1) || (9 < 1) evaluates to True (1), because one operand is true (4>1).
!(5 == 4) evaluates to True (1), because the operand is false.

Q.36 Why do we use functions? Give the advance features of functions. **(4)**

Ans: We use functions due to following reasons:

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.37 Write a program to test if a character from the keyboard is a lower case letter. **(8)**

Ans: A C program to test if a character from the keyboard is a lower case letter:

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
void main()
{
   char a;
   clrscr();
   printf("\nenter any character :\n");
   a=getchar();
   if(islower(a)>0)
      printf("\nit is lower case letter");
   else
      printf("\nit is not a lower case letter");
   getch();
}
```

Q.38 What are the features of C preprocessor? Give the differences between macros and functions? **(6)**

Ans: A **pre-processor** is a program that processes the source code before it passes through the compiler. It operates under the control of preprocessor directive. These are placed in the source program before the main.

To define a macro, # define statement is used. This statement, also known as macro definition takes the following general form:

#define identifier string

The pre-processor replaces every occurrence of the identifier in the source code by the string. The preprocessor directive definition is not terminated by a semicolon. For example

#define COUNT 100 will replace all occurrences of COUNT with 100 in the whole program before compilation. Similarly we can define small functions with the help of macros. For example, a macro defined as

#define SQUARE(x) (x*x) will calculate square of argument when it is called. This is called macro definition.

Macros Vs Functions:

A macro's definition is expanded into the code each time the macro is encountered in the source code. If your program invokes a macro 100 times, 100 copies of the expanded macro code are in the final program. In contrast, a function's code exists only as a single copy. Therefore, in terms of program size, the better choice is a true function.

When a program calls a function, a certain amount of processing overhead is required in order to pass execution to the function code and then return execution to the calling program. There is no processing overhead in "calling" a macro. In terms of speed, a macro has the advantage.

Q.39 Write a function, which takes an array of real numbers and its size as arguments and returns the maximum. Using the above function write a program to read a set of real numbers from the keyboard and find the maximum number in the array. **(10)**

Ans: A C program to read a set of real numbers from the keyboard and find maximum among them using a max function:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int i,n;
   float a[100];
   clrscr();
   printf("\nhow many elements u want to enter :\n");
   scanf("%d",&n);
   printf("\nEnter the elements:");
   for(i=0;i<n;i++)
   scanf("%f",&a[i]);
   max(a,n);
   getch();
}

max(float a[],int n)
{
   int i;
   float k,large;
   large=a[0];
   for(i=1;i<n;i++)
   {
      if(a[i]>large)
```

```
            {
               k=a[i];
               a[i]=large;
               large=k;
            }
          }
          printf("largests element is   :   %f",large);
        }
```

Q.40 Define a pointer? Write a program to assign any number at random to an integer variable k and display the same through pointer. **(8)**

Ans: A pointer is a variable which contains the address in memory of another variable. We can have a pointer to any variable type. The unary or monadic operator **&** gives the "address of a variable". The indirection or dereference operator ***** gives the "contents of an object pointed to by a pointer".

A pointer is declared as follows:

```
int *ptr;
```

where *ptr is a pointer of int type.

A Program to display the value of an int variable through pointer is listed below"

```
#include<conio.h>
void main()
{
      int k;
      int *ptr;
      clrscr();
      k=10;
      ptr=&k;
      printf("\n Value of k is %d\n\n",k);
      printf("%d is stored at addr %u\n",k,&k);
      printf("%d is stored at addr %u\n",*ptr,ptr);
            *ptr=25;
      printf("\n Now k = %d\n",k);
      getch();
}
```

Output of the program is:

Value of k is 10

10 is stored at addr 65524

10 is stored at addr 65524

Now k=25.

Q.41 Write a program to read the coordinates of the end points of a line and to find its length. Use a structure variable named 'line' to store the relevant information about its end points. **(8)**

Ans: A Program to read end points of a line and finding its length is:

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

```
#include<conio.h>
#include<math.h>
struct lyn
{
   int a1;
   int b1;
   int a2;
   int b2;
};
void main()
{
   int a,b;
   float length;
   struct lyn line;
   clrscr();
   printf("enter the coordinates of end point A");
   scanf("%d%d",&line.a1,&line.b1);
   printf("\nenter the coordinates of end point B");
   scanf("%d%d",&line.a2,&line.b2);
   a=line.a1-line.a2;
   b=line.b1-line.b2;
   length=sqrt(pow(a,2)+pow(b,2));
   printf("length of the line is %f",length);
   getch();
}
```

Q.42 Explain library string functions. Write a program to copy the contents of the string "HELLO" to another string. **(8)**

Ans: Using C standard library functions, we can copy, concatenate, compare, and search strings. Some of the important Library string functions are described below:

(i) strcat() Function concatenates two strings together and has the following form:

strcat(string1,string2);

When this function is executed, string2 is appended to string1 by removing the null character at the end of string1.

C permits nesting of strcat functions as strcat(strcat(string1,string2),string3);

(ii) strcmp() is the string comparison function defined in string.h header file. It has the following form:.

int strcmp (const char *s1, const char *s2);

strcmp will accept two strings. It will return an integer. This integer will either be:

Negative if s1 is less than s2.

Zero if s1 and s2 are equal.

Positive if s1 is greater than s2.

Strcmp performs a case sensitive comparison; if the strings are the same except for a difference in case, then they're countered as being different. Strcmp also passes the address of the character array to the function to allow it to be accessed.

(iii) strcpy() function is just like a string-assignment operator which take the following form:
char *strcpy (char *dest, const char *src);
strcpy is short for string copy, which means it copies the entire contents of src into dest. The contents of dest after strcpy will be exactly the same as src such that strcmp (dest, src) will return 0.src may be a character array variable or a string constant.

A C program to copy the contents of string "HELLO" to another string:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   char str1[20],*str2;
   int m,i,flag=0,j;
   clrscr();
   printf("enter the 1st string");
   gets(str1);
   str2="Hello";
   printf("enter the index after which u want to insert Hello in
   1st : ");
   scanf("%d",&m);
   i=0;
   while(i<=m)
   {
      i++;
   }
   j=0;
   while(str2[j]!='\0')
   {
      str1[i]=str2[j];
      i++;
      j++;
      if(str1[i]=='\0')
         flag=1;
   }
   if(flag==1)
      str1[i]='\0';
      printf("%s",str1);
   getch();
}
```

Q.43 Explain the following types of errors, which are considered while testing programs.

1. Syntax errors.
2. Run-time errors.

3. Logical errors.
4. Latent errors. **(8)**

Ans:

(i) Syntax errors: Syntax errors also known as compilation errors are caused by violation of the grammar rules of the language. The compiler detects, isolate these errors and terminate the source program after listing the errors.

(ii) Run-time errors: Errors such as mismatch of data types or array out of bound error are known as runtime errors. These errors are generally go undetected by the compiler so programs with run-time error will run but produce erroneous results.

(iii) Logical errors: These are the errors related with the logic of the program execution. These errors are not detected by the compiler and are primarily due to a poor understanding of the problem or a lack of clarity of hierarchy of operators. Such errors cause incorrect result.

(iv)Latent errors: These are hidden errors which come into the picture only when a particular data set is used. For example, consider the following statement:

```
int z=x/x-y;
```

An error will occur when x and y are equal.

Q.44 Define the functions:
(i) fwrite() (ii) fread() **(6)**

Ans:

For binary File I/O we use fwrite and fread .

(i) fwrite( ): it write from memory into a file.The declaration is given below:

```
size_t fwrite(const void *ptr, size_t size_of_elements, size_t number_of_elements, FILE
*a_file);
```

(ii)fread( ): It is used for reading into the memory. The declaration is given below:

```
size_t fread(void *ptr, size_t size_of_elements, size_t number_of_elements, FILE *a_file);
```

Both take four arguments. The first argument is the name of the array or the address of the structure you want to write to the file. The second argument is the size of each element of the array; it is in bytes. For example, if we have an array of characters, the size_of_elements is one. sizeof operator can be used get the size of the various datatypes;
The third argument tells how many elements we want to read or write; for example, if it is a 100 element array, we will pass 100.The final argument is simply the file pointer we've been using. When fread is used, after being passed an array, fread will read from the file until it has filled the array, and it will return the number of elements actually read.

For example,

```
FILE *fp;
fp=fopen("c:\\test.bin", "wb");
char x[10]="ABCDEFGHIJ";
fwrite(x, sizeof(x[0]), sizeof(x)/sizeof(x[0]), fp);
```

Q.45 Give the difference between sprintf() and sscanf() functions. **(6)**

Ans:

sprintf: One can use this function as an alternative to "itoa". "sprintf" takes three arguments- char *variable to hold the converted number, a string containing a format specifier and the number to be converted into a string. "sprinf" returns the number of characters in the string (not included the null character). The following example converts a few numbers into string format, and prints out the result:

```
#include <stdio.h>
void main()
 {
     char str[10];
     int i;
     i = sprintf(str, "%o", 15);
     printf("15 in octal is %s\n",   str);
     printf("sprintf returns: %d\n\n", i);
     i = sprintf(str, "%d", 15);
     printf("15 in decimal is %s\n", str);
     printf("sprintf returns: %d\n\n", i);
}
Output:
15 in octal is 17
sprintf returns: 2
15 in decimal is 15
sprintf returns: 2
```

sscanf converts a string into various formats. It also takes three arguments- a char * variable that contains data to be converted, a string containing a format specifier that determines how the string is converted and a memory location to place the result of the conversion. "sscanf" returns the number of items converted. The following example performs conversions in various formats:

```
#include <stdio.h>
void main()
 {
     char* ints = "20, 40, 60";
     char* floats = "10.4, 24.66";
     int i;
     int n;
     float f;
     i = sscanf(ints, "%d", &n);
     printf("n: %d\n", n);
     printf("sscanff returns: %d\n\n", i);
     i = sscanf(floats, "%f", &f);
     printf("f: %f\n", f);
     printf("sscanff returns: %d\n\n", i);}
```

Output:
n: 20
sscanff returns: 1
f: 10.400000
sscanff returns: 1

Q.46 What are static and dynamic tools? **(4)**

Ans: Static and Dynamic tools:
There are two basic types of verification tools:

- Static verification tools examine the driver code without running the driver. Because these tools do not rely on tests that exercise the code, they can be extremely thorough. Theoretically, static verification tools can examine all of the driver code, including code paths that are rarely executed in practice. However, because the driver is not actually running, they can generate false-positive and false-negative results, that is, they can misinterpret the code, report an error in a code path that cannot occur in practice, or miss an error that can occur.
- Dynamic verification tools examine the driver code while the driver is running, typically by intercepting calls to commonly used driver support routine and substituting calls to their own error-checking versions of the same routines. Because the driver is actually running while the dynamic tools are doing the verification, false-positive results are rare. However, because the dynamic tools detect only the actions that occur while they are monitoring the driver, the tools can miss certain driver defects if the driver test coverage is not adequate.

The best practice is to use a combination of static and dynamic verification tools. Static tools allow you to check code paths that are difficult to exercise in practice, while the dynamic tools find serious errors that are occurring in the driver.

Q.47 Explain divide & conquer strategy to solve any problem. **(4)**

Ans: Divide and Conquer strategy: The original problem is solved by repeatedly solving a divided sub-problem that is of smaller size and can be solved more efficiently. The problem is split into smaller sub-problems recursively until we eventually reach a stage where sub-problem is small enough to break further and can be solved easily. This strategy is called Divide and Conquer strategy. Suppose a problem P is associated with a set S and an algorithm partitions S into smaller sets such that solution of the problem P is reduced to solution of its smaller sub-sets. Then this algorithm is called divide and conquer algorithm. For example, in searching for a key in an ordered list, the size of the list is halved after every search.

Q.48 Explain the common programming errors. **(6)**

Ans: Common programming errors:

- Missing semicolon: Every C statement must end with a semicolon. A missing semicolon is confusion to the compiler and may result in misleading error messages.
- Missing braces: Very common error as it is common to forget a closing brace. Number of opening braces should match number of closing braces.
- Undeclared variables: C requires declaration of variables before their use.

- Forgetting the precedence of operators: Expression are evaluated according to precedence of operators. It is very common for beginners to forget this.
- Mismatch of parameters in function calls: There may be mismatch in actual and formal parameters in function calls.
- Missing '&' operator in scanf call.
- Crossing the bounds of an array.
- Unending and sometimes wrong loops.
- Using uninitialized pointer that points to garbage.
- Improper comment characters.

Q.49 Write down steps, which are included in implementation of algorithms. **(6)**

Ans: The steps included in the implementation of algorithms are:
- Clear understanding of the problem and algorithm designed to solve that problem.
- Decide a set of input and derive the corresponding output.
- Choose an appropriate programming language to convert algorithm to program.
- Use appropriate data structures and identify various procedures.
- Make distinct and separate modules of the whole problem
- Using the chosen language, write clean source code and well-written comments.
- Debug separate modules and check their validity in terms if already designed input set and corresponding output.
- Integrate separate modules into one unit and test it as a whole.

Q.50 Write a C program to find the sum of the following series using a function declaration
sum = x – (x^3)/3! + (x^5)/5! – ... (x^n)/n! **(8)**

Ans: A C program to find the sum of Sine series is listed below:

```
#include<conio.h>
#include<math.h>
void main(){
 float x,dig,sum; int i,n; clrscr();
 printf("\n Enter the value of x : ");
 scanf("%f",&x);
 printf("\n Enter the value of n : ");
 scanf("%d",&n);       /*convert x into radians*/
 x=x*3.1412/180;
 sum=x;
 dig=x;
 for(i=1;i<=n;i++){
     dig=(dig*pow((double)(-1),(double)(2*i-
1))*x*x)/(2*i*(2*i+1));
       sum+=dig;}
 printf("\n The sum is : %6.2f",sum);getch();}
```

Q.51 Write an algorithm to generate Fibonacci series. **(10)**

Ans: An algorithm to generate nth member of Fibonacci sequence is:

1. Start
2. Scan the number 'n' upto which the series to be generated.
3. Initialize Sum=0, x=0, y=1 and i=3.
4. Print x and y as the part of series.
5. Repeat a to e until i<=n
   a. Calculate Sum=x+y
   b. Print Sum as part of series.
   c. Assign y to x
   d. Assign sum to y
   e. i=i+1
6. Stop

Q.52 What is meant by identifiers? How do identifiers differ from keywords? **(6)**

Ans: 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.53 What are the bitwise logical operators? **(4)**

Ans: Operators that are used for manipulation of data at bit level are known as bitwise operator. Bitwise logical operator are binary operator and require two integer type operand. These work on their operand bit by bit starting from the least significant bit. There are following three logical bitwise operators:

- Bitwise AND(&)
- Bitwise OR(|)
- Bitwise exclusive OR(^)

Results of all three logical bitwise operations are tabulated below:

Op1	Op2	Op1&Op2	Op1|Op2	Op1^Op2
1	1	1	1	0
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0

Q.54 Explain the following statements:
(i) getchar() (ii) putchar() **(6)**

Ans:

(i) getchar( ):

The function getchar() obtains the next character from the stream stdin. It provides buffered character input with echo, and its prototype is int getchar(void);

(ii) putchar( ):

The putchar() Function located in stdio.h, is as follows:
int putchar(int c);
This function writes the character stored in c to stdout. Although the prototype specifies a type int argument, we pass putchar() a type char. The function returns the character that was just written, or EOF if an error has occurred.
A program to demonstrate getchar() and putchar() is given below along with output and explanation:

```
#include <stdio.h>
main(){
   int ch;
   while ((ch = getchar()) != `\n')
        putchar(ch);}
```

Today is good weather
Today is good weather
When getchar() function is called, it waits to receive a character from stdin. Because getchar() is a buffered input function, no characters are received until Enter is pressed. However, each pressed key is echoed immediately on the screen. When we press Enter, all the entered characters , including the newline, are sent to stdin by the operating system. The getchar() function returns the characters one at a time, assigning each in turn to ch. Each character is compared to the newline character '\n' and, if not equal, displayed on-screen with putchar().When a newline is returned by getchar(), the while loop terminates.

Q.55 What is the difference between a structure declaration and a structure initialization? **(8)**

Ans:

A structure is defined as follows:

```
     struct coord {
       int x;
       int y;
     };
```

The struct keyword, which identifies the beginning of a structure definition, must be followed immediately by the structure name, or tag (which follows the same rules as other C variable names).Within the braces following the structure name is a list of the structure's member variables. The preceding statements define a structure type named coord that contains two integer variables, x and y. They do not, however, actually create any instances of the structure coord. In other words, they don't set aside any storage for the structures. There are two ways to declare structures. One is to follow
the structure definition with a list of one or more variable names, as is done here:

```
     struct coord {
        int x;
```

```
    int y;
} first, second;
```

These statements define the structure type coord and declare two structures, first and second, of type coord. first and second are each instances of type coord; first contains two integer members named x and y, and so does second. This method of declaring structures combines the declaration with the definition. The second method is to declare structure variables at a different location in your source code from the definition. The following statements also declare two instances of type coord:

```
struct coord {
    int x;
    int y;
};
  struct coord first, second;
```

Structure members can be initialized using a dot operator as follows:

first.x=10;
first.y=20;

Q.56 Write a C program to declare a self referential structure using dynamic allocation and to display the content of the structure. **(8)**

Ans: A linked list is a self referential structure which contain a member field that point to the same structure type. In simple term, a linked list is collections of nodes that consist of two fields, one containing the information about that node, item and second contain the address of next node. Such a structure is represented as follows:

```
struct node
{      int item;
        struct node *next;  };
```

A program to demonstrate a self referential structure using dynamic allocation and display the content of the structure is:

```
//demonstrate a simple linear linked list
#include<stdio.h>
#include<stdlib.h>
#define NULL 0
struct linked_list
{
 int number;
 struct linked_list *next;
};
typedef struct linked_list node;
void main()
{
 node *head;
 void create(node *p);
 void print(node *p);
 clrscr();
 head=(node *)malloc(sizeof(node));
```

```
        create(head);
        printf("\n");
        print(head);
        getch();
      }
      void create(node *list)
      {
        printf("Input a number\n");
        printf("(type -999 to end) : ");
        scanf("%d",&list->number);
        if(list->number == -999)
              list->next=NULL;
        else{
              list->next=(node *)malloc(sizeof(node));
              create(list->next);}
        return;
      }
      void print(node *list)
      {
        if(list->next!=NULL){
              printf("%d - ->",list->number);
        if(list->next->next == NULL)
              printf("%d",list->next->number);
        print(list->next);     }
        return;
}
```

Q.57 Write a program to find the sum & average of the given numbers using the do----while loop. **(6)**

Ans: A C program to find the sum and average of given numbers using do-while loop is:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
   int i=0,n;
   float sum,avg,num;
   clrscr();
   sum=0;
   printf("How many numbers you want to find sum\n");
   scanf("%d",&n);
   printf("Enter the numbers\n");
   do{
       scanf("%f",&num);
        sum=sum+num;
        i++;}while(i<n);
   avg=sum/n;
```

```
printf("Sum=%f\n",sum);
printf("Average=%f\n",avg);
getch();}
```

Q.58 What is a function? List out the advantages & disadvantages of using functions in C? **(4)**

Ans: 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.

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.

Disadvantages of using functions in C code:

When a program calls a function, a certain amount of processing overhead is required in order to pass execution to the function code and then return execution to the calling program. If we are calling a small function again and again, this processing overhead results in inefficiency.

Q.59 When passing parameters to functions, explain the difference between pass-by-value and pass-by-reference. **(4)**

Ans: *The two ways of parameter passing mechanism are:*

(i) Pass by value or sending the values of the arguments- The value of each of the actual arguments in the calling function is copied into corresponding formal arguments of the called function. The changes made to the formal arguments have no effect on the values of actual arguments in the calling function. This technique of passing arguments is called pass by value illustrated by the following example.

(ii) Pass by reference or sending the addresses of the arguments- the addresses of actual arguments in the calling function are copied into formal arguments of the called function. Using these addresses we are actually working on actual argument so changes will be reflected in the calling function. This technique of passing arguments is called pass by reference, illustrated by following example.

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
   int i=10,j=20;
   clrscr();
   printf("The values before swap is i: %d, j:%d\n",i,j);
   swapv(i,j);
   printf("The values after swap is i: %d, j:%d\n",i,j);
   printf("\n");
   swapr(&i,&j);
   printf("The values after swap is i: %d, j:%d\n",i,j);
   printf("\n");
   getch();
}
swapv(int x,int y)
{ int temp;
   temp=x;
   x=y;
   y=temp;
  }
swapr(int *x,int *y)
{
  int temp;
   temp=*x;
   *x=*y;
   *y=temp;
  }
```

The value of i and j is 10 and 20 only after calling function swapv, that is call by value. However the result of calling swapr(), call by reference is i=20 and j=10.

Q.60 Write a program to find the sum of given non-negative integers using a function declaration. Sum = 1+2+3+4+-------n **(8)**

Ans: A Program to find the sum of given non-negative integers using a function declaration:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
              int Sum( int [],int );
void main()
{
 int i,sum,n;
 int a[100];clrscr();
 printf("How many numbers you want to enter\n");
 scanf("%d",&n);
 printf("Enter Nos.\n");
```

```
 for(i=0;i<n;i++)
    scanf("%d",&a[i]);
 sum=Sum(a,n);
 printf("\n Sum of the %d number is %d\n",n,sum);
 getch();
}
          int Sum(int [],int n)
{
   int i,sum=0;
   for(i=0;i<n;i++)
      sum=sum+a[i];
 return(sum);
}
```

Q.61 What is an array & how is an array variable different from an ordinary variable? **(4)**

Ans:An array is a collection of data storage locations, each having the same data type and the same name. Each storage location in an array is called an array element. A particular value is indicated by writing a number called index or subscript after array name. For example a[5] stands for 6^{th} element in the array a. The complete set of values is called array while the individual values are called elements. Arrays can be of any variable type.

Q.62 Define an *union* which contains two members-**colour** which is an array of 12 characters and **size** which is int. Define two union variables and assign values to them. **(4)**

Ans: Union definition

```
union  item
{
              char colour[12];
         int size;
};
```

Union enables us to treat the same space in memory as a number of different variables. We can define a union member as
union item item1;
We cannot assign different values to the different union elements at the same time. We can do Item1.size=5; or item1.colour[0]=50;

Q.63 Write a program to read a set of lines from the keyboard & remove the white spaces such as tab, space, carriage return, new line, line feed & vertical tab from the text & display onto the screen. **(8)**

Ans: A program to read a set of lines from keyboard and display onto the screen after deleting white spaces is as follows:

```
#include<stdio.h>
#include<conio.h>
void main()
```

```
{
 char a[100],b[100];
 int i=0,j=0;
 clrscr();
 printf("enter the set of lines");
 gets(a);
 while(a[i]!='\0')
 {
  while(a[i]!=' '&&a[i]!='\t'&&a[i]!='\0')
  {
     b[j]=a[i];
     j++;
     i++;
   }
   while(a[i]==' '||a[i]=='\t')
     i++;
 }
 b[j]='\0';
 printf("%s",b);
 getch();
}
```

Q.64 Distinguish between the following functions when operating on files:

- **i.** rewind and ftell
- **ii.** printf and fprintf
- **iii.** feof and ferror
- **iv.** getc and putc **(8)**

Ans:

(i) rewind and ftell

rewind( ) sets the position to the beginning of the file. It also takes a file pointer and reset the position to the start of the file. For example:

rewind(fp);

n=ftell(fp); would assign 0 to n because file pointer has been set to the start of the file by rewind.

ftell( ): gives the current position in the file from the start. ftell takes a file pointer and returns a number of type long, that corresponds to the current position. For example:

n=ftell(p); would give the relative offset n in bytes of the current position. This means that n bytes have already been read (or written).

(ii) printf anf fprintf

The printf() function is the most versatile function in C standard library that is used by a program to display data on-screen. For example, to print a text message on-screen, call the printf() function, passing the desired message enclosed in double quotation marks as follows:

printf("Nice weather!");

fprintf is just like printf with the only difference that this work only on file. The general syntax for fprintf is:

fprintf(fp,"control string",list);

For example:

fprintf(fp,"%s %d",name,age);

where fp is the file pointer of file opened in write mode. Control string contains output specifications for the items in the list. list is the names of the variables. In the example, name and age are the variables. %d and %s are the control strings.

(iii)feof and ferror

feof function is used to test for an end of file condition. It takes a file pointer as an argument and returns a non zero if all of the data from the specified file has been read and a zero otherwise. For example:

if**(feof(fp))**
printf("End of data.");

ferror function reports the status of the file indicated. It also performs in the similar manner. It takes a file pointer as an argument and returns a non zero if and error has been indicated upon that point. It returns a zero otherwise. For example:

if**(ferror(fp)!=0)**
printf("An error has occurred.");

(iv)getc and putc

getc and putc funtions are used to handle one character at a time. getc reads a character from the file that is opened in read mode. For example:

c=getc(fp2);

this statement will read a character from the file whose file pointer is fp2.

The putc function is used to write a character to the file. For example:

putc(c,fp2);

this statement will write the character stored in the character variable c to the file whose file pointer is fp2.

Q.65 Write a program to copy the contents of one string to another string using a pointer method. **(8)**

Ans: A C program to copy the contents of one string to another using a pointer method is:

```
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
#define length 50
void main()
{
 char *s1,*s2,c;
 int i=0;
 clrscr();
 s1=(char*)malloc(length*sizeof(char));
 s2=(char*)malloc(length*sizeof(char));
```

```
 printf("enter string\n");
 gets(s1);
 while((c=*(s1+i))!='\0')
 {
  s2[i]=c;
  i++;
 }
 s2[i]='\0';
 printf("Copied string is\n");
 printf("%s",s2);
 getch();
}
```

Q.66 What is meant by conditional compilation? **(4)**

Ans: C pre-processor offers a feature called conditional compilation that is used to switch on or off a particular line or group of lines in a program. Compiler skips over part of a source code by inserting the pre-processing commands #ifdef and #endif, which have the general form as:

```
#ifdef macro
        Stmt1;
        Stmt2;
#endif
```

If macro has been #defined, the block of code will be processed as usual; otherwise not.

Q.67 What are the different functions involved in ctype.h? **(6)**

Ans:ctype.h: The different functions involved in this header are character testing functions and conversion functions. For example

- isalpha(c) returns an int type data and determine if the argument c is alphabetic. It returns nonzero value if true; 0 otherwise.
- toascii(c): is a conversion function defined in ctype.h that converts value of argument to ascii.

Other character testing functions in ctype.h are:
Isalnum(c), isascii(c),isdigit(c),islower(c),isspace(c),isupper(c),isxdigit(c),ispunct(c) etc.
Other conversion functions in ctype.h are tolower(c),toupper(c).

Q.68 Write a program to read a set of values from the keyboard using a pointer structure operator & to display the contents of the structure onto the screen. **(6)**

Ans: A program to read a set of values using a pointer structure operator and to display the contents of the structure onto the screen:

```
#include<stdio.h>
#include<conio.h>
struct student
{
```

```
  char name[20];
  int roll_no;
};
void main()
{
  struct student stu[3],*ptr;
  clrscr();
  printf("\n Enter data\n");
  for(ptr=stu;ptr<stu+3;ptr++)
  {printf("Name");
    scanf("%s",ptr->name);
    printf("roll_no");
    scanf("%d",&ptr->roll_no);
  }
  printf("\nStudent Data\n\n");
  ptr=stu;
  while(ptr<stu+3)
  {
     printf("%s   %5d\n",ptr->name,ptr->roll_no);
     ptr++;
    }
getch();
}
```

Q.69 How problem definition phase plays an important role in terms of the problem-solving aspect? Justify your answer. **(8)**

Ans: Problem definition plays an important role in terms of the problem-solving aspect. Before start attacking any problem, the problem should be completely understood and that can be done by defining the problem precisely. One should be sure to specify all necessary user interface, data structures, input and corresponding output. In problem definition phase, working should be done with major details, and main tasks can be divided into subtasks. In this way one can go to the minute details slowly. Algorithm is designed and it can be verified in view of predefined inputs and corresponding output. If problem is defined precisely, then one can evaluate the relationships between the data elements, find out the operations that must be performed on the logically related data elements and also one can identify the easy and efficient operation. Program verification can also be done efficiently only if the problem is defined properly.

Q.70 What is an algorithm? Write an algorithm to compute factorial of a number n where $n \geq 0$. **(8)**

Ans: Algorithm is the most fundamental concept in computer science. An algorithm is an ordered sequence of well-defined effective operations which will produce an output given an input and terminate in a finite amount of time. Algorithms can be expressed in many kinds of notation, including natural languages, pseudocode, flowcharts, and programming languages. Natural language expressions of algorithms is verbose and are rarely used for complex or

technical algorithms; while pseudocode and flowcharts are structured ways to express algorithms and also independent of a particular implementation language. Most algorithms are intended to be implemented as programs.

An algorithm for computing factorial of a number is as follows:

1) Start
2) Scan the number 'n' for which factorial is to be calculated.
3) Initialize i=1.
4) Initialize fact=1..
5) Repeat a to b until i<=n
   a. Calculate fact=fact*i.
   b. i=i+1
6) Print the value of fact.
7) Stop.

Q.71 Write a program that reads a character from the keyboard and then prints it in the reverse case i.e. if the input is in upper case, the output will be in lower case and vice versa. **(8)**

Ans:
```
#include<conio.h>
void main()
{
 char ch;
 clrscr();
 printf("\n Enter a character: ");
 scanf("%c",&ch);
 if(ch>=65 && ch<=90)
       ch+=32;
 else if(ch>=97 && ch<=122)
       ch-=32;
 else
       printf("\n The entered character is not an
alphabet!!");
 printf("\n The entered character in reverse case is :
%c",ch);
 getch();
}
```

Q.72 Design an algorithm to evaluate the function sin(x) as defined by the infinite series expansion

$$\sin(x) = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots. \quad \textbf{(8)}$$

Ans:
```
#include<conio.h>
#include<math.h>
void main()
{
 float x,dig,sum;
```

```
 int i,n;
 clrscr();
 printf("\n Enter the value of x : ");
 scanf("%f",&x);
 printf("\n Enter the value of n : ");
 scanf("%d",&n);
 /*convert x into radians*/
 x=x*3.1412/180;
 sum=x;
 dig=x;
 for(i=1;i<=n;i++)
 {
     dig=(dig*pow((double)(-1),(double)(2*i-
1))*x*x)/(2*i*(2*i+1));

       sum+=dig;
 }
 printf("\n The sum is : %6.2f",sum);
 getch();
}
```

Q.73 Write short notes on following:

i. Compilation errors.

ii. Linker errors. **(6)**

Ans:

(i) Compilation errors: Compilation errors are caused by violation of the grammar rules of the language. The compiler detects, isolate these errors and terminate the source program after listing the errors. These are of two types:

a. Syntax errors -- Common syntax errors include
   missing or misplaced ; or },
   missing return type for a procedure,
   missing or duplicate variable declaration.

b. Type errors -- These include
   type mismatch on assignment,
   type mismatch between actual and formal parameters.

(ii) Linker errors: Errors such as mismatch of data types or array out of bound error are known as Linker errors or run. These errors are generally go undetected by the compiler so programs with run-time error will run but produce erroneous results. Following are the types:

a. Output errors -- the program runs but produces an incorrect result. This indicates an error in the meaning of the program (logic error).

b. Exceptions -- the program terminates abnormally. Examples include
   division by zero,
   null pointer,

out of memory.

Q.74 Write a program to read an integer number from the keyboard, add 1 to it if the number read is even & again add 1 to it if the number is less than 20. Otherwise, keep the number unchanged. **(6)**

Ans:
```
#include<stdio.h>
#include<conio.h>
void main()
{
   int n;
   clrscr();
   printf("enter a number");
   scanf("%d",&n);
   if(n%2==0)
   { n=n+1;
    if(n<20)
      n=n+1;
   }
   printf("The Number is:%d",n);
   getch();
 }
```

Q.75 Give the difference between #include <stdio.h> and #include "stdio.h". **(4)**

Ans:#include "filename": The search for the file is made first in the current directory and then in the standard directories as mentioned in the include search path.
#include <filename>: This command would look for the file in the standard list of directories.
Both of these directives cause the entire contents of filename to be inserted into the source code at that point in the program.

Q.76 Explain the use of default in switch block with the help of an example? **(5)**

Ans: The general form of switch statement is as follows:
```
switch (expression)
{
            case value-1:
            block-1
            break;
    case value-2:
            block-2
            break;
      …
      …
    default:
            default block;
```

```
            break;

 }
```

When switch is executed, the value of the expression is successively compared against the value value-1, value-2,…. If a case is found whose value matches with the value of the expression, then the block of statements that follow the case are executed. The default is an optional case. When present, it will be executed if the value of the expression does not match with any of the case values. If it is not present, no action will take place if all matches fail and control will come out of the switch block.

Q.77 Write a program to read a set of real numbers & find the maximum & the minimum number entered with the help of do.. while loop. **(6)**

Ans:

```
#include<conio.h>
     float largest(float*,int);
     float smallest(float*,int);
     void main()
     {
      float a[20],k,l;
      int n,i,m;
      clrscr();
      do
      {
       printf("how many numbers u want to enter");
       scanf("%d",&n);
       printf("enter numbers");
       for(i=0;i<n;i++)
       scanf("%f",&a[i]);
       k=largest(a,n);
       l=smallest(a,n);
       printf("largest element  =  %f",k);
       printf("\nsmallest element =  %f",l);
       printf("\npres 1 to perform again  :");
       scanf("%d",&m);
      }while(m==1);
      getch();
     }
     float largest(float a[], int n)
     {
      float large,t;
      int i=1;
      large=a[0];
      do
      {
       if(a[i]>large)
       {
        t=a[i];
```

```
    a[i]= large;
    large=t;
   }
   i++;
  }while(i<n);
  return (large);
}
float smallest(float a[], int n)
{
  float small,t;
  int i=1;
  small=a[0];
  do
  {
   if(a[i]<small)
   {
    t=a[i];
    a[i]=small;
    small=t;
   }
   i++;
  }while(i<n);
  return (small);
}
```

Q.78 Give the difference between break & continue statement. **(5)**

Ans:

The **break** command will exit the most immediately surrounding loop regardless of what the conditions of the loop are. Break is useful if we want to exit a loop under special circumstances.

```
#include <stdio.h>
void  main()
{
   int a;
    printf("Pick a number from 1 to 4:\n");
   scanf("%d", &a);
    switch (a)
   {
     case 1:
       printf("You chose number 1\n");
       break;
     case 2:
       printf("You chose number 2\n");
       break;
     case 3:
       printf("You chose number 3\n");
```

```
            break;
          case 4:
            printf("You chose number 4\n");
            break;
          default:
            printf("That's not 1,2,3 or 4!\n");
        }
        getch();
      }
```

Continue is another keyword that controls the flow of loops. If we are executing a loop and hit a continue statement, the loop will stop its current iteration, update itself (in the case of for loops) and begin to execute again from the top. Essentially, the continue statement is saying "this iteration of the loop is done; let's continue with the loop without executing whatever code comes after me."

The syntax of continue statement is simple

continue;

Continue cannot be used with switch, like break.

Q.79 Write a program to generate 100 random numbers lying in the range 0.0000 to 1.0000. Evaluate their mean with the help of a function. **(8)**

Ans:

```
#include<stdlib.h>
main()
{
 float a=0,mean=0;
 int i=0;
 for (i=0;i<100;)
 {
       a=rand();
       if (a>0 && a<9999)
       {   a=a*0.0001;
            printf ("%.4f ",a);
            mean+=a;
            i++;
       }
 }
 mean/=100;
 printf ("\nMean of above mentioned numbers=%f",mean);
 getch();
}
```

Q.80 Define a function & explain why function prototype is essential. **(8)**

Ans: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.

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.

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.

Q.81 Write a program to sort an array of real numbers in ascending order. **(8)**

Ans:
```
#include<stdio.h>
#include<conio.h>
void main()
{
 float a[20];
 int i,j,n,c,flag;
 clrscr();
 printf("how many numbers u want to enter :\n");
 scanf("%d",&n);
 printf("\nenter the numbers :\n");
 for(i=0;i<n;i++)
 scanf("%f",&a[i]);
 for(i=0;i<n-1;i++)
 {
  for(j=0;j<n-1-i;j++)
  {
   if(a[j]>a[j+1])
```

```
      {
       c=a[j];
       a[j]=a[j+1];
       a[j+1]=c;
       flag=0;
      }
     }
     if(flag)
      break;
     else
      flag=1;
   }
   printf("sorted elements :\n");
   for(i=0;i<n;i++)
   printf("%f\t",a[i]);
   printf("\n");
   getch();
  }
```

Q.82 Define the following:

(i) Auto variables. (ii) Static variables.
(iii) External variables. (iv) Register variables. **(8)**

Ans:(i)Auto variables: The features are as follows

Declaration place:-declared inside a function in which they are to be utilized, that's why referred as local or internal variables.

Declaration syntax:- A variable declared inside a function without storage class specification by default is an automatic variable. However, we may use the keyword auto to declare it explicitly.

```
main()
{
     auto int age;
}
```

Default initial value:-Garbage value

Scope:-created when the function is called and destroyed on exit from the function.

Life:- till the control remains within the block in which defined.

(ii) Static variables: The features are as follows

Declaration place:-may be declared internally or externally.

Declaration syntax:-we use the keyword static to declare a static variable.

Static int age;

Default initial value:- Zero

Scope:-in case of internal static variable, the scope is local to the function in which defined while scope of external static variable is to all the functions defined in the program.

Life:- value of variable persists between different function calls.

(iii) External variables: The features of variables are as follows

Storage: Memory
Default initial value: Zero
Scope: global
Life: As long as program execution does not come to an end

(iv)Register variables: The features of variables are as follows
Storage: CPU registers
Default initial value: Garbage value
Scope: Local to the block in which defined
Life: till the control remains within the block in which defined

Q.83 What is a macro & how is it different from a pre processor? **(6)**

Ans:A **macro** is a preprocessor directive. A pre-processor is a program that processes the source code before it passes through the compiler. It operates under the control of preprocessor directive. These are placed in the source program before the main. The preprocessor directive can be divided into three categories:
Macro substitution directives,
File inclusion directives.
Compiler control directives.

Q.84 Differentiate between #include Directive & # define Directive. **(4)**

Ans: To define a macro, # define statement is used. This statement, also known as macro definition takes the following general form:
#define identifier string
The pre-processor replaces every occurrence of the identifier in the source code by the string. The preprocessor directive definition is not terminated by a semicolon. For example
#define COUNT 100 will replace all occurrences of COUNT with 100 in the whole program before compilation.
An external file containing functions or macro definitions can be included as part of the program by using preprocessor directive
#include "filename" where filename is the name of the file containing the required definitions or functions. By using "#include" , the preprocessor inserts the entire contents of the filename into the source code of the program.

Q.85 Write a program to read any number x & to evaluate $x^2 - 2$. Use a function to evaluate $x^2 - 2$. Call this function through its pointer. **(6)**

Ans:
```
#include<conio.h>
#include<math.h>
void main()
{
 int x,val;
 clrscr();
```

```
printf("\n Enter the value of x : ");
scanf("%d",&x);
val=(pow(x,2)-2);
printf("The value of (x^2 - 2) is : %d",val);
getch();
}
```

Using a function:

```
#include<conio.h>
void main()
{
 int x,c;
 clrscr();
 printf("enter the value of x : ");
 scanf("%d",&x);
 c=(power(x,2)-2);
 printf("x^2-2 = %d",c);
 getch();
}
power(int a,int b)
{
 int k;
 if(b==1)
 return(a);
 else
 k=a*power(a,b-1);
 return(k);
}
```

Q.86 Explain in detail most commonly used dynamic memory allocation functions. **(8)**

Ans: Most commonly used dynamic memory allocation functions:

malloc( ): It is a memory allocation function that allocates requested size of bytes and returns a pointer to the first byte of the allocated space. The malloc function returns a pointer of type void so we can assign it to any type of pointer. It takes the the following form:

ptr= (cast type *) malloc(byte-size);

where ptr is a pointer of type cast-type. For example, the statement

x=(int *) malloc(10 *sizeof(int)) means that a memory space equivalent to 10 times the size of an int byte is reserved and the address of the first byte of memory allocated is assigned to the pointer x of int type.

The malloc function can also allocate space for complex data types such as structures. For example:

ptr= (struct student*) malloc(sizeof (struct student)); where ptr is a pointer of type struct student.

calloc( ): It is another memory allocation function that allocates space for an array of elements, initializes them to zero and then returns a pointer to the memory. This function is normally used for requesting memory space at run time. It takes the following form:

ptr= (cast type *) calloc(n, element-size);

This statement allocates contiguous space for n blocks, each of size element-size bytes.

realloc( ): realloc is a memory allocation function that modifies the size of previously allocated space. Sometime it may happen that the allocated memory space is larger than what is required or it is less than what is required. In both cases, we can change the memory size already allocated with the help of the realloc function known as reallocation of memory. For example, if the original allocation is done by statement

ptr= malloc(size);

then reallocation is done by the statement

ptr=realloc(ptr,newsize); which will allocate a new memory space of size newsize to the pointer variable ptr and returns a pointer to the first byte of the new memory block.

Q.87 Write a program to create a link list of a set of integer numbers entered in the ascending order and add a new number to the linked list, such that the numbers in the new list also remain in the ascending order. **(8)**

Ans:
```
#include <stdio.h>
#include <conio.h>
#include <alloc.h>
/* structure containing a data part and link part */
struct node
{
   int data ;
   struct node *link ;
};
void add ( struct node **, int ) ;
void display ( struct node * ) ;
int count ( struct node * ) ;
void delete ( struct node **, int ) ;
void main( )
{
   struct node *p ;
   p = NULL ;  /* empty linked list */
   add ( &p, 15 ) ;
   add ( &p, 1 1) ;
   add ( &p, 16 ) ;
   add ( &p, 14 ) ;
   add ( &p, 17 ) ;
   clrscr( ) ;
   display ( p ) ;
   printf ( "\nNo. of elements in Linked List = %d", count ( p )
  ) ;
}
/* adds node to an ascending order linked list */
void add ( struct node **q, int num )
```

```
{
   struct node *r, *temp = *q ;
   r = malloc ( sizeof ( struct node ) ) ;
   r -> data = num ;
 /* if list is empty or if new node is to be inserted before the
first node */
 if ( *q == NULL || ( *q ) -> data > num )
 {
       *q = r ;
       ( *q ) -> link = temp ;
 }
 else
 {
       /* traverse the entire linked list to search the position
to insert the
           new node */
     while ( temp != NULL )
     {
               if ( temp -> data <= num && ( temp -> link ->
data > num || temp -> link == NULL ))
         {
         r -> link = temp -> link ;
         temp -> link = r ;
         return ;
          }
   temp = temp -> link ;  /* go to the next node */     } }
}
```

Q.88 Which programming design approach is followed by 'C' language (TOP-DOWN/BOTTOM-UP). Justify. **(3)**

Ans: C language follows Top down programming design approach. In C, the user thinks of a problem in terms of function modules or blocks. C is a structured language lends itself to top down approach. The essence of top down approach is to divide the whole problem into number of independent tasks called modules which are further divided into sub modules an so on. These modules form the basis of functions in the program.

Q.89 Evaluate the following expression. Show the hierarchy displaying all steps during evaluation

(i) int i = 2*3/4+4/4+8–2+5/8
(ii) int k = 3/2*4+3/8+3
(iii) float s = q*a/4–6/2+2/3*6/g (q = 4, a=2, g=2); **(6)**

Ans:

(i) output is 8 as
i=2*3/4+4/4+8-2+5/8
=1+1+6+0

=8

(ii) output is 7 as

k=3/2*4+3/8+3

=1*4+0+3

=4+0+3

=7

(iii) output is -1.000000

s=q*a/4-6/2+2/3*6/g

=4*2/4-6/2+2/3*6/2

=8/4-3+0* 3

=2-3

=-1

Since k is in float so output is -1.000000

Q.90 What is the difference between Testing & Debugging? Explain different Debugging Techniques. **(7)**

Ans:Program testing is the process of checking program, to verify that it satisfies its requirements and to detect errors. These errors can be of any type-Syntax errors, Run-time errors, Logical errors and Latent errors. Testing include necessary steps to detect all possible errors in the program. This can be done either at a module level known as unit testing or at program level known as integration testing.

Debugging is a methodical process of finding and reducing the number of bugs in a computer program making it behave as expected. One simple way to find the location of the error is to use print statement to display the values of the variables. Once the location of the error is found, the error is corrected and debugging statement may be removed.

Different debugging techniques are:

- To place print statements throughout the program to display the values of variables.
- Conditional compilation can be used to switch on or off debugging statements.
- Elimination and refinement: location of error is arrived by listing the possible causes of the error.
- Backtrack: The incorrect result is backtracked through the program logic until error is located.

Q.91 Enumerate the steps that need to be taken to design efficient algorithms. **(6)**

Ans: Steps that need to be taken to design efficient algorithms:

1. For any algorithm, the first step should be to prove that it always returns the desired output for all legal instances of the problem.

2. Second step to analyze an algorithm is to determine the amount of resources such as time and storage necessary to execute it. Usually the efficiency or complexity of an algorithm is stated as a function relating the input length to the number of steps (time complexity) or storage locations (space complexity). In theoretical analysis of algorithms it is common to estimate their complexity in asymptotic sense, i.e., to estimate the complexity function for reasonably

large length of input. Big O notation, omega notation and theta notation are used for this purpose.
There are many techniques for solving a particular problem. We must analyze these algorithms and select the one which is simple to follow, takes less execution time and produces required results.

Q.92 Replace the if-else statements by conditional operators

```
main( )
{
     int code;
     scanf("%d", &code);
     if(code>1)
        printf("\nJerusalem");
     else
        if(code<1)
           printf("\nEddie");
        else
           printf("\nC Brain");
}
```

Ans:
Replacement of if-else statements by conditional operator:
code>1?printf("\nJerusalem"):code<1?printf("\nEddie"):printf("\nBrain");

Q.93 Write a program to determine whether a given number is an Armstrong number or not. (Hint: $1^3+5^3+3^3 = 153$) **(4)**

Ans: A program to determine if a given number is an Armstrong number or not:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
 int n,r,k,sum=0;
 clrscr();
 printf("enter any number");
 scanf("%d",&n);
 k=n;
 while(n>0)
 {
  r=n%10;
  sum=sum+pow(r,3);
  n=n/10;
 }
 if(sum==k)
 printf("%d is an armstrong number",k);
```

```
  else
  printf("%d is not an armstrong number",k);
  getch();
 }
```

Q.94 (i) Write a C program to read two matrices and display their sum. **(8)**
(ii) What is meant by the scope of variables and summarise the variables types of storage class in C? **(8)**

Ans: A C program to read two matrices and display their sum is given below:

```
#include<stdio.h>
#include<conio.h>
void main()
{
 int m,n,a[10][10],b[10][10],c[10][10],i,j;
 clrscr();
 printf("enter the value for no.of rows");
 scanf("%d",&m);
 printf("enter the value for no. of columns");
 scanf("%d",&n);
 printf("enter the elements for matrix A :\n");
 for(i=0;i<m;i++)
 {
     for(j=0;j<n;j++)
     scanf("%d",&a[i][j]);
 }
 printf("enter the elements for matrix B :\n");
 for(i=0;i<m;i++)
 {
     for(j=0;j<n;j++)
     scanf("%d",&b[i][j]);
 }
 printf("MATRIX A :\n");
 for(i=0;i<m;i++)
 {
  for(j=0;j<n;j++){
      printf("%d",a[i][j]);
      printf("\t");}
  printf("\n");
 }
 printf("MATRIX B :\n");
 for(i=0;i<m;i++){
  for(j=0;j<n;j++){
      printf("%d",b[i][j]);
      printf("\t");}
  printf("\n");
 }
```

```
 printf("sum of two matrix :\n");
 for(i=0;i<m;i++)
 {
  for(j=0;j<n;j++)
  {
      c[i][j]=a[i][j]+b[i][j];
      printf("%d",c[i][j]);
      printf("\t");
  }
  printf("\n");
 }
 getch();
}
```

(ii): The scope of a variable determines the region of the program in which it is known. An identifier's "visibility" determines the portions of the program in which it can be referenced—its "scope." An identifier is visible only in portions of a program encompassed by its "scope," which may be limited to the file, function or block in which it appears.

File scope: The variables and functions with file scope appear outside any block or list of parameters and is accessible from any place in the translation unit after its declaration. Identifier names with file scope are often called "global" or "external." The scope of a global identifier begins at the point of its definition or declaration and terminates at the end of the translation unit. A function has file scope.

Function scope: A label is the only kind of identifier that has function scope. A label is declared implicitly by its use in a statement. Label names must be unique within a function however a label having the same name in two different functions is allowed.

Block scope: The variables with block scope appear inside a block or within the list of formal parameter declarations in a function definition. It is visible only from the point of its declaration or definition to the end of the block containing its declaration or definition. Its scope is limited to that block and to any blocks nested in that block and ends at the curly brace that closes the associated block. Such identifiers are sometimes called "local variables."

There are four storage classes in C:

a. Automatic storage class: The features of variables are as follows
   - -Storage: Memory
   - -Default initial value: Garbage value
   - -Scope: Local to the block in which defined
   - -Life: till the control remains within the block in which defined.

b. Register storage class: The features of variables are as follows
   - -Storage: CPU registers
   - -Default initial value: Garbage value
   - -Scope: Local to the block in which defined
   - -Life: till the control remains within the block in which defined

c. Static storage class: The features of variables are as follows
   - -Storage: Memory
   - -Default initial value: Zero
   - -Scope: Local to the block in which defined

-Life: value of variable persists between different function calls.

d. External storage class: The features of variables are as follows

-Storage: Memory
-Default initial value: Zero
-Scope: global
-Life: As long as program execution does not come to an end

Q.95 Write 'switch' statement that will examine the value of an integer variable flag & print one of the following messages:

(i) HOT, if flag=1
(ii) LUKE WARM, if flag=2
(iii) COLD, if flag=3
(iv) OUT OF RANGE, if any other value **(5)**

Ans: A program to demonstrate switch statement to display given messages:

```
void main(){
    int flag;
    printf( "Enter any value\n" );
    scanf( "%d", &flag );
    switch ( flag ) {
        case 1:            /* Note the colon, not a
semicolon */
     printf( "HOT\n" );
     break;
 case 2:
     printf( "LUKE WARM\n" );
            break;
        case 3:
     printf( "COLD\n" );
            break;
        default:
     printf( "OUT OF RANGE\n" );
     break;    }
    getchar();}
```

Q.96 What does 'return' statement do in a function? Can a function have more than one return statement? Explain. Write the user-defined code for finding factorial( ) of a given number using Recursion. **(6)**

Ans: 'return' statement in a function is used to return a value to the calling program. The return statement takes one of the following forms:

return; It does not return any value and act as a closing brace of the function.

or

return (expression); It returns the value of the expression to the called function.

A function can have more than one return statement. This happens when the value returned is based on certain conditions. For example:

```
if(x<=y)
  return x;
else
  return y;
```

All functions by default return int type data.

A program for finding factorial of a given number using recursion is as follows:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,c;
clrscr();
printf("enter the number   :");
scanf("%d",&n);
c=fact(n);
printf("factorial of %d = %d",n,c);
getch();
}
fact(int n)
{
int factorial;
if(n==1||n==0)
return(1);
else
factorial=n*fact(n-1);
return (factorial);
}
```

Q.97 Define an array. Write a program for 2-D Matrix Multiplication using arrays. **(8)**

Ans: An **array** is a collection of data storage locations, each having the same data type and the same name. Each storage location in an array is called an array element. A particular value is indicated by writing a number called index or subscript after array name. For example a[5] stands for 6^{th} element in the array a. The complete set of values is called array while the individual values are called elements. Arrays can be of any variable type.

A C program for 2-D matrix multiplication:

```
#include<stdio.h>
#include<conio.h>
void main()
{
 int a[10][10],b[10][10],c[10][10],i,j,k,m,n,p,q;
 clrscr();
 printf("enter no. of rows and column for 1st matrix\n");
 printf("no.of rows\n");
 scanf("%d",&m);
 printf("no. of columns");
 scanf("%d",&n);
```

```
printf("\nenter no. of rowsand columns for 2nd matrix");
printf("\nno. of rows");
scanf("%d",&p);
printf("\nno. of columns");
scanf("%d",&q);
if(n==p)
{
 printf("enter elements for matrix A");
 for(i=0;i<m;i++)
 {
  for(j=0;j<n;j++)
  scanf("%d",&a[i][j]);
 }
 printf("enter elements for matrix B");
 for(i=0;i<p;i++)
 {
  for(j=0;j<q;j++)
  scanf("%d",&b[i][j]);
 }
 printf("\nMATRIX A:\n");
 for(i=0;i<m;i++)
 {
  for(j=0;j<n;j++)
  {
      printf("%d",a[i][j]);
      printf("\t");
  }
  printf("\n");
 }
 printf("\nMATRIX B:\n");
 for(i=0;i<p;i++)
 {
  for(j=0;j<q;j++)
  {
   printf("%d",b[i][j]);
   printf("\t");
  }
  printf("\n");
 }
 printf("\n");
 printf("MULTIPLICATION :\n");
 for(i=0;i<m;i++)
 {
  for(j=0;j<q;j++)
  {
   for(k=0;k<p;k++)
   {
```

```
            c[i][j]=c[i][j]+(a[i][k]*b[k][j]);
        }
        printf("%d",c[i][j]);
        c[i][j]=0;
        printf("\t");
         }
     printf("\n");
     }
   }
   else
   printf("multiplication is not possible");
   getch();
 }
```

Q.98 Write an algorithm to find greatest common divisor of two positive non-zero integers. **(5)**

Ans: An algorithm to find greatest common divisor of two positive integers is given below:

```
 void gcd()
 {
  int a,b,r,h,k,c;
  clrscr();
  printf("enter two numbers");
  scanf("%d%d",&a,&b);
  h=a;
  k=b;
  while(r!=0)
  {
   r=a%b;
   a=b;
   b=r;
  }
  printf("G.C.D. of %d and %d = %d",h,k,a);}
```

Q.99 Point out and rectify the error(s), if any in the following code:

```
main( )
{
     int code, flag;
     if(code = = 1 & flag = = 0)
        printf(“/n The Eagle has landed);
}
```

(3)

Ans: In the problem bitwise AND (&) operator is used. In fact it should be logical AND '&&'.

Q.100 What would be the output of following code? **(4)**

```
main( )
{
     float a=13.5;
```

```
        float *b, *c;
        b =&a; // suppose the address of 'a' is 1006.
        c = b;
        printf("\n%u%u%u", &a, b, c");
        printf("\n%f %f%f%f%f", a, *(&a), *&a, *b, *c);
    }
```

Ans: The output of the given code assuming address of a is 1006 is:

1006 1006 1006

13.500000 13.500000 13.500000 13.500000 13.500000

The first printf statement giving the address of a, that is assigned to b and c also.
The second printf statement is giving the value at this address that is 13.5 printed as 13.500000 (taking 6 places after decimal)

Q.101 Write at least any 2 differences between malloc( ) & calloc( ) function. **(4)**

Ans: malloc( )and calloc( ):

malloc( ): It is a memory allocation function that allocates requested size of bytes and returns a pointer to the first byte of the allocated space. The malloc function returns a pointer of type void so we can assign it to any type of pointer. It takes the the following form:

ptr= (cast type *) malloc(byte-size);

where ptr is a pointer of type cast-type. For example, the statement

x=(int *) malloc(10 *sizeof(int)) means that a memory space equivalent to 10 times the size of an int byte is reserved and the address of the first byte of memory allocated is assigned to the pointer x of int type.

The malloc function can also allocate space for complex data types such as structures. For example:

ptr= (struct student*) malloc(sizeof (struct student)); where ptr is a pointer of type struct student.

calloc( ): It is another memory allocation function that allocates space for an array of elements, initializes them to zero and then returns a pointer to the memory. This function is normally used for requesting memory space at run time. It takes the following form:

ptr= (cast type *) calloc(n, element-size);

This statement allocates contiguous space for n blocks, each of size element-size bytes.

Q.102 What are the advantages of using pointers. **(4)**

Ans: The **advantages** of using a pointer are as follows:

- Through pointers we can access a variable that is declared outside a function.
- Data tables can be handled in an efficient manner by pointers.
- The length and complexity of a program is reduced.
- The execution speed in reduced.
- By using pointer to an array of character strings helps in saving of data storage space in memory.

Q.103 Differentiate between call by reference and call by value. Use suitable examples to explain. **(4)**

Ans: Call by value and Call by reference

Call by value means sending the values of the arguments- The value of each of the actual arguments in the calling function is copied into corresponding formal arguments of the called function. The changes made to the formal arguments have no effect on the values of actual arguments in the calling function. This technique of passing arguments is called call by value illustrated by swapv(int x, int y) function in the following example.

Call by reference means sending the addresses of the arguments- the addresses of actual arguments in the calling function are copied into formal arguments of the called function. Using these addresses we are actually working on actual argument so changes will be reflected in the calling function. This technique of passing arguments is called call by reference, illustrated by swapr(int *x,int *y) in following example.

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
 int i=10,j=20;
 clrscr();
 printf("The values before swap is i: %d, j:%d\n",i,j);
 swapv(i,j);
 printf("The values after swap is i: %d, j:%d\n",i,j);
 printf("\n");
 swapr(&i,&j);
 printf("The values after swap is i: %d, j:%d\n",i,j);
 printf("\n");
 getch();
}
swapv(int x,int y)
{ int temp;
   temp=x;
   x=y;
   y=temp;
  }
swapr(int *x,int *y)
{
  int temp;
   temp=*x;
   *x=*y;
   *y=temp;
  }
```

The value of i and j is 10 and 20 only after calling function swapv, that is call by value. However the result of calling swapr(), call by reference is i=20 and j=10

Q.104 Write macro definition for the following:

(i) Minimum of 2 values; MIN (a, b)

(ii) To check whether entered character is a digit (or) not; ISDIGIT(Y) **(6)**

Ans:

(i) A macro for finding minimum of two numbers:

```
#include<stdio.h>
#include<conio.h>
#define min(a,b) ((a>b)?b:a)
void main()
{
  .
  .
  .

  z=min(x,y);          //Statement in program can be used as
  ..}
```

(ii) A macro to check whether entered character is a digit or not:

```
#include<stdio.h>
#include<conio.h>
#define isdigit(a) ((48<=a&&a<=57)?1:0)
void main()
{
  .
  .
  .
  y=isdigit(x);          //Statement in program can be used
  as

  .
  .
  .
  .
}
```

Q.105 Write different built-in (library) functions provided by 'C' language for handling I/O operations on files. **(10)**

Ans: There are several functions used for I/O operations in files:

- **getc and putc functions** :

getc and putc funtions are used to handle one character at a time. getc reads a character from the file that is opened in read mode. For example:c=getc(fp2);

this statement will read a character from the file whose file pointer is fp2.

The putc function is used to write a character to the file. For example:putc(c,fp2);

this statement will write the character stored in the character variable c to the file whose file pointer is fp2.

- **getw and putw** :

These are integer oriented functions. Both work just like getc and putc with the difference it is used with integers only.

The general syntax is:

```
getw(fp2);
putw(integer,fp2);
```

In the above staments getw will read an integer from file whose file pointer is fp2. similarly putw will write the integer in file having file pointer fp2.

- **fprintf and fscanf** :

These two functions are just like printf and scanf with the only difference that these work only on files. The general syntax and examples are are the same for both fprintf and fscanf .

syntax:fprintf(fp,"control string",list);

example:fprintf(fp,"%s %d",name,age);

In the above syntax, fp is the file pointer of file opened in write mode. Control string contains output specifications for the items in the list. list is the names of the variables. In the above example, name and age are the variables. %d and %s are the control strings.

Q.106 What would be output of following code?

Justify your answer

(i)
```
f1( ){
    Static int count= 5;
    printf("\n count=%d", count- -);
    if(count != 0)
    f1( );}
```

(ii)
```
int i=0;
main( )
{
    printf("\main's i=%d", i);
    i++;
    val( );
}
val( ){
    i=100;
    printf("\n val's i=%d", i);
    i++;}
```

Ans:

(i) The output is:

count=5
count=4
count=3
count=2
count=1

Value of count is initially 5, it is printed and then count is decremented by 1.

Since count!=0 so function f1 is called again, 4 is printed and same process is repeated till count =0 so the above output.

(ii) The output is:

main's i=0
val's i=100

main function is calling global variable i with value=0 so first printf statement is printing 0 while in the function val ( ), the local variable takes precedence over global and so value 100 is printed.

Q.107 Write a C program to find if a number is present in a list of N numbers or not. **(10)**

Ans: A C program to find if a number is present in a list of N numbers or not:

```
#include<stdio.h>
#include<conio.h>
void main(){
        int i,n,m,flag=0; int a[10];        clrscr();
        printf("how many elements u want to enter");
        scanf("%d",&n);
        printf("enter element in the array");
        for(i=0;i<n;i++)
        scanf("%d",&a[i]);
        printf("enter the element u want to search");
        scanf("%d",&m);
       for(i=0;i<n;i++)   {
       if(a[i]==m)     {
          flag=1;
         break;  }}
     if(flag==0)
           printf("not present");
    else
           printf("present");
  getch();     }
```

Q.108 Write a C function using pointers to exchange the values stored in two memory locations in the memory. **(6)**

Ans: A C program to exchange the values stored in two memory locations:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int i,j;
void main(){
  i=10,j=20;clrscr();
 printf("The values before exchange is i: %d, j:%d\n",i,j);
 exchange(&i,&j);
 printf("The values after exchange is i: %d, j:%d\n",i,j);
 printf("\n");
 getch();}
exchange(int *x,int *y){ int temp;
   temp=*x;
```

```
*x=*y;
*y=temp;  }
```

Q.109 What are the different types of errors that can occur during I/O operations on a file? **(6)**

Ans: Typical error situations that may occur during I/O operations on a file:

- Trying to read beyond the end–of-file mark.
- Device overflow.
- Trying to use file that has not been opened.
- Opening a file with an invalid filename.
- Attempting to write to a write-protected file.
- Trying to perform an operation on a file when the file is opened for another type of operation.

Q.110 Write user-defined function for copying a string to another **(6)**

Ans: A program to copy a string to another:

```
#include<conio.h>
void main(){
 char str1[20],str2[20];
 int m,i,flag=0,j;
 clrscr();
 printf("enter the 1st string");
 gets(str1);
 printf("enter the 2nd string");
 gets(str2);
 printf("enter the index after which u want to insert 2nd
 string in 1st : ");
 scanf("%d",&m);
 i=0;
 while(i<=m){
     i++;}
 j=0;
 while(str2[j]!='\0'){
  str1[i]=str2[j];
  i++;
  j++;
  if(str1[i]=='\0')
  flag=1;
 }
 if(flag==1)
     str1[i]='\0';
     printf("%s",str1); getch();}
```

Q.111 Explain the salient features of typedef. **(6)**

Ans: Typedef statement allows user to define an identifier that would represent an exiting data type. The user-defined data type identifier can be used further to declare variables. It has the following syntax typedef datatype identifier; where datatype refers to exiting data type and identifier is the new name given to this datatype. For example typedef int nos; nos here symbolizes int type and now it can be used later to declare variables like nos num1,num2,num3;

Q.112 Write a program in C to find the sum and average of the given numbers stored in an array of n values. **(8)**

Ans: A C program to find the sum and average of the given numbers stored in an array of n values:

```
       #include<stdio.h>
#include<conio.h>
         void main()     {
     int i,a[100],n,sum; float average; clrscr();
     printf("enter the value of n less than 100");
     scanf("%d",&n);
     printf("enter  values: \n");
     for(i=0;i<n;i++)
        scanf("%d",&a[i]);
     sum=0;
     for(i=0;i<n;i++)   {
             sum=sum+a[i];     }
     printf("\n%d",sum);
     average=(float)sum/n;
     printf("\n%f",average);
     getch();     }
```

Q.113 Explain goto statement

Ans: C support **"goto"** statement to branch unconditionally from one point to another in the program.The goto keyword is followed by a label, which is basically some identifier placed elsewhere in the program where the control is to be transferred.During running of a program, the statement like "goto label1;" cause the flow of control to the statement immediately following the label "label1". We can have a forward jump or a backward jump.

```
     #include <stdio.h>
     void main() {
      int attempt, number = 46;
      looping: /* a label */
       printf("Guess a number from 0-100\n");
       scanf("%d", &attempt);
       if(number==attempt) {
        printf("You guessed correctly!\n\n");  }
       else {
        printf("Let me ask again...\n\n");
```

```
goto looping; /* Jump to the label*/  }}
```

Q.114 Name and explain the primitive or fundamental data types that Compilers support. **(4)**

Ans: Fundamental data types: Typical primitive types may include-

- Character (`char`)-usually stored in 8 bits(one byte) of internal storage; unsigned char have values between 0 to 255 while signed chars range from -128 to 127.
- Integer (`int`, `short`, `long`, `byte`) with a variety of precisions; C has three classes of internal storage, namely short int, int, and long int, in both signed and unsigned forms.
- Floating-point number (`float`, `double`, `real`, `double precision`); These are stored in 32 bits, with 6 digits of precision.
- Boolean having the values true and false.
- Reference (also called a pointer or handle), a small value referring to another object's address in memory.

Q.115 Write an algorithm to find the sum of the squares of the first n positive integers.

$1^2 + 2^2 + \ldots\ldots\ldots\ldots\ldots\ldots + n^2$ **(8)**

Ans: An algorithm to find sum of the squares of the first n positive integers is given below:

```
void SumSq()
{
 int i,n,sum=0;
 printf("enter the length of the series");
 scanf("%d",&n);
 for(i=1;i<=n;i++)
     sum=sum+i*i;
 printf("sum of squares upto %d = %d",n,sum);
}
```

Q.116 Write an algorithm to find the value of x^n where n is a positive integer greater than 1. **(8)**

Ans: An algorithm power( ) given below finds the value of x^n

```
void power()
{
 int a,b,i,result=1;
 printf("enter value of x :");
 scanf("%d",&x);
 printf("enter value of n :");
 scanf("%d",&n);
 printf("x raise to power n = ");
 for(i=1;i<=n;i++)
 result=result*x;
 printf("%d",result);
 }
```

Q.117 Write an algorithm to find the highest common factor of two positive numbers. **(8)**

Ans: An algorithm to find highest common factor of two positive numbers:

```
void main(){
    int a,b,r,h,k,c;
    printf("enter two numbers");
    scanf("%d%d",&a,&b);
       h=a;
       k=b;
     while(r!=0)    {
        r=a%b;
        a=b;
        b=r;     }
       printf("H.C.F. of %d and %d = %d",h,k,a);
}
```

Q.118 Name two types of program testing and explain them in detail. **(8)**

Ans: Program testing:**White box testing** strategy deals with the internal logic and structure of the code. White box testing also known as glass, structural, open box or clear box testing, tests code written, branches, paths, statements and internal logic of the code etc.
In order to implement white box testing, the tester has to deal with the code and hence is needed to possess knowledge of coding and logic i.e. internal working of the code. White box test also needs the tester to look into the code and find out which unit/statement/chunk of the code is malfunctioning. White box testing is applicable at unit, integration and system level however it is mainly done at unit level.
Black box testing Black-box test design treats the system as a "black-box", so it doesn't explicitly use knowledge of the internal structure. It takes an external perspective of the test object to derive test cases. These tests can be functional or non-functional, though usually functional. The test designer selects valid and invalid input and determines the correct output. There is no knowledge of the test object's internal structure.
This method of test design is applicable to all levels of software testing: unit, integration, functional testing, system and acceptance.

Q.119 Explain the following operators with the help of examples
(i) Conditional operator. (ii) sizeof operator.
(iii) Increment operator. (iv) Cast operator. **(4 × 3)**

Ans:
(i) Conditional operator: It is a ternary operator of the form
exp1? exp2: exp3 where exp1,exp2 and exp3 are expressions. It means that (condition)?(evaluate if condition was true):(evaluate if condition was false). For example:
my_variable = (x > 10)?"a":"b";
To accomplish the same using a standard if/else statement, would take more than one line of code as given below:
if (x > 10) {

```
 my_variable = 'a';
}
else {
my_variable = 'b';}
```

(ii)sizeof operator: It is a compile time operator and when used with an operand, it returns the number of bytes the operand occupies.

```
int m=sizeof(operand1);
```

so integer m will have number of bytes operand1 occupies.

(iii)Increment Operator: This is a unary operator '++' adds 1 to the operand with which it is used. That is ++num is same as num=num+1.

Increment operator is of two types-postfix and prefix operator which mean the same thing when form independent statements. However when used in expression on the right-hand side of an assignment, these behave differently. For example:

```
(i)   int i,j;
        i=2;
        j=++i;
(ii)  int i,j;
        i=2;
        j=i++;
```

In case (i), the value of j and i would be 3 because prefix operator first add 1 to the operand then assign it to the variable on left. While in case (ii), the value of j is 2 and i would be 3 because postfix operator first assign the value to the variable on the left and then increment the value.

(iv)Cast operator: The general form of a cast operator in C is:

(type name) expression, used for casting an expression to type "type name". Type name is one of the standard C data types and expression may be a constant, variable or an expression. For example, the statement

```
X=(int) 5.6 will convert 5.6 to an integer by truncation.
Y=(int)(a+b), the result of a+b is converted to integer.
```

Q.120 Write a program to convert the given temperature in Fahrenheit to temperature in Celsius where cel = ((fah – 32) * 5/9); and Fahrenheit is denoted by fah. **(4)**

Ans: A program to convert the given temperature in Fahrenheit to Celcius:

```
#include<stdio.h>
#include<conio.h>
void main()
{
 float fah,cel;
 clrscr();
 printf("enter temp in fahrenheit");
 scanf("%f",&fah);
 cel=((fah-32)*5/9);
```

```
 printf("\ntemp in celsius = %f",cel);
 getch();
}
```

Q.121 Write a program to display the real, imaginary and equal roots of a quadratic equation $ax^2 + bx + c = 0$. **(8)**

Ans: A program to display real, imaginary and equal roots of a quadratic equation:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
 int a,b,c,disc;
 float root1,root2,x1,y1;
 clrscr();
 printf("enter the value of a,b,c");
 printf("\nA  :\t");
 scanf("%d",&a);
 printf("\nB  :\t");
 scanf("%d",&b);
 printf("\nC  :\t");
 scanf("%d",&c);
 disc=((b*b)-(4*a*c));
 if(disc==0)
 {
  printf("/nroots are  real and equal");
  root1=(-b)/(2*a);
  printf("\nroots of the equation are :");
  printf("\nROOT1 : %f",root1);
  printf("\nROOT2 : %f",root1);
 }
 if(disc>0)
 {
  printf("roots are real and distinct");
  root1=((-b)+sqrt(disc))/(2*a);
  root2=((-b)-sqrt(disc))/(2*a);
  printf("\n roots of the equation are :");
  printf("\nROOT1 : %f",root1);
  printf("\nROOT2 : %f",root2);
   }
if(disc<0)
{
 disc=disc*(-1);
 printf("roots are imaginary");
 x1=(-b)/(2*a);
```

```
        y1=sqrt(disc)/(2*a);
        printf("ROOT1 : ");
        printf("%f",x1);
        printf("+i");
        printf("%f",y1);
        printf("\nROOT2 : ");
        printf("%f",x1);
        printf("-i");
        printf("%f",y1);
      }
      getch();
      }
```

Q.122 What do you mean by a loop? Explain the difference between the do loop, while loop, and for loop with the help of an example. **(8)**

Ans: Loop is a control structure used to perform repetitive operation. Some programs involve repeating a set of instruction either a specified number of times or until a particular condition is met. This is done using a loop control structure. A program loop consists of two parts: Body of the loop and control statement. The control statement tests certain conditions and then decides repeated execution or termination of statements. Most real programs contain some construct that loops within the program, performing repetitive actions on a stream of data or a region of memory. There are several ways to loop in C described below:

1.While statement: The basic format of while statement is

```
    while (conditional expression)
    {
 ...block of statements to execute...
    }
```

The `while` loop continues to loop until the conditional `expression` becomes false. Once this expression become false, the control is transferred out of the loop. On exit, the program continues with the statement immediately after the body of the loop. For example:

```
i=0;
        while(i<10)
         {
               printf("Hello world\n");
      i++;  }
```

This statement will print "Hello world" 10 times in a new line and come out of the loop when 'i' become 10.

2. Do statement: This loop construct is of the form:

```
      do
        {    ...block of statements to execute...

      }while(conditional expression);
```

While construct checks the conditional expression before the loop is executed. Sometimes it is necessary to execute the body of the loop before the conditional expression is evaluated. Such

situations are handled by do-while loop construct. On reaching the do statement, the body of the loop is evaluated and at the end of the loop, the conditional expression is checked for true or false. If true,it continues to evaluate the body again and when condition become false, the control is transferred to the statement immediately after the while statement.For example:

```
   do
     {
         printf( "Input a character\n");
ch=getch( );
  }while(ch='n');
```

This segment of program reads a character from the keyboard until 'n' is keyed in.

3. **For statement:** This is another entry-controlled loop having a general form:

```
    for (expression_1; expression_2; expression_3)
    {
  ...block of statements to execute...
    }
```

The expression_1 is for initialization of the control variable. The condition is tested upon using the expression_2. If the condition is true, the body of the loop is executed. Then control is transferred back to the for loop expression_3 where control variable is incremented using an assignment statement and new value of variable is checked through expression_2 and process is repeated till the expression_2 is evaluated to be false. On the termination of loop, the execution continues with the statement immediately following the loop.

The `for` loop is a special case, and is equivalent to the following while loop:

```
expression_1;
   while (expression_2)
     {
  ...block of statements...
  expression_3;
     }
```

For instance, the statement in while lop takes the following form in the 'for' loop:

```
for (i = 1;  i <10;  i = i+1) {
  printf("Hello World\n");
}
```

Q.123 How are values initialized in one – dimensional array? Should the entire array be initialised in the definition? **(4)**

Ans: The values are initialized in one-dimensional array just like an ordinary variable. The general form of initialization of arrays is:

type array-name[size]={list of values};

The values in the list are separated by commas. For example:

int marks[6]={60,70,80,40,79,89};

It is not necessary to initialize all the values in the array. However If they are not given any values, they would be garbage values in it. For example:

int marks[6]={30, 70 ,80,40}; here marks is array of 6 integer where only 4 values are provided; for marks in rest two subject, it will take the garbage value.
If the array is initialized where it is declared, mentioning the dimension of the array is optional. For example:

int marks[]={56,89,67,34,89,90};

Q.124 Write a program to find the average of 10 real numbers in an array. **(4)**

Ans: A program to find average of 10 real numbers:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
 int i=0;
 float sum,avg,num;
 clrscr();
 sum=0;
 while(i<10)
 {
   scanf("%f",&num);
   sum=sum+num;
   i++;
 }
 avg=sum/10;
 printf("Sum=%f\n",sum);
 printf("Average=%f\n",avg);
 getch();
}
```

Q.125 Define a structure for a student having name, roll number and marks obtained in six subjects. Write a program to input the details for 20 students and print the details of the students who have scored more than 70% marks overall. **(12)**

Ans: A C program to input details for 20 students and print the details of students who have scored more than 70% marks overall is given below:

```
#include<stdio.h>
#include<conio.h>
struct student
{
 char name[10];
 int roll_no;
 int marks[6];
 int total;
 int per;
};
void main()
```

```
{
 struct student stu[20];
 int i,j,req;
 clrscr();
 for(i=0;i<20;i++)
 {
 stu[i].total=0;
 printf("enter data for %d students :",i+1);
 printf("\nenter name");
 scanf("%s",stu[i].name);
 printf("\nenter roll no   ");
 scanf("%d",&stu[i].roll_no);
 printf("\nenter marks in subjects\t");
  for(j=0;j<6;j++)
  {
  printf("\nenter marks in %d subject\t",j+1);
  scanf("%d",&stu[i].marks[j]);
  stu[i].total=stu[i].total+stu[i].marks[j];
  }
  stu[i].per=stu[i].total/6;
  printf("\n");
 }
 for(i=0;i<20;i++)
 {
  if(stu[i].per>70)
  {
   printf("\nSTUDENT    %d",i+1);
   printf("\nname  :");
   printf("%s",stu[i].name);
   printf("\nroll no");
   printf("%d",stu[i].roll_no);
   for(j=0;j<6;j++)
   {
    printf("\nmarks in %d subject\t",j+1);
    printf("%d",stu[i].marks[j]);
   }
   printf("\nTOTAL    :%d",stu[i].total);
  }
 }
 getch();
}
```

Q.126 Print the following sequence of integers 1, 3, 9, 27, 81, 243 using (i) For statement (ii) While statement **(8)**

Ans: A C Program to print the sequence using (i) for statement (ii) while statement

(i)
```
#include<stdio.h>
```

```
#include<conio.h>
#include<math.h>
void main(){
     int i,n,l,k;
     clrscr();
     for(i=0;i<=5;i++){
         k=pow(3,i);
         printf("%d\t",k);}
     getch();}
(ii)#include<stdio.h>
#include<conio.h>
#include<math.h>
void main(){
     int i,n,l,k;
     clrscr();
     i=0;
     while(i<=5){
        k=pow(3,i);
        printf("%d\t",k);
        i++;}
getch();}
```

Q.127 If p is a pointer to an address and j is a numeric value, than what does the expression like p + j mean. **(2)**

Ans: If p is a pointer to an address and j is a numerical value, then p+j points to an address j locations after the current location.

Q.128 List the four stages involved in program design. **(2)**

Ans: Four stages involved in program design are:

- Understanding the problem and algorithm designed for the problem.
- Decide what is to be given as input and corresponding output.
- Choose an appropriate programming language and data structures.
- Derive test cases and test exhaustively each and every module of the program.

Q.129 How are the data elements initialized in a multidimensional array? What is the scope of rules for the multidimensional array. **(8)**

Ans: Multidimensional array: Multidimensional arrays can be described as "arrays of arrays". For example, a bidimensional array can be imagined as a bidimensional table made of elements, all of them of a same uniform data type.

int arr[3][5]; represents a bidimensional array of 3 per 5 elements of type `int`.

Similarly a three dimensional array like

int arr[3][4][2]; represent an outer array of three elements , each of which is a two dimensional array of four rows, each of which is a one dimensional array of five elements.

Multidimensional arrays are not limited to two ot three indices (i.e., two dimensional or three dimensional). They can contain as many indices as needed. However the amount of memory needed for an array rapidly increases with each dimension. For example:
char arr [100][365][24][60][60];declaration would consume more than 3 gigabytes of memory.
Memory does not contain rows and columns, so whether it is a one dimensional array or two dimensional arrays, the array elements are stored linearly in one continuous chain. For example, the multidimensional array

```
int arr[3][4][2]= {
                    {  {1,2},{3,4},{5,6},{7,8} },
                  {  {9,1},{1,2},{3,7},{4,7} },
                   {  {6,1},{18,19},{20,21},{22,23} },
                   };  is stored in memory just like an one-dimensional array shown below:
```

1	2	3	4	5	6	7	8	9	1	1	2	3	7	4	7	6	1	18	19	20	21	22	23
101							**117**								**133**								

Multidimensional arrays are just an abstraction for programmers, since we can obtain the same results with a simple array just by putting a factor between its indices.

Q.130 List any four different bitwise operators available in 'C' language. **(6)**

Ans: Bitwise operator: These permit the programmer to access and manipulate individual bits within a piece of data. These operators can operate upon ints and chars but not on floats and doubles. 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

X^Y=0000 0000 0001 0100

Q.131 Explain the method of program verification briefly. **(8)**

Ans: Method of program verification

- is used for answering the question :Have we built the software right? That is does it match the specification?
- Verification is the job of the developer.
- They use testing techniques for verification. In verification various reviews, inspections, and walkthroughs are used.
- The developer does Verification that the module or the project is doing the same as for the intended.

Reviews are made to check whether program goal has been achieved or not. One should distinguish between the reviews made about program resources use, and checking whether a system model or technical proposal is correct. These are two different review processes; one is to achieve correctness and other to ensure effective utilization of resources.

Inspections are made to evaluate program's qualitative features. Roles are allocated to people involved in inspection-Programmer whose product is under review, an inspector who evaluate and a moderator who control the review process.

Walkthrough are usually made to detect errors in the system. It is a procedure that is commonly used to check the correctness of the program produced by structured system analysis.

Q.132 Design an algorithm to compute summation of a set of numbers. **(8)**

Ans: A C algorithm to compute the summation of a set of numbers:

```
void main()
{
int i,sum,n;
int a[100];
clrscr();
printf("How many numbers you want to enter\n");
scanf("%d",&n);
printf("Enter Nos.\n");
for(i=0;i<n;i++)
   scanf("%d",&a[i]);
sum=Sum(a,n);
printf("\n Sum of the %d number is %d\n",n,sum);
getch();
}
int Sum(int a[],int n){
   int i,sum=0;
   for(i=0;i<n;i++)
      sum=sum+a[i];
 return(sum);
}
```

Q.133 Explain the top-down technique for algorithm design. **(8)**

Ans: Top-down design is a strategy that can be applied to take the solution of the computer problem from a vague outline to a precisely defined algorithm and program implementation. The design suggests that the general statements about the solution can be taken one at a time, and can be broken down into a set of more precisely defined subtasks. The subtasks can be implemented as program statements.
To make computer solutions to problems appropriate data structure should be selected. The data structure is selected by considering the points like easy searching, updating, excessive use of storage, recovering of earlier state in the computation etc.
For implementation of subtasks the iterative constructs, or loops and structures are used. To construct any loop, three things must be taken into account, the initial conditions, the invariant relation, termination.

Q. 134 Write an algorithm to reverse the digits of an integer. **(8)**

Ans:
l. Establish n, the positive integer to be reversed.
2. Set the initial condition *for* the reversed integer dreverse.
3. While the integer being reversed is greater than zero do
   (a) use the remainder function to extract the rightmost digit of the number being reversed;
   (b) increase the previous reversed integer representation dreverse by a factor of 10 and add to it the most recently extracted digit to give the current dreverse value;
   (c) use integer division by 10 to remove the rightmost digit from the number being reversed.

Q.135 Explain the following statements
(i) for loop
(ii) while loop
(iii) do-while loop

(8)

Ans:
(i) The for loop is entry-controlled loop that provides a more concise loop control structure. The general form of the for loop is
for(initialization; test-condition; increment) {
body of the loop}

1. initialization -The initialization of the control variable is done first, using assignment statements such as i = 1 and count = ( ).
2. test-condition - The value of the control variable is tested using the test-condition. The test-condition is a relational expression, such as i<10 that determines when the loop will exit. If the condition is true, the body of the loop is executed; otherwise the loop is terminated and the execution continues with the statement that immediately follows the loop.
3. increment - the control variable is incremented using an assignment statement.

(ii) While loop
while (test condition){

body of the loop}

The test condition is evaluated and if the condition is true, then the body of the loop is executed. If the condition is false, the control is transferred out of the loop.

(iii) Do while do

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

The program evaluate the body of the loop first. 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. The process continues as long as the condition is true. If the condition becomes false, the loop will be terminated.

Q.136 Write an algorithm to generate prime numbers in the first n positive integers.**(8)**

Ans:

1. Initialize and write out the first 3 primes. Also initialize the square of the 3rd prime.
2. Initialize x to 5.
3. While x less than n do
   (a) get next x value excluding multiples of2 and 3;
   (b) if not past end of multiples list then
      (b.1) if x square of largest prime then
      (1.a) include next prime multiple as its square,
      (1.b) update square by squaring next prime > x
   (c) while have not established x is non-prime with valid prime multiples do
      (c.1) wile current prime multiple is less than x, increment by current prime value doubled,
      (c.2) do prime test by comparing x with current multiple;
   (d) if current x prime then
      (d.1) write out x and if it is less than n store it.

Q.137 Write an algorithm to find the square root of a number. **(8)**

Ans: An algorithm for computing square root of a number is as follows:

1. Start
2. Initialise tol=0.000005
3. Scan the number 'val' for which square root is to be calculated.
4. Initialize oldval=val
   Initialize newval = (oldval + val/oldval)/2
5. while (fabs((newval-oldval)/newval) > tol)
   oldval = newval;
   newval = (oldval + value/oldval)/2;
6. Print "Square root of given number is:" newval
7. Stop

Q.138 Write a program to find the sum of odd and even numbers between 1 and 100.**(8)**

Ans:
```
#include<stdio.h>
main( )
{
int oddsum=( ), evensum=( ), I;
for (i=l; i <= 100; i+=2)
{
  oddsum += i;
  evensum += i+ 1;
}
printf("\n\n sum of odds is %d and evens=%d\n",
oddsum,evensum);
}
```

Q.139 Define pointers and discuss the advantages & disadvantages of pointers. **(6)**

Ans: Advantages of pointers are:

1. Function cannot return more than one value. But when the same function can modify many pointer variables and function as if it is returning more than one variable.
2. Flexible memory management: Allocating and deallocating memory as needed during run time allows you to create large objects, such as arrays, quickly and immediately free the memory when it is no longer required. In the case of arrays, we can decide the size of the array at runtime by allocating the necessary space.
3. Pointers provide a performance advantage by allowing you to access computer memory directly.
4. Pointers are not just for objects in memory; they can also be used for functions, thus allowing a function to be passed as a parameter to another function.

Disadvantages of pointers:

1. If sufficient memory is not available during runtime for the storage of pointers, the program may crash.
2. If the programmer is not careful and consistent with the use of pointers, the program may crash.
3. Direct access to memory means you can do things that perhaps you should not. Sometimes unintentionally (or intentionally) access to memory that is not yours, you could overwrite critical memory, modify the code of a running application, or cause your application or another application to behave or exit
unexpectedly.

Q.140 Write a program to swap two elements using pointer concept. **(6)**

Ans:
```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int i,j;
void main( )
{
i=10, j=20;
```

```
clrscr0;
printf("The values before swap is i: %d, j:%d\n",i,j);
swap(&i,&j);
printf("The values after swap is i: %d,j:%d\n",i,j);
printf("\n");
getch( );
}
swap(int *x,int *y)
{ int temp;
   temp=*x;
   *x=*y;
   *y=temp;
}
```

Q.141 Classify the different types of programming errors. **(4)**

Ans: Programming errors can be classified broadly into following categories:
(i) **Compilation errors**: Compilation errors are caused by violation of the grammar rules of the language. The compiler detects, isolate these errors and terminate the source program after listing the errors.
These are of two types:
a. Syntax errors -- Common syntax errors include
missing or misplaced; or },
missing return type for a procedure,
missing or duplicate variable declaration.
b. Type errors -- These include
type mismatch on assignment,
type mismatch between actual and formal parameters.

(ii) **Linker errors**: Errors such as mismatch of data types or array out of bound error are known as Linker errors or run. These errors are generally going undetected by the compiler so programs with run-time error will run but produce erroneous results. Following are the types:
a. Output errors -- the program runs but produces an incorrect result. This indicates an error in the meaning of the program (logic error).
b. Exceptions -- the program terminates abnormally. Examples include
division by zero,
null pointer,
out of memory.
Some common programming errors are listed below:

- Missing semicolon: Every C statement must end with a semicolon. A missing semicolon is confusion to the compiler and may result in misleading error messages.
- Missing braces: Very common error as it is common to forget a Closing brace. Number of opening braces should match number of closing braces. .
- Undeclared variables: C requires declaration of variables before their use. .
- Forgetting the precedence of operators: Expression are evaluated according to precedence of operators. It is very common for beginners to forget this.

- Mismatch of parameters in function calls: There may be mismatch in actual and formal parameters in function calls.
- Missing '&' operator in scanf call.
- Crossing the bounds of an array.
- Unending and sometimes wrong loops.
- Using uninitialized pointer that points to garbage.
- Improper comment characters.

Q.142 What are the input and output functions used with the files. **(4)**

Ans: Input functions used in files are getc( ) and fscanf( )

getc( ) It is used to read a character from a file that has been opened in read mode.

fscanf( ) It works in the same way as scanf( ). It has one more parameter placed as first, which is the data file pointer.
fscanf(filej)ointer, "control string", list);

put( ) It has two parameters, a character expression and the file pointer declared as FILE. This prints one character by character into the file referenced by the file pointer.
putc(c, fileyointer);

fprintf( ) It works in the same way as printf( ). It has one more parameter placed at first, which is the data file pointer. This file pointer can be user-defined or the standard output device file stdout.
fprintf(file yointer, "control string", list);

Q.143 Write a menu driven program to find perimeter of rectangle, circle, square, triangle. **(8)**

Ans:

```
/* PROGRAM TO FIND PERIMETER OF RECTANGLE, CIRCLE,
 SQUARE, TRIANGLE */
#include<stdio.h>
# include<math.h>
#define pi 3.14
main( );
{
float 1, b, r, side;
int a, c;
printf(" 1 for rectangle");
printf(" 2 for circle");
printf(" 3 for square");
printf(" 4 for triangle");
scanf("%f  %f  %f  %f ", &1, &b, &r, &side);
switch( c)
{
case 1: printf("%f ", 1 * b);
     break;
case 2: printf("%f ", pi * r * r);
     break;
case 3: printf("%f ", 4 * side);
```

```
     break;
case 4: printf("%f ", (a + b + c)/2);
     break;
}
}
```

Q.144 Differentiate between Structure and Array. **(4)**

Ans:
(i) An array is a collection of related data elements of same type. Structure an have elements of different types.
(ii) An array is derived data type whereas a structure is a programmer-defined one.
(iii) Any array behaves like a built-in data type. Declare an array variable field use it. In the case of structure, first, data structure is to be designed and declared before the variables of that type are declared and used.

Q.145 Define a function? Explain Call by value and Call by reference. **(8)**

Ans:A function is a self-contained block of statements.
Function include the following elements function name, type, list of parameters, local variable declarations, function statements and a return statement. function_type function_name(parameter list)
{
return statement;
}
The technique used to pass data from one function to another is known as parameter passing. Parameter passing can be done in 2 ways.
1. Call by value (pass by value)
2. Call by reference (pass by pointers)
In call by value, values of actual parameters are copied to the variables in the parameter are copied to the variables in the parameter list of the called function. The called function works on the copy and not on the original values of the actual parameters.
In call by reference, the memory addresses of the variables rather than the copies of values are sent to the called function. The called function directly works on the data in the calling function and the changed values are available in the calling function for its use.

```
main( )                    main( )
{                          {
   . . . . .                  . . . . .
function1(x,y,z)           function1(x, y, &s,&d);
   . . . . .                  . . . . .
}                          }
functionl(int a,           function1 (int a, int b, int
int b, int c)              *sum, int *dift)
{                          {
   . . . . .                  . . . . .
   . . . . .                  . . . . .
}                          }
```

Q.146 Write a program to accept the elements of the structure
(i) emp_no (ii) basic_pay
and display the same structure along with the DA, CCA and gross salary. DA and CCA are calculated as follows
DA = 51% of basic_pay
CCA = Rs.100/- consolidated **(8)**

Ans:
```
main( )
{
int i, n, cca;
float da, gross;
struct
{
int emp_no, basic_pay;
}emp[10];
printf("Enter the number of employees");
scanf("%d", &n);
for(i=0; i<n; i++)
{
scanf("%d %d", &emp[i].emp _no,  &emp[i].basic_pay);
}
printf(Empno      Basic pay                          DA   CCA
Gross\n");
for (i=0; i<n; i++)
{
da=emp[i].basic-pay * .51;
cca= 100;
gross=emp[i]. basic_pay+da+cca;
printf("\n%4d %5d %7.2f %3d %8.2f ", emp[i].emp_no,
emp[i].basic_pay, da, cca, gross);
}}
```

Q.147 Explain the following operators
(i) Arithmetic Operator (ii) Relational Operator
(iii) Logical Operator (iv) Increment & Decrement Operator
(v) Conditional Operator

(10)

Ans:
(i) Arithmetic operator – C provides all the basic arithmetic operators.
+ addition
– subtraction
* multiplication
/ division
% modulo division
ex. a+b, a*b, a/b ~

(ii) Relational operator – These operators are used to compare two variables or expressions. C provide the following relational operators:

<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal
=	to equal to
!=	not equal to

(iii) Logical operator

&&	AND
||	OR
!	NOT

The logical operator && and || are used to test more than one condition.
Ex. a>b && x = = 10

(iv) Increment and Decrement operator
+ + increment
– – decrement
The operator + + adds 1 to the operand, – – subtracts 1
They both are unary operators.

Ex.	m = 10	m = 10
	X = ++m	x = m++
	Then x = 10, m = 10	then x = 10, m = 11

(v) Conditional Operator - A ternary operator pair "?" is available in C to construct conditional expressions of the form
exp 1 ? exp2 : exp3
where exp1, exp2, exp3 are expressions.
expl is evaluated first. If it is nonzero(true), then the expression exp2 is evaluated and becomes the value of the expression. If exp 1 is false, exp3 is evaluated, and its value becomes the value of the expression.
Ex. a=l0, b=15 x=(a>b)? a:b;

Q.148 Write a program to subtract two matrices. **(6)**

Ans:/* PROGRAM TO FIND THE SUBTRACTION TWO MATRICES */

```
#include<stdio.h>
#include<conio.h>
void main( )
{
int a[3][3], b[3][3], i, j, c[3 ][3 ];
clrscr( );
for(i = ( ); i < 3; i++)
{
for(j = ( );j < 3;j++)
{
printf("enter a[%d][%d]", i, j);
scanf("%d", &a[i][j] );
printf("enter b[%d][%d]", i, j);
scanf("%d ", &b[i]fj]);
```

```
}
}
for(i = 0; i < 3; i++)
{
for(j = 0; j < 3; j++)
     {
     c[i][j] = a[i][j] - b[i][j];
     printf("%d", c[i][j]);
     getch( );
     }
}
}
```

Q.149 Write a C function to find the maximum number in a set of n numbers. **(6)**

Ans:
1. Establish an array a[1..n] of n elements where n>1.
2.Set temporary maximum max to first array element.
3. While less than n array elements have been considered do
(a) ifnext element greater than current maximum max then assign it to max.
4. Return maximum max for the array of n elements.

Q.150 Write a C program to find the sum of digits of a number using recursion. **(8)**

Ans:
```
#include<stdio.h>
main( )
{
int n,sum,add(int);
printf("Enter an integer");
scanf("%d", &n);
sum=add(n);
printf("\nn The sum of the digits of %d is %d",sum);
}
int add(int a)
{
int t;
if(a = = 0)
return(0);
else
t=a%10+add(a/10);
return(t);
}
```

Q.151 Determine the value of the following expression
(i) x=a-b/3+c*2-1 a=15, b=12, c=3
(ii) z=2*x/(3*y) x=8.8, y=3.5 **(2)**

Ans:
(i) x = 15 – 12 / 3 + 3 * 2 – 1 = 15 – 4 + 3 * 2 – 1 = 15 – 4 + 6 – 1

= 11 + 6 – 1
=17 – 1
=16

(ii) z =2*8.8/(3*3.5)
= 2*8.8/(10.5)
= 17.6/10.5
= 1.67619

Q.152 Explain briefly the salient features of the various generation of computers. **(8)**

Ans: Salient Features of various generations of computer are:
i) First generation computers used vacuum tubes or valves, which worked on the principle of thermionic emission. The vacuum tubes contained filaments which, on heating, emitted electrons. Disadvantages of the first generation computers are:

- Extremely slow compared to the speed of the present day computers.
- Huge in size and not portable.
- Needed heavy air conditioning.
- Costly
- The system of processing was batch processing.

ii) Second generation computers replaced vacuum tubes with transistors. The transistors led to the development of new electronic hardware, including the first minicomputers. Advantages of second generation computers over the first generation ones are:

- Small size
- Operation speed relatively high.
- Portable and robust.
- Less cost.

iii) Third generation computers used integrated circuits (ICs). The integrated circuits are very small in size, required nominal power. The advantages of this generation of computers was that the size decreased considerably and speed was very high. They were portable and handy and the cost was also very low.

iv) Fourth generation computers used more powerful leis. like Medium, large and very large scale ICs. The most important electronic device developed during this generation was the microprocessor, which led to the development of micro computers. This led to computers which are :

- Very powerful in mathematical calculations and data processing abilities.
- Very fast, operating in the range of nanoseconds
- Capable of Multitasking
- Cheapest among all generation of computers.

v) Fifth generation computers are thought of to be intelligent ones (using Artificial Intelligence) which is lacking in today's computers. They would probably:

- work in parallel and not serially.

- Do a greater number of tasks at a time.
- Be faster and more powerful than the fourth generation computers and have large memories.
- Will do knowledge-processing in addition to data-processing jobs.

Q.153 Distinguish between minicomputer, microcomputer and mainframe computer. **(8)**

Ans:

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

Q.154 Distinguish between the following:
(i) Cache and virtual memory
(ii) Static and dynamic RAM **(8)**

Ans:

(i) Cache and virtual memory.

Cache memory:- A CPU cache is a cache used by the central processing unit of a computer to reduce the average time to access memory. The cache is a smaller, faster memory which stores copies of the data from the most frequently used main memory locations. As long as most memory accesses are to cached memory locations, the average latency of memory accesses will be closer to the cache latency than to the latency of main memory.

Virtual memory is an addressing scheme implemented in hardware and software that allows non-contiguous memory to be addressed as if it were contiguous. All current implementations of virtual memory support that Programs can address data that does not currently reside in main memory. When this occurs, the hardware and operating system automatically load the data at the requested address from auxiliary storage into main memory. This occurs transparently to the user program. As a result, programs can reference a larger amount of (RAM) memory than actually exists in the computer.

(ii) Static and dynamic RAM.

In static type of RAM i.e. SRAM, once the data is stored, it will be retained as it is in the memory as the static charge. It is relatively costlier than Dynamic RAM.
In case of Dynamic RAM i.e. DRAM, data once stored will not be retained forever unless we provide an electrical pulse to 'make it remember' repeatedly after some interval of time and this type of remembering pulse is known as the refresh pulse.

Q.155 What is DVD-ROM? What are the differences between CD-ROM and DVD-ROM? **(8)**

Ans:
a) DVD (also known as "Digital Versatile Disc" or "Digital Video Disc") is a popular optical disc storage media format used for data storage. Primarily uses are for movies, software, and data backup purposes, DVDs are of the same form factor as compact discs (CDs), but allow for 8 times the data storage capacity (single-layer, single-sided).

i) Storage capacity: Going by the storage capacity, the DVD specifications have four disk configurations, ranging from 4.7 GB to 17 GB. It ranges from single-sided, single-layer disks that are much like traditional CD ROM disks to double-sided double-layered DVDs. Normally a DVD ROM with a minimum storage capacity can store up to the data of 7 CD ROMs. Normally the capacity of a CD ROM is only 650 MB. The capacity of DVD ROM starts from 4.4 GB, which is 7 times more than a normal CD ROM. However, this storage capacity of DVD ROM relates to single-layer storage. But with double-layer storage, the storage capacity of DVD ROM can go up to 8 GB, which is 12.5 times higher than a normal CD ROM. Furthermore, double-sided, DVD ROM drives can store 8.8 GB, 14 times greater than a CD ROM's capacity. Double-sided, dual-layer DVD-ROM drives store 15.9 GB, 25 times greater than a CD ROMs capacity. To put it differently a CD ROM can hold the entire encyclopedia, but a DVD ROM can hold the encyclopedia, the dictionary, the thesaurus, and the phone number/address of everyone in whole India with still a lot of space left! With the storage of a DVD ROM, you can see interactive multimedia programs that use hours of full-screen video to help you see movies, play games or even learn your courses.

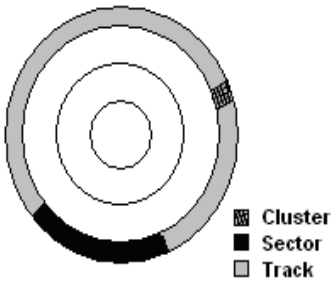
ii) Speed of data transfer: DVD ROM scores over the CD ROM. Going by figures the minimum transfer rate of CD ROM drives is 150 KB per second. However, the minimum transfer rate of DVD ROM drives is 11.08 MB per second. The data transfer rate of DVD ROM is approximately equivalent to a 92X CD ROM.

iii) Reliability: Individually though CD ROMs are very reliable and have a long shelf life yet if compared to the DVD, the DVD scores over the CDs in terms of reliability. Since DVD ROM discs are made of plastics bonded together, the discs are more rigid than CD ROM discs. Though the DVDs can also catch scratch just like CDs yet the scratches in the DVD normally do not matter as they are usually out of focus of the laser and therefore it does not affect the readability of the DVDs. In case of CDs, scratches could affect the data and could also make it useless if the scratches destroy critical tracks. Besides scratches, other factors like error correction also makes DVD much better and reliable in comparison to the CDs. Normally, in case of the DVD, the error correction is 10 times more effective than the CD error correction. Moreover the CDs normally follow RS-CIRC error correction mode, whereas DVD have RS-PC error correction. Likewise the error correction overhead in case of CD ROM is very high at 34 per cent, whereas it is only 13 per cent in case of DVD drive.

Q.156 What is a sector and track in hard disk? **(4)**

Ans:A Sector in the context of computing refers to a small area of a storage device, for example a hard disk drive.
For instance, a typical hard disk drive when low-level formatted is split into tracks, sectors and clusters:

- Tracks are concentric circles around the disk.
- Sectors are segments of a track.
- Clusters are a set of sectors.



Q.157 Briefly describe the function of serial access secondary storage device. **(4)**

Ans: Serial access storage device: These devices are used where data is to be accessed serially. Normally these devices are used for backup of large amount of data as it is cheap. Magnetic tape is example of this device.

Q.158 What are the different types of key switches used in keyboards? Explain their working with the help of neat diagram. **(8)**

Ans:Membrane Contact Keyswitches: The *membrane* contact keyswitch is a variant of the rubber dome design and works in a rather similar way. The basic mechanism is the same: contact pairs on a circuit board, and rubberized boots or "dimples" with a carbon button underneath. Press down the key and the rubber deforms, the carbon touches the contacts and a keystroke is sensed. The big difference here is that individual keycaps and plastic plungers are replaced with a thin membrane that fits over the rubber domes. There may not even be separate rubber domes, just molded "dimples" for each key, the carbon buttons in each dimple. The user presses directly on the membrane to collapse the domes and create contact with the printed circuit board. Travel is very small with this design, since there is no keycap and no plunger.

Capacitive Keyswitches: All of the other keyswitch technologies described in this section are *contact* designs. They all work using different ways of causing physical contact to establish a circuit and register a keypress. There is one keyboard technology, however, that detects keystrokes without using any form of contact at all. The *capacitive* keyswitch design makes use of a design characteristic of capacitors to determine when a key has been pressed. A capacitor is an electronic component that is comprised (at least conceptually) of a pair of parallel metal plates. When an electric field is applied to the plates, a charge is stored there.

Foam and Foil Contact Kevswitches: Like the mechanical contact keyswitch, the *foam and foil contact* keyswitch design also uses contact to complete a circuit and indicate when a keypress is made. However, it takes a very different approach to creating the contact. Each keyswitch is constructed of a (usually plastic) plunger on the top, connected to a foam pad. The foam pad is coated with foil on the bottom. A spring wraps around the plunger at the top, suspending the key in its normal position. Below all the keyswitches is a circuit board, printed with many pairs of copper contacts; one pair is oriented under the foam pad of each keyswitch. When they key is pressed, the foam pad moves down and touches the pair of copper contacts, completing the circuit and telling the keyboard that a key was pressed.

When the key is released, the spring pulls the plunger back up, breaking the contact.
Mechanical Contact Kevswitches: The simplest keyswitch technology in conceptual terms, *mechanical contact keyswitches* are "classical" switches, working in much the same way that many other types of switches do in the world around us. They work simply on the basis of two contacts mechanically touching each other to complete a circuit, not unlike a doorbell in some ways.

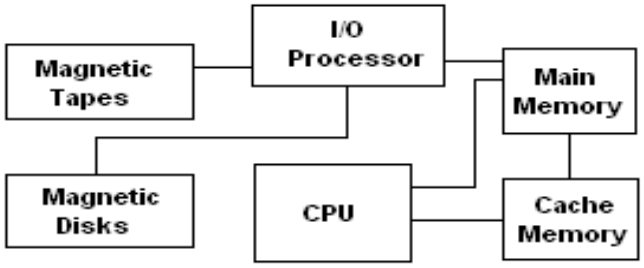
Q.159 Explain the term Hardware, Software and Firmware, specifying the relationship between them? Name at least two items of all categories. **(8)**

Ans:

Hardware	Software	Firmware
The physical components of a computer system, such as the computer itself, its modules and peripherals (the input/output devices and auxiliary storage units) are called hardware.	Software is basically the set of instructions grouped into programs that make the electronic devices in the computer to function in the desired ways. The different categories of software are system software (E.g are Operating system like DOS, UNIX, XENIX etc) and application software (E.g. are Word processors, electronic spreadsheets etc.). Software drives the hardware.	

Q.160 Briefly describe memory hierarchy and explain logic for making memory address map. **(8)**

Ans: Memory hierarchy:- The memory hierarchy system consists of all storage devices employed in a computer system from the slow but high-capacity auxiliary memory to a relatively faster main memory, and to an even smaller and faster cache memory accessible to the high-speed processing logic.
Example diagram of memory hierarchy in a computer system:



Memory address map:- The designer of a computer system must calculate the amount of memory required for the particular application and assign it to either RAM or ROM. The interconnection between memory and processor is then established from knowledge of size of memory required, and the type of RAM and ROM chips available.
The addressing of memory can be established by means of a table that specifies the memory address assigned to each chip. This table is called Memory Address Map, which is a pictorial representation of assigned address space for each chip in the system.

Q.161 What is a cache? How it is mapped with main memory? **(8)**

Ans: Cache memory: - The cache memory is employed in computer systems to compensate for the speed differential between main memory access time and processor 10gic.CPU logic is usually faster than main memory access time, with the result that processing speed is limited primarily by the speed of main memory.
Mapping of memory:- The basic characters tics of cache memory is its fast access time. Therefore, very little or no time must be wasted when searching for words in the cache. The transformation of data from main memory to cache memory is referred as a mapping process. Three types of mapping procedures are of practical interest when considering the organization of cache memory: Associative mapping, Direct mapping and Set-associative mapping.

Q.162 What is arithmetic and logic unit? Explain various functions performed by this? **(8)**

Ans: The arithmetic and logic unit is the core of any processor, it is the unit that performs simple arithmetic-logic calculations and shift operations. A typical ALU will have two input ports (A and B) and a result port (Y). It will also have a control input telling it which operation (add, subtract, and, or, etc) to perform and additional outputs for condition codes like carry, overflow, negative, zero result.
Operations: This is used for most of logical processing, for example, for calculations or comparisons. The arithmetic operations like + , – , * , and / are performed here. The logical operations like < , >, = , <= , >= and < > are also performed here.
In most of the arithmetical operations the result is in numerical form while in the case of logical operations the result can be YES/NO or TRUE/FALSE.

Q.163 What are cursor control devices? Explain working of any one such Device. **(8)**

Ans: These are the devices that are used for control of the cursor. They allow the position of control on the computer and the functions of the software to be executed. Example: mouse, trackball, light pen etc.

Mouse:- It is a pointing device having a small box with a round ball on the bottom and three buttons on the top. The mouse is attached on the terminal by cables. It allows the user to manipulate the cursor on the screen. The mouse captures the cursor and the movements of mouse control the operations on the computer.

Mouse could be optical, offering quite and reliable operation, or mechanical which is cheaper but noisier. User can move the mouse, stop it at a point where the pointer is to be located and with the help of buttons, make selection of choices.

Q.164 Explain the difference between an impact and non-impact printer? Which one is advantageous over the other and why? **(8)**

Ans: Impact printers use variations of standards typewriter printing mechanism where a hammer strikes paper through inked ribbon.

Non-Impact printer uses chemical, heat or electrical signals etch or induce symbols on paper. Many of these require special coated or treated paper.
Advantages due to applications :- Impact printers have the ability to produce multiple copies of documents as these printers are relatively cheaper compared to other technologies. So, these are good options if it is to be used for high volume work because of great speeds and less cost.

Non-impact printers provide good quality print at speeds unapproachable by other technologies like in Laser printer. These printers cannot produce multiple copies at one running.

Q.165 Write various specific features of Windows operating system. **(8)**

Ans: It is a multi-tasking operating system with graphic user interface Main features are:
(i) Program Manager: When the start button is pressed, a program manager is available by the name 'program'. This program manager stores the menu for an opening installed program of the window system.
(ii) File manager: A file manager called window explorer appears in
(iii)Print Manager: This is used to set up a printer.
(iv) Task Switching: This allows switch over between open windows.
(v) Close, Minimize, Maximize Button: This is used to open/close the running program.
(vi) My computer: This shows the complete details of the content of the drive and directories.
(vii) Network Neighbourhood: It connects the network to available resources.
(viii) Recycle Bin: This is a temporary storage for deleted files.

Q.166 What do you understand of multiprogramming and time sharing of system? **(8)**

Ans: Multiprogramming
In this many jobs of different users are performed in the memory at a time. The processor executes a portion of one program and then a portion of another. This way it continues the execution of the program in a sequence, until it comes to the end of the program. There are three different states of multiprogramming:
1. Ready: the program is able to use the process or when it is assigned to it.
2. Blocked: the program is waiting for the input and output operations to complete and is able to utilize the processor till then.
3. Running: the program is under processing by CPU.
Timesharing
It is most desirable for an individual user to minimize turn around time. This system allows several users to use the system simultaneously. There are several terminals connected to the system, which operate simultaneously. The CPU allots a fixed time period to each user and serves them in turn. This time sharing system is called interactive processing system.

Q.167 What types of softwares are used for specific application? Explain any one example. **(8)**

Ans: Application softwares are used and developed for the specific area of application of the user. These are customer oriented softwares suitable for specific area of application.
Example: Business Application
Micro electronic technology is enabling offices to function more efficiently nowadays, by various concerns:
a. In business forecasting.
b. To keep records up-to-date.
c. To carry out automatic checks on the stock of a particular item.
d. To prepare pay bills and personnel records.
e. In accounting, invoicing and billing.
f. In banking operations and data storage.
g. In business correspondence and communications.
h. In functions of various types in Life Insurance Business.
i. As an aid to management etc.

Q.168 What is the purpose of CONFIG.SYS and AUTOEXEC.BAT file. **(8)**

Ans: The file Config.sys contains special commands that configure hardware components and application programs. This helps to organize the storage devices and to use operating system to format label, copy and organise hard and floppy disc. It helps to organize the data stored in files. It provides an easy to use interface that let to communicate with the computer. Autoexec.bat is executed immediately after the Config.sys file and contains users own customized startup procedures to load programs automatically, display messages and specifies the software to be loaded. The path of the program is indicated by following example:
Drive:\directory\file
c:\tc\bin.

Q.169 What is mail merge? Write the steps to be followed in mail merge? **(8)**

Ans: In any working environment, there are times when a similar type of letter to be sent to many persons. In such case, general contents of each letter remain same, only name, address and other few information varies. So Mail Merger tool of word-processor is used to avoid same letter again and again.
It requires two different files which contain the following:
1. General body of letter
2. Data which varies from letter to letter
Steps followed in mail merge:
1. Press "N" at Opening menu to get non-document fi1e. Name the file with extension .DAT then key in various data of each client as records. Each field in the record should be separated by "," (comma)
Then save the non-document file. .
2. Press "D" at Opening Menu to open a document fi1e. Name the file and then type the letter.
.DF - This command informs wordprocessor about the name of data file to be used for mail merge.
.RV - This dot command is used in Mail Merge to specify the names of the variables which are used in document file. They are defined in order in which fields are stored ego NAME, ADDRESS1, ADDRESS2
All these places where these variables are used, the names would be preceded and succeeded by a "&" . This will communicate to word processor that following word is not a text but variable.

Q.170 Write applications and advantages of Spread Sheet? **(8)**

Ans: Applications: Spread sheets perform calculations, recalculates the result if any data stored in them changes. It is helpful for creating financial reports, comparing reports etc. It has a strong feature for creating groups which allow to illustrate relationship between two or more sets of data and understands the trends of data changes more easily.
Example: budgets, annual report of firm, payroll, bills, mark sheets, banking, inventory etc.
Advantages:

- Several mathematical, trigonometric, financial and statistical functions are built in. All sorts of complicated calculations can be performed very easily using these functions facilitating rapid operation.
- The results are accurate.
- Worksheet can be very large and any part can be viewed or edited.
- Data can also be viewed in graphs.
- Part or complete worksheet (WS) can be printed.
- WS can be stored and when required can be retrieved and edited.
- Any existing WS can be merged with any existing/new WS.
- Reports can be printed

Q.171 Write down the steps of inserting a formula in an EXCEL spread sheet. Which are the mathematical operators, used in Excel's formula? **(8)**

Ans:A formula is a numeric expression containing mathematical operation that result in a single value. In this we can perform addition, subtraction, multiplication, division or exponentiation.
Move the cell pointer to destined cell and click on the function option (From toolbar OR Menu Bar). Choose the function accordingly.
<some text is missing>
Mathematical Operator:

1	^	Exponentiation
2	+	Addition
3	–	Subtraction
4	*	Multiplication
5	/	Division

Any data can be changed without changing sheet and effect can be seen. This feature is one of the most useful features of spread sheet.

Q.172 For the purpose of deleting the file and directory which commands are used in a DOS system? Explain by suitably taking one example. **(8)**

Ans: For deleting a files and directories, the following commands are used:

Command	:	del (delete) eg.del XX.c (xx.c file name)
Command	:	rd (remove directory) eg. rd cc (cc is directory and should be empty)
Command	:	del tree (delete tree) eg. del tree yy (yy is directory which may not necessarialy be empty ,this command deletes all files in directory yy) .

Q.173 Write a note on capabilities and limitations of a computer. **(4)**

Ans: Capabilities of a Computer:- Computers are capable of having great processing speeds, huge memory, accuracy and versatility. The can be put to use in various areas like:

- For calculations in engineering and scientific research
- In data processing jobs.
- Commercial and financial applications
- Office automation.
- In CAD CAM
- Robotics
- Weather forecasting

Limitations of a computer:

- They need to be first programmed with specific instructions
- They cannot decide how are to be programmed or provide their own input.
- They can't interpret the data they generate.
- They can't implement any decisions they suggest.
- They can keep track of scientific data, but they can't conceive or express the ideas for continued research.

Q.174 What is a Super Computer? Give an example? **(4)**

Ans:A super computer is generally characterized as being the fastest, most powerful and most expensive computer.
One of the most powerful supercomputer today, The Cray-2 is set up in a Cshape, and is small enough to fit in a space of a large business desk.
Supercomputers recognize the largest word lengths of 64 bits or more. They calculate at rates upto 1.2 billion instructions per second. They can take input
from over 10,000 individual workstations.
Supercomputers are widely used in scientific applications such as aerodynamic design and simulation, processing of geological data, processing, of data regarding genetic coding and collecting and processing weather data.

Q.175 Explain the term single address, two address, three address and four address instructions. **(8)**

Ans:Single address instruction:- This type of instruction uses accumulator (AC) register for all data manipulation.
Eg: LOAD A, ADD B, STORE T etc.
Two address instruction:- In this type of instruction two memory address are provided.
Eg. MOV R1, A, ADD RI, B
Three Address instruction:- Three address instruction uses three operands in the instruction.
Eg. ADD R1, A, B

MULX, R1, R2

Q.176 Write a small machine language program for a two address computer to add two numbers and output the sum. **(8)**

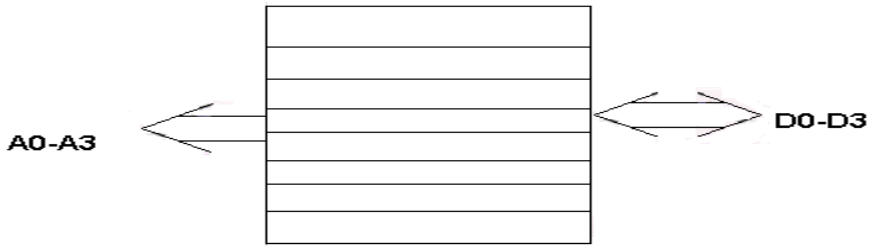
Ans: Let
MOV R1, A
ADD R1, B
MOV C, R1

Q.177 What is the size of MAR of a 16KB memory? Assume that the word size is 1 byte. **(4)**

Ans:MAR size is 16 bits.
As 16KB = 16 * 1024 = 2^{16}.

Q.178 Draw a block diagram of a memory which has 8 words of 4 bits each. **(8)**

Ans:



Q.179 How is data recorded on a CDROM? How is it read? **(8)**

Ans:CD-ROM (an abbreviation of "Compact Disc read-only memory") is a Compact Disc that contains data accessible by a computer. While the Compact Disc format was originally designed for music storage and playback, the format was later adapted to hold any form of binary data. CD-ROMs are popularly used to distribute computer software, including games and multimedia applications, though any data can be stored (up to the capacity limit of a disc). Some CDs hold both computer data and audio with the latter capable of being played on a CD player, whilst data (such as software or digital video) is only usable on a computer. These are called Enhanced CDs. Data is stored on the disc as a series of microscopic indentations ("pits", with the gaps between them referred to as "lands"). A laser is shone onto the reflective surface of the disc to read the pattern of pits and lands. Because the depth of the pits is approximately one-quarter to one-sixth of the wavelength of the laser light used to read the disc, the reflected beam's phase is shifted in relation to the incoming beam, causing destructive interference and reducing the reflected beam's intensity. This pattern of changing intensity of the reflected beam is converted into binary data.

Q.180 If a hard disk rotates at 3600 rpm and surface recording density on it is 1 Mbpi, what is the data transfer rate of the disk? **(4)**

Ans: Data Transfer Rate = (Spindle Speed /60 * recording density)
= 3600/60 * 1 = 60 MBPS

Q.181 Define the terms "Seek time" and "Latency time" of a magnetic disk. **(4)**

Ans: Seek time - The seek time of a hard disk measures the amount of time required for the read/write heads to move between tracks over the surfaces of the platters. Seek time is one of the most commonly discussed metrics for hard disks, and it is one of the most important positioning performance specifications.
Latency Time - The hard disk platters are spinning around at high speed, and the spin speed is not synchronized to the process that moves the read/write heads to the correct cylinder on a random access on the hard disk. Therefore, at the time that the heads arrive at the correct cylinder, the actual sector that is needed may be anywhere. After the actuator assembly has completed its seek to the correct track, the drive must wait for the correct sector to come around to where the read/write heads are located. This time is called latency. Latency is directly related to the spindle speed of the drive.

Q.182 Explain the working of voice recognition devices? What are its applications? **(8)**

Ans: Speech recognition (in many contexts also known as automatic speech recognition, computer speech recognition or erroneously as voice recognition) is the process of converting a speech signal to a sequence of words, by means of an algorithm implemented as a computer program.
Speech recognition applications that have emerged over the last few years include voice dialing (e.g., "Call home"), call routing (e.g., "I would like to make a collect call"), simple

data entry (e.g., entering a credit card number), preparation of structured documents (e.g., a radiology report), domestic applications control and content-based spoken audio search (e.g. find a podcast where particular words were spoken).
Voice recognition or speaker reorganization is a related process that attempts to identify the person speaking, as opposed to what is being said.
Speaker-dependent dictation systems requiring a short period of training can capture continuous speech with a large vocabulary at normal pace with a very high accuracy. Most commercial companies claim that recognition software can achieve between 98% to 99% accuracy (getting one to two words out of one hundred wrong) if operated under optimal conditions.
This explains why some users, especially those whose speech is heavily accented, might actually perceive the recognition rate to be much lower than the expected 98% to 99%. Speech recognition in video has become a popular search technology used by several video search companies.

Q.183 Explain the working of Thermal and Crystal based inkjet printers. **(8)**

Ans:Thermal Ink Jet

Most consumer ink jet printers work by having a print cartridge with a series of tiny electrically heated chambers constructed by photolithography. To produce an image, the printer runs a pulse of current through the heating elements. A steam explosion in the chamber forms a bubble, which propels a droplet of ink onto the paper (hence Canon's tradename for its inkjets, *Bubblejet).* The ink's surface tension as well as the condensing and thus contraction of the vapour-bubble, pulls another charge of ink into the chamber through a narrow channel attached to an ink reservoir.
The ink used is usually water-soluble pigment or dye-based but the print head is produced usually at less cost than other ink jet technologies.
This is not the same thing as a thermal printer, which produce images by heating thermal paper, as seen on some fax machines, cash register and ATM receipts, and lottery ticket printers.
Crystal based Ink Jet
All Epson printers and most commercial and industrial ink jet printers use a piezoelectric material in an ink-filled chamber behind each nozzle instead of a heating element. When a voltage is applied, the crystal changes shape or size, which generates a pressure pulse in the fluid forcing a droplet of ink from the nozzle. This is essentially the same mechanism as the thermal inkjet but generates the pressure pulse using a different physical principle. Piezoelectric ink jet allows a wider variety of inks than thermal or continuous ink jet but is more expensive.
Ink Jet with Piezoelectric is very fast and cost effective. When the Piezo crystal has an applied voltage, the crystal will shake the ink stream, causing it to break off in very small, fine droplets as the ink leaves the orifice plate hole. This droplet of ink can then be either charged or not charged, depending on whether the droplet of ink is to be printed or not.
If the droplet is to be printed onto the paper, the ink droplet is not charged. However, if the droplet is not required to be printed to the paper, it is charged with a positive bias, this way the ink droplet is then attracted to the negatively biased charge plate, the ink will hit the plate and will be vacuumed away by an ink recycle system. (This is used during the printer's

automatic head-cleaning procedure, albeit consuming usable ink in the process.)

Q.184 Differentiate between single-user and multiuser operating system? **(5)**

Ans:A Single user operating system is mainly DOS here one program runs at a time and it is a character user interface, whereas in multiuser more then one program runs at a time and it works in a time sharing mode. The CPU time is shared with different users in a round robin fashion.

Q.185 Briefly define Multiprocessing and Multiprogramming. **(5)**

Ans: Multiprocessing is a method of processing using more than one central processing unit where more than one set of instruction can be executed at a time.
Multiprogramming is the simultaneous handling of multiple independent programs by interleaving or overlapping their execution.

Q.186 Compare XCOPY, DISKCOPY and copy commands in DOS. **(6)**

Ans:XCOPY command in DOS helps in copying a directory structure. DISKCOPY command helps in making a copy of a Disk while copy command is used to copy a single or a group of files from one location to another.

Q.187 What is the advantage of mail merge. **(5)**

Ans: Mail merge is a utility provided by Ms-Word which helps in sending the same message or letter to many people. Fields from a data source are added to a main document containing the required text and can be sent to the printer or an email. Mail merge can also be used to make envelopes, mailing labels from a database.

Q.188 Write the steps for creation of bar charts of annual sale of a company saving five products. **(5)**

Ans: The steps for creating bar charts of annual sale of a company having five products are:-

- First select the data of worksheet from which you want to create chart.
- Choose the Chart menu from Insert menu
- A chart wizard dialog box appears on the screen.
- In Step 1 of 4 we have to select the chart type.
- In Step 2 of 4 we have to mention the data range.(sale of products)
- In Step 3 of 4 we have to select the chart options Title, legends, axis, data table, gridlines, data labels etc.
- In Step 4 of 4 we have to select the chart location i.e the same sheet or new sheet.

Q.189 Give the full form of
i) MSI
ii) EDSAC
iii) RAID **(3)**

Ans:
MSI → Medium Scale Integration
EDSAC → Electronic Delay Storage Automatic Calculator
RAID → Redundant Array of Inexpensive Disks

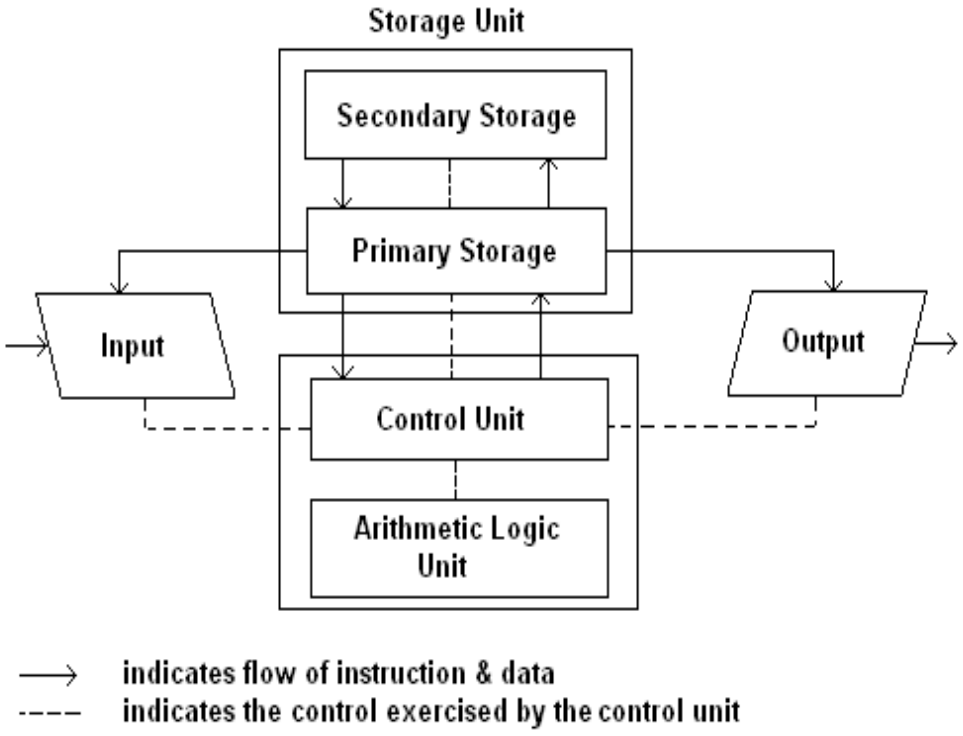
Q.190 Write a short note on super computers and their applications. **(5)**

Ans: Supercomputers are the most powerful and most expensive computers available at a given time. They are mainly used for processing complex scientific applications that require large processing power Super Computers are multi-processing and parallel processing technologies to solve complex problems faster, and hence they are parallel computers or parallel processing systems. Modern Super Computers employ hundreds of processors and are known as massively parallel processors.

Applications
(i) Super Computers are used to analyse large volumes of seismic data during oil-seeking explorations to identify the possible areas of getting petroleum products.
(ii) Super Computers are used to stimulate airflow around a aircraft at different speeds at altitudes for processing an effective aerodynamic design to develop aircrafts with super performance.

Q.191 Describe the basic functions performed by a computer system, with the help of a suitable diagram. **(8)**

Ans: The five functional units of a digital computer system are input unit, output unit, storage unit, arithmetic logic unit, control unit.
INPUT UNIT:- Input device is a device, which accepts (read) the list of instructions and data from the outside world, converts it into machine readable form and transmits it, to the memory unit of the computer.
Ex. Keyboard, Mouse, Scanners, OMR, OCR etc
OUTPUT UNIT :- Output unit is a device, which translates information regarding the result of data processing into a human acceptable form and is supplied to the outside world.
Ex. Monitor, Printer, Plotter etc.
STORAGE UNIT:- Storage unit holds the instructions and data to be processed, and the intermediate and final results of processing. It is of two types, Primary memory (main memory) and secondary memory (auxiliary memory). Primary memory is the built in memory, which holds the data and instructions between the processing steps. CPU directly addresses the main memory.
Ex. RAMs and Semiconductor memories.
Secondary memory stores data file, compilers, application programs etc. The CPU does not directly read from the secondary memory. The information is first transferred to the primary memory and then accessed by the CPU.
Ex Hard Disk, CDs etc.



ALU (Arithmetic Logic Unit) The actual execution of the instructions is carried out in this unit during the processing operations.
Control unit (CU):- It manages and co-ordinates the operations of all the other components of the computer system. Both ALU & CU together form the Central processing unit of the computer system.

Q.192 Find the data transfer rate (in bytes/sec) for a disk pack, with a disk pack capacity of 20 Mbytes, 19 storage surfaces and 700 tracks/surface. **(4)**

Ans: Given Disc capacity = 200 Megabytes = 200 x 1024 kbytes.
No. of cylinders = No. of tracks per surface = 700
No. of tracks/cylinders = No. of used surface of the disk pack
= 19 – 2 = 17
Data Storage/Track = (200x1024)/(700x17) = 17.21 kbytes

Q.193 Give the limitations and uses of a magnetic disk. **(3+3)**

Ans: The uses of magnetic disks

- Used in application based on random data processing
- Used as a shared online secondary storage devices ex. Winchested disks
- Used as backup device for offline storage of data for later retrieval when required. Ex. Floppy disks, Tape disks, Disk packs

Limitations of magnetic disks

- Different to maintain the security and information stored on magnetic disks which are used as online secondary storage devices.
- They must be stored in a dust-free environment
- Costly compared to magnetic tapes
- Disk crash or drive failure often results in loss of entire data stored on it. It is not easy to recover the lost data.

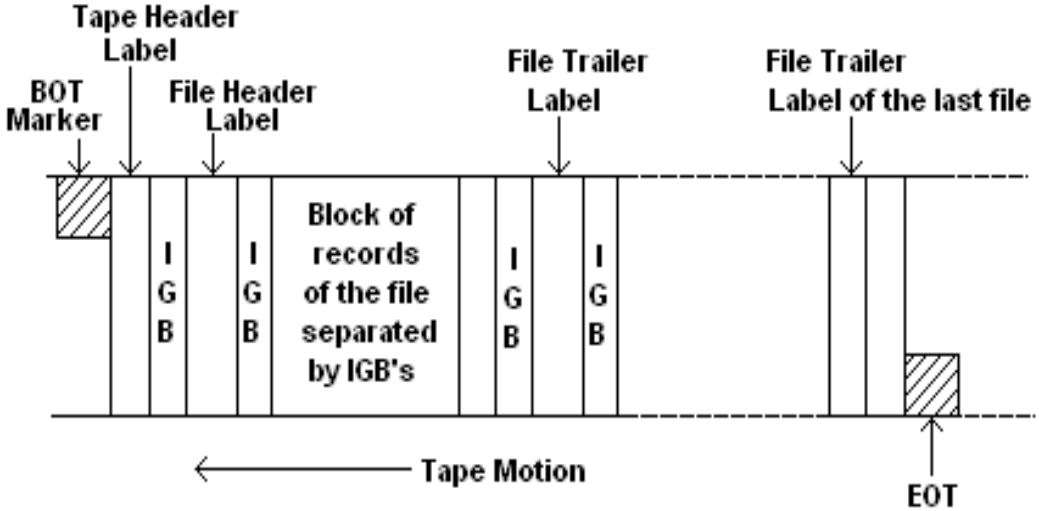
Q.194 Explain how information is recorded on a magnetic tape. **(6)**

Ans: The data on the tape is organized in the form of records separated by gaps. A gap between two consecutive records is the inter-block gap. A gap of records from a block, and a set of blocks constituted a file. A single tape can store many files. Many files are stored one after, the other sequentially.

To identify the files, the computer adds a file header label identifying the beginning of the file and file trailer label to signify the end of file. The file header label contains the label attribute, i.e. name of the file data when created etc.

A tape is wound on the spool & during I/O operation it moves from a supply spool to take-up spool. The first and last several feet of he tape are unused to allow the threading on take-up and supply spools. Two markers BOT and EOT are placed on the opposite edges of tapes for machine identification process. Beginning of tape (BOT) a reflective metal foil indicates the beginning of the tape to the equipment. End of tape (EOT), a reflective metal foil indicates the end of the unusable tape to the equipment.

Tape header label is placed between the BOT and the first file header file. It contains the tapes attributes such as the tape identifier, the number of files it contains, the data it was last used, and other control information that helps to prevent an important tape from accidentally being erased.



Data Organization on a Magnetic Tape

Q.195 Explain the working of an ink-jet printer. What are the two different types of ink-jet printers? **(4+2)**

Ans: Inkjet printers that form character and all kinds of images by spraying small drops of ink on the paper. The printer head of an inkjet printer contain upto 64 tiny nozzles, which can be

selectively heated up in a few microseconds by an integrated circuit register. When the register heats up, the ink near it vapourizes and is ejected through the nozzle and makes a dot on the paper placed in front of the head moves horizontally.
Inkjet printers can be both monochrome and colour. A monochrome inkjet printer uses a single print head, whereas a color inkjet printer uses multiple print heads, one per colour. Colour inkjet printers can produce multiple colour output.

Q.196 Explain the following terms with reference to a CRT monitor:
(i) Screen size
(ii) Resolution
(iii) Refresh rate **(3x2)**

Ans:
(i) SCREEN SIZE:- Display screens of monitor vary in size from 5 to 25 inches (diagonal dimension). Monitor with 17 to 21 inch screen sizes are common today with personal computers and other desktop computers. With users increasingly viewing multiple windows, video clips, animated objects, and complex graphics, monitors with larger screens are preferred. However, the larger the screen the costlier is the monitor,.

(ii) RESOLUTIONS:- Resolution refers to the number of point to which electron beam can be directly (pixels) on the screen. As the pixels are arranged in scan lines, the resolution of a monitor depends on its total number of scan lines and total number or pixels per scan line. The total no. of scan lines is called a monitor's VERTICAL RESOLUTION and the total number of pixels per san line is called its HORIZONAL RESOLUTION. The overall resolution of a monitor is expressed as a multiple of its horizontal and vertical resolutions. The higher the resolution of a monitor, the clearer will be its displayed images. Very high resolution of monitors, project extremely clear images that look almost like photographs.

(iii) REFRESH RATE:- The actual number of times that the electron beam scans the entire screen per second is called the REFRESH RATE of the monitor. The higher the refresh rate, the better will be the display quality of a monitor because it will be less strenuous for the eyes to continually view the screen. Today most monitors operates at 60Hz. That is, they refresh the screen 60 times per second. Better quality monitors operate at 70 to 90 Hz, refreshing the screen 70 to 90 times per second.

Q.197 Explain the function of light pen and joystick. **(4)**

Ans:LIGHT PEN Alight pen is an input device that uses a light sensitive stylus connected by a wire to a video terminal. The user brings the light pen to the desired point on the screen surface and presses a button, causing it to identify the location on the screen. Light pens are used to select the options from a menu displayed on screen or to draw images in graphics system by "dragging" the cursor around the screen. The pixels (dots) on a display screen are constantly being refreshed (re-illuminated) over again. When the user presses the light pen button, allowing the pen to sense light, the pixels being illuminated at the movement identifies on the screen.

JOYSTICK A joystick is an omni-directional level that is used to move the cursor on screen more rapidly then it can be moved with the directional arrow keys. It has a spherical ball at its lower end as well as its upper end. The lower spherical ball moves in a socket. Thus, a joystick can be moved right or left, forward or backward. The electronic circuitry inside the joystick detects and measures the displacement of the joystick from its central position. This information is sent to CPU. The pointer on the CRT screen moves according to the position of the joystick. The joystick is used extensively in video games, but is also used as an input device in computer aided design (CAD) systems.

Q.198 How does a bar code reader read the data? **(4)**

Ans:A bar code reader is a device used for reading (decoding) bar-coded data. It may be a hand-held scanner, or may be embedded in a stationary scanner. It scans a bar-code image and converts into and converts in into a alphanumeric value, which is then fed to the computer to which the bar-code reader is connected. Just as though the alphanumerical value had been typed on a keyboard.
A bar code reader uses a laser-beam scanning technology. The laser beam is stroked across the pattern of bars of a bar-code. Different bar codes having different patterns of bars reflects the beam in different ways, which is sensed by a light-sensitive detector. Reflected them transmitted to recognition logic circuits which convert in into an alphanumeric value.

Q.199 Explain how magnetic ink character reader is used in banks?
(4)

Ans: Magnetic-Ink Character Reader (MICR) is used by the banking industry for faster processing of the large volume of cheques being handled everyday by this industry. Banks that employ, MICR technology use a special type of cheque. The bank's identification code (name, branch etc), account number, and the cheque number are pre-printed (encoded) using characters from a special character set on all these cheques with a special ink that contains magnetizable particles of iron oxide before the cheques are given to the customers for use. When a customer presents a filled-in cheque at a bank, a bank employee manually enters (keys in) that amount written on the cheque in the lower right corner of the cheque using an MICR inscribers, which prints the amount with the magnetic ink. The data of the transaction is automatically recorded for all cheques processed that day. The cheque is then processed using an MICR reader sorter, which can recognize magnetic ink characters. The MICR reader-sorter reads the data on the cheques for distribution to other banks or for further processing. As the cheques enter the reading unit, they pass through a magnetic field, which causes the particles in the ink to become magnetized. Read heads then interpret these characters by examining their shapes. t\he sorter is basically used to sort the cheques into different pockets according to their identification code numbers.

Q.200 What is system software? Mention the most commonly used system software. Explain utility programs? **(2+2+4)**

Ans: System software is a set o one or more programs designed to control the operation and extend the processing capability of a computer system. Example: Operating System.

Some of the most commonly known types of system software are operating systems, programming languages translators, utility programs, performance monitoring software and communications software.
The various tasks performed by the utility programs are

- Hard disks or floppy disks are formatted.
- The files on a hard disk are recognized to conserve storage space.
- The backup of files stored in the hard disk is taken on to a tape or floppy disk.
- A particular file is located from a directory of hundreds of files.
- The amount of available storage space in the hard disk is found.
- The system is scanned for computer viruses.
- The stored records are sorted in a particular order based on some key fields.

Q.201 Write the steps in the execution of an instruction by the CPU. **(6)**

Ans:

- The address of next instruction is read from the program counter by the control unit. The instruction is read into the instruction register of the control unit.
- The operation part is sent to the decode and the address part is sent to the memory address register by the control unit.
- The instruction is interpreted and the control unit sends signal to the appropriate unit i.e., involved to carry out the task specified in the instruction. Ex. For any arithmetic or logic operation, the signal is sent to the CPU. The control unit ensures that the data corresponding to the address part is loaded in the suitable register in ALU, before the signal is sent to the ALU.
- When each instruction is executed, the address of the next instruction is loaded into the program counter and the above steps are repeated.

Q.202 Name the technique to reduce the speed mismatch between the slow I/O devices and the CPU. Explain how it is achieved? **(6)**

Ans: The speed mismatch between slow I/O device such as a printer and the CPU is normally of the order of few thousand. Thus, while a slow I/O device is inputting/outputting a few bytes to/from the main memory, the CPU can perform several thousands of operations. As a result, during reading/writing of data from/to data, from/to memory, the CPU is idle for a large percentage of I/O time. Spooling reduces the idle time of the CPU by placing all data that comes from the input device or goes to an output device on a magnetic disk. The primary reason for doing this to keep the program and data readily available to the fast and expensive CPU on a high speed I/O medium such as a disk. In most computer systems, special low-cost I/O processors are used for spooling the input data from a slow input device on to the disk or for outputting the spooled output data from the disk or for slow output device. These I/O processors function independent of the main processor (CPU). This enables the main high speed, expensive CPU to the fully devoted to main computing jobs. The process of spooling is transparent to the user programs. In general, spooling makes better use of both main memory and the CPU.

Q.203 Give the main features of Windows NT operating system. **(4)**

Ans: The main features of Windows NT operating system are:
(i) Unlike UNIX, its native interface is a GUI. The look and feel of Microsoft Windows NT's GUI is similar to that of Microsoft Windows GUI.
(ii) It supports multiprogramming and is also designed to take advantage of multiprocessing on systems having multiple processors.
(iii) It has in-built networking and communications feature so that any computer with Microsoft Windows NT can be (able) to work as a network client or server.
(iv) It provides strict system security.
(v) It has a rich set of tools for software development and system administration.
(vi) It can run Microsoft windows and many UNIX application directly.
(vii) It design is based on a microkernel so that it can be easily ported to many different types of machines, and its features can be easily enhanced by users.
(viii) It is a true 32-bit operating system in the sense that it can make full use of the 32-bit architectures of the processors an the memory and I/O bust to provide fast processing capability.

Q.204 Explain the difference between volatile and non volatile memory. Give an example of each type of memory. **(4)**

Ans: If the storage unit can retain the data store in it even when the power is turned off or interrupted, it is called NON-VOLATILE storage.
On the other hand, if the data stored are lost when the power is turned off or interrupted, it is called VOLATILE storage.
A non-volatile storage is desirable. The primary storage units are volatile and the secondary units are non-volatile.

Q.205 How does the bus width affect the overall speed of the computer system? Name and explain the three types of I/O buses. **(2+1+3)**

Ans: The bus width of a data bus is an important parameter that affects the overall speed of a computer system. This is because each wire of a bus can transfer one bit at a time. Hence, 8-bit bus (one having 8 parallel wires) can move 8 bits (one byte) at a time, a 16-bit bus can transfer two bytes and a 32-bit bus can transfer four bytes at a time. A wider data bus enables more bits of data to travel simultaneously resulting in faster exchange of data.

(i) DATA BUS: The data bus is used to transfer data between the CPU and I/O devices. A wider data bus will enable faster exchange of data between the CPU and I/O devices.
(ii) ADDRESS BUS: A computer system normally has multiple I/O devices like disk, tape, network etc simultaneously connected to it. Each I/O device has a unique identifier (or address) associated with it. The address bus is used to carry the address of the I/O device to be accessed by the CPU.

(ii) CONTROL BUS: The control bus is used to carry commands such as START, READ, WRITE, REWIND, TAPE, etc., from the CPU to I/O devices. It is also used to carry the status information of the I/O devices of the CPU.

Q.206 Explain the terms data cache and instruction cache. Name and explain the cache replacement policy. **(4+2)**

Ans: The instruction cache is used for storing program instructions and the Data cache is used for storing data. This allows faster identification of availability of accessed word in the cache memory and helps in further improving the processor speed.
REPLACEMENT POLICY: When a new block is to be fetched into the cache, another may have to be replaced to make room for the new block. The replacement policy decides which block to replace in such situation. It will be best to replace a block that is least likely to be needed again in the near future. Although it is impossible to identify such a block, a reasonably effective strategy is to replace the block that has been in the cache longest with no reference to it. This policy is referred to as the least recently used (LRU) algorithm. Hardware mechanisms are needed to identify the least recently used block.

Q.207 Differentiate between:
(i) CPU bound jobs and I/O bound jobs.
(ii) Tightly coupled and loosely coupled multiprocessing systems.
(iii) Internal fragmentation and External fragmentation.
(iv) Multiprocessing and Multiprogramming. **(4x4)**

Ans:
(i) **CPU bound jobs**
These jobs mostly perform numerical calculation with little I/O operations. They are so called because the heavily utilize the CPU during the course of their processing. Programs used for scientific and engineering computations usually fall in this category of jobs.
I/O bound jobs
These jobs normally input vast amount of data, perform very little computation, and output large amount of information. They are so called because during he course of their processing, the CPU utilization is very low and most of the time they perform I/O operations. Programs used for commercial data processing applications usually fall in this category of jobs.

(ii) **Tightly coupled multiprocessing systems**
In this system there is a single system wide primary memory that is shared by all the processors.
Loosely coupled multiprocessing systems
In this system, the processors do not share memory, and each process has its own local memory. In contrast to the tight coupled multiprocessing systems, the processors of loosely coupled systems can be located far from each other to cover a wider geographical area.

(iii) **Internal Fragmentation**
It is a situation when a process allocated more memory than its actual memory requirement and than its actual memory requirement and the additional memory allocated to the process remains

unutilized because it is neither used by the process to which it is allocated nor it can be allocated to any other process for use. Internal fragmentation may occur in the following situations:
(a) When the system uses fixed numbers of fixed-sized memory portions, any space in a partition that is in excess of the actual memory requirement of the process loaded into it is an internally fragmented memory space.
(b) When the system uses variable number of variable sized memory partitions, it may happen that there is a free block of size 200K and a new process arrives whose memory requirement is 199.5K. If we allocate exactly the requested size of memory, we will be left with a free-block of size 0.5K. Thus, the allocated memory may be slightly larger than the requested memory, and the additional memory that gets allocated to a process in this way is an internally fragmented memory space.

External Fragmentation

IT is a situation when enough total free memory space exists to satisfy the memory need of a process, but still the process cannot be loaded because the available free memory need of a process, but still the process cannot be loaded because the available free memory is continuous. For example, if the sizes of the free blocks FREE1, FREE2 and FREE3 and 400K, 100K and 100K respectively, then the total available free memory is 600K. Now if a new process arrives whose memory requirement is 500K, it cannot be loaded because 500K of contiguous free memory space is not available. Thus it is an example of external fragmentation. The amount of unusable memory area due to external fragmentation depends on the total size of the main memory and the average memory requirement of the process. However, statistical analysis indicates that as much as one-third of memory may be unusable due to external fragmentation.

(iv) **Multiprocessing**

Multiprocessing is the simultaneous execution of two or more processes by a computer system having more than one CPU. Multiprocessing makes it possible for the system to simultaneous work on several program segments of one or more programs.

Multiprogramming

Multiprogramming is the interleaved execution of two or more processes by a single CPU computer system. Multiprogramming involves executing a portion of one program, then a segment of another, etc., in brief consecutive time periods.

Q.208 Explain how the following features are carried out in the word processing Packages
(i) Entering mathematical equations.
(ii) Spell-checking.
(iii) Printing a document. **(6)**

Ans:

(i) On the insert menu, click object, and then click on the create New tab.
In the object type dialog box, click Microsoft Equation 3.
Click OK

(ii) Checking Spelling
On the tools menu, click options and then click the spelling and grammar tab.
Clear the check grammar with spelling check box

Click OK

(iii) Printing a document
On the File Menu, click print
In the print dialog box, enter the name of the printer paper range, no. of copies
Click OK

Q.209 Give the steps to insert a series of values in continuous locations like odd numbers 1,3,5,…

Ans:
- On the Edit Menu, click Fill and then click on series
- In the series dialog box, fill in the required data
- Click OK

Q.210 Give the two ways of renaming a work sheet in Microsoft Excel. **(4)**

Ans:
(i) - Right click on the sheet tab
- Select the Rename option

(ii) - Select the sheet option from the Format Menu
- From the sheet option select the Rename option

Q.211 Give the full form of the following abbreviations: **(3)**
i) RISC
ii) UNIVAC
iii) BASIC

Ans:
i) RISC : Reduced Instruction Set Computer.
ii) UNIVAC : Universal Automated Computer.
iii) BASIC : Beginners All purpose Symbolic Instruction Code

Q.212 Write about the hardware components used in the various generations of the computers. Give two examples of different computers used in different generations. **(5)**

Ans:

GENERATIONS	**HARDWARE COMPONENTS**	**EXAMPLES OF DIFFERENT COMPUTERS**
1ST Generation	Vacuum tubes, electromagnetic relay memories and punched cards for secondary storage were used.	ENIAC, EDVAC, ADSAC, UNIVAC I IBM 701
2nd Generation	Transistors, magnetic core	Honeywell 400, IBM

	memory, magnetic tapes and disks for secondary storage were used.	7030, CDC1604, UNIVAC LARC
3rd Generation	IC's with Large Scale Integration technologies, large magnetic core memory, more capacity magnetic tapes and disks for secondary storage were used.	IBM 360/370, PDP-8, PDP-11, CDC 6600
4th Generation	IC's with Very Large Scale Integration chips, microprocessors, semiconductor memories, large capacity hard disks as inbuilt storage and magnetic tape and floppy disks as portable storage media were used.	IBM OC and its clone, Apple II, TRS-80, VAX 9000, CRAY-1, CRAY-2, CRAY-X/MP
5th Generation	Ultra Large Scale Integration Chips Large capacity main memory, large capacity hard disks with RAID support, optical disks as portable read-only storage media were used.	IBM notebooks, Pentium PCs, SUN workstations, IBM SP/2, SGI Origin 2000, PARAM 10000

Q.213 A 6 platter hard disk has 600 tracks per surface. There are 10 sectors per track and 512 bytes per sector. What is the storage capacity of the disk? How many cylinders does the disk pack have? How many tracks are there per cylinder? **(4)**

Ans: Storage capacity of one surface = No. of tracks x No. of sectors x No. of per sector
No. of tracks = 600
No. of sectors =10
No. of bytes per sector = 512
No. of surfaces = 10 (upper and lower surfaces are not used)
Storage capacity of one surface = 600 x 10 x 512 = 3072000 bytes.
Storage capacity of disk pack = Storage capacity of one surface x No. of surfaces
= 3072000 x 10
= 30720000 bytes
= 29.29 Mb
= 30 MB
No. of cylinders = No. of tracks on each disk = 600
No. of tracks per cylinder = No. of usable surfaces on disk = 10

Q.214 Define the following terms with reference to a magnetic disk. **(4x1.5=6)**
i) Access time. ii) Seek Time.
iii) Latency Time. iv) Transfer rate.

Ans:i) Access Time: Access time is the interval between the instant a computer makes a request for transfer of data from a disk system to the primary storage and the instant this operation is completed. Access time depends on seek time, latency time and transfer rate.

ii) Seek Time: The time required to position the read/write head over the desired track is called the seek time. It depends upon the position of the access arms assembly when the read/write command is received.

iii) Latency Time: The time required to spin the desired sector under the read/write head is called the latency time. It is also known as the rotational delay time and depends on the distance of the desired sector from the initial position of the head on the specified track. It also depends on the rotational speed of the disk.
iv) Transfer Rate: Transfer rate is the rate at which data are read from or written to the disk. Transfer rate depends on the density of the stored data and rotational speed of the disk.

Q.215 Mention two advantages and two limitations of magnetic tapes for storage of digital information. **(6)**

Ans: The advantages of magnetic tapes for storage of digital information are

- Unlimited storage: The storage capacity of a magnetic tape is virtually unlimited because we can use as many tapes as required for recording our data.
- Ease of handling: Since tape reels are compact in size and lightweight, they are easily portable from one place to another and are much easier to handle and store.

The limitations of magnetic tapes for storage of digital information are:

- NO direct access: Magnetic tape is a sequential access media and hence data recorded on tape cannot be accessed directly. It is only retrieved serially that is if a data item is at the end of the tape, all the earlier part have to be read before accessing the required information. If access is required frequently, then magnetic tape is not a suitable storage media for such type of data. Too much machine time would be wasted in retrieving the data that is requested.
- Indirect interpretation: Data stored on a magnetic tape is in the form of tiny invisible magnetized and non-magnetized spots. Hence the contents of a tape cannot be interpreted and verified directly. Instead a print run must be made if the accuracy of tape data is questioned. This needs machines interpretation of the stored data.

Q.216 What is the voice reproduction system? How does it function? Give two applications. **(6)**

Ans:A Voice reproduction system produces an audio output from a set of pre recorded audio responses. The pre-recorded sounds are first converted into digital data and permanently stored in the computers memory. When the audio output has to be 'produced, the appropriate sound is selected from the pre-recorded sounds and the selected sound is converted back into analog form and is then sent to the speaker to produce the audio output.
Applications:

- Voice reproduction systems are used in automatic teller machines to provide step-by-step guidance to customers on how to use the ATM.
- Voice reproduction systems are used in talking alarm clocks and home appliances.

- Voice reproduction systems are used in automatic answering machines to give the vacancy status in a particular flight or train.

Q.217 Name the display technologies used by the flat panel monitors. Which is the most commonly used in flat panel monitors and why? **(4)**

Ans: Liquid-crystal display (LCD), electro-luminescent display (ELO) and gas-plasma display (GPO) are the display technologies used in flat panel monitors. Liquid-crystal display. (LCD) monitors are most commonly used. LCD monitors use a special kind of liquid crystals to display images on the screen which are normally transparent, but opaque when charged with electricity.

Q.218 Give one example of the following printers: **(4)**

i) Impact printers. ii) Non impact printers.
iii) Character printers. iv) Page printers.

Ans:
i) Impact printers: Dot matrix printers, chain printers, band printers and drum printers.
ii) Non-impact printers: Ink jet printers, laser printers.
iii) Character printers: Dot matrix printers. ink jet printers,
iv) Page printers: Laser printers.

Q.219 Give full form of the following: **(2)**

i) OCR
ii) MICR

Ans:
i) OCR: Optical Character Recognition.
ii) MICR: Magnetic-Ink Character Recognition.

Q.220 What is an image scanner? Explain the two types of image scanners. Give the limitations of image scanners. **(2+4+2)**

Ans: Image scanner is an input device, which translates paper documents into an electronic format that can be stored in a computer. The input documents can be typed text, pictures, graphics, or even hand written material. They are also known as optical scanners because optical technology is used for converting an image into electronic form.
The two types of image scanners are flat-bed scanners and hand-held scanners.
Flat-Bed Scanner: A flat-bed scanner consists of a box having a glass plate on its top and a lid that covers the glass plate. The document to be scanned is placed upside down on the glass plate. When the scanner is activated the light source which is situated below the glass plate moves horizontally from left to right. After scanning one line the light beam moves up a little and scans the next line. The process is repeated until the document is scanned completely.
Hand-Held Scanner: A hand-held scanner consists of a set of light emitting diodes encased in a small case that is convenient to hold in hand during operation. During scanning the

scanner is dragged slowly from one end of the document to the other with its light on. Dragging should be very steady over the document, otherwise the conversion into its bit map will not be correct. They are mostly used where high accuracy is not important and the volume of the scanned document is low. They are cheaper than flatbed scanners. .
The limitations of image scanners are:.

- Word processing cannot be done on the input documents, as the documents are stored as images instead of text.
- Storing the document as an image requires lot of storage space than compared to storing the document as a text.

Q.221 Explain with the help two examples, how software can be used as teaching and learning tool? **(4)**

Ans:

- Education softwares make the learning process interesting by incorporating audio-visual effects in various applications and by giving encouraging comments to the students.
- Education softwares help in conducting on line examinations, where the students can answer on line and know the results simultaneously. They also change the difficulty level of the questions based on the previous performance of the students.
- Education softwares help the individuals to learn foreign languages easily by incorporating both text and sound. The learner has the flexibility to hear the computer pronounce the word by selecting ttie word on the screen and also know the meaning of the phrase or the word.
- Education softwares act like a perfect tutor. They analyze the mindset of the user and guide them in answering the questions in the right manner. They help them tackle tough questions by giving them more exercises on the related areas.
- Reading material from encyclopedias and dictionaries is available on the CD- ROM. They help in searching for interesting facts also available in the Internet.

Q.222 What are the functions of a communication software? **(4)**

Ans: The functions of communication software are

- To ensure that connectivity is properly established between the source and the destination computer systems.
- The data is encoded in the right format at the source computer.
- Transfer of data from the source computer to the destination computer.
- The received data is decoded at the destination computer.

Q.223 Distinguish between single use and multi-user operating system with an example. **(4)**

Ans: 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.224 What are the functions of the following? **(3)**

(i) Program control Register.
(ii) Instruction Register.
(iii) Decoder.

Ans:i) Program Control Register: Program control Register holds the address of the next instruction to be executed by the CPU. Once the CPU has taken the current instruction, the program control register is automatically incremented to point to the next instruction.

ii) Instruction Register: Instruction register holds the current instruction that is being executed. The address part of the instruction is separated and sent to the memory address register and the operation part is send to the control section.

iii) Decoder: The decoder has the necessary control circuitry to decode and interpret the meaning of each and every instruction supported by the CPU.

Q.225 Explain the operation of a cache memory. **(3)**

Ans: Cache memory is a small, high speed buffer between the processor and the main memory. It temporarily stores the active data and instructions during processing. During processing, the CPU attempts to read from the cache memory and if found the data/instruction. IS- transferred to the CPU. If not found, a block of main memory containing the requested word is transferred to the cache memory and then to the CPU.

Q.226 Name and explain the different types of ROM's. **(6)**

Ans: There are two types of ROMs - manufacture-programmed and user-programmed.
PROM (Programmable Read-Only Memory): User - programmed ROM is also known as PROM as the user can program it. PROM chips cannot be reprogrammed.
EPROM (Erasable Programmable Read-Only Memory): The data on EPROM chips can be erased and reprogrammed to store new information. EPROM chips are of two types - UVEPROM and EEPROM.
UVEPROM (Ultra Violet Erasable Programmable Read-Only Memory): The stored information is erased by exposing the chip to ultraviolet light. .
EEPROM (Electrically Erasable Programmable Read-Only Memory): The stored information is erased by using high voltage electric pulses. It is also known as flash memory.

Q.227 In DOS, what do these commands do? **(4)**

i) COPY ii) CHKDSK iii) PROMPT iv) DISKCOMP

Ans:
i) COPY: Copy command is used to make a duplicate copy of the original file, It is an internal command.

ii) CHKDSK: CHKDSK command is used to check the specified drive for error. It is an external command.

iii) PROMPT: PROMPT command is used to set a new DOS prompt instead of the usual C> or A>.
iv) DISKCOMP: DISKCOMP command compares contents of the two disks, sector by sector. It locates errors and displays them. It is an external command.

Q.228 Explain about the different types of memory buses interconnecting the memory and the CPU. **(6)**

Ans: The three types of memory buses interconnecting the memory and the CPU are the data bus, address bus and the control bus.
i) Data bus: Data bus is used to transfer data between the CPU and the memory. It is a bi-directional bus. Bus width affects the overall speed of the computer. A wider data bus enables more bits of data transfer.

ii) Address bus: Address bus is used to carry the address of the memory location whenever data is to be transferred to or from memory. It is a unidirectional bus. The width of the address bus must be equal to the number of bits in the memory address register (MAR). The width of. address bus also determines the maximum number of memory locations a computer can have.

iii) Control bus: The CPU sends the control signals to the memory that specify whether the' data is to be read from or written to the specified location. It is a bi-directional bus.

Q.229 Give the major features of windows 98. **(6)**

Ans: The major features of windows 98 are:

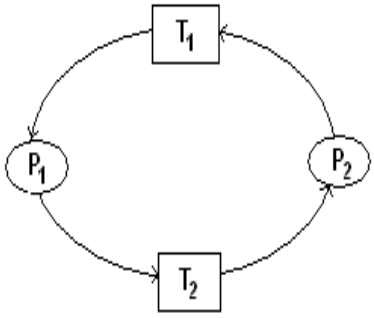
- Faster than Windows 95: The maintenance wizard of windows 98 checks the hard disk for problems and frees its space so that the programs run faster without any additional hardware.
- Web Integration: The interactive content of the Internet is combined with the power of computer and improved web features are provided.
- Improved Reliability: New utilities, wizards and resources are introduced which help the computer in running smoothly and efficiently. System file checker of Windows 98 restores the critical files if they are changed. Scan disk runs automatically if the computer is not shut down properly.
- Multiple Display: Several monitors can be used to run different programs on separate monitors.
- Power management: The computer can be started in few seconds i.e. the start up
- time is improved. The programs can be restored in their last saved positions.

- More Entertaining: New features like enhanced television, video playback and web TV for windows makes computer more entertaining.

Q.230 When are the two processes said to be in a deadlocked state. Explain with the help of an example. **(8)**

Ans: When competing processes prevent their mutual progress even though no single one requests more sources than are available, then the situation is called deadlock and the processes involved are said to be in a deadlock state. It may happen that some of the processes that entered the waiting state (because the requested' resources were not available at the time of request) will never again change state; because the resources they have requested were held up by other waiting processes.
Consider a system having two tape-drives T1 and T2 and the resource allocation strategy is such that a requested resource is immediately allocated to the requesting process if the resource is free. Consider two concurrent processes P1 and P2 requesting the tape drives in the following order:

1. P1 requests for one tape drive and the system allocates T1 to it.
2. P2 requests for one tape drive and the system allocates T2 to it.
3. P1 requests for one more tape drive and enters a waiting state because no tape drive is available.
4. P2 requests for one more tape drive and enters a waiting state because no tape drive is available.



Example of a deadlock situation involving processes P1 and P2 and resources T1 and T2
From now on, P1 and P2 will wait for each other indefinitely, since P1 will not release T1 until it gets T2 to carry out its designated task, that is, not until P2 has released T2, whereas P2 will not release T2 until it gets T1. Therefore, the two processes are in a state of deadlock. The requests made by the two processes are legal because each is requesting for only two tape drives, which is equal to the total number of tape drives available in the system. However, the deadlock problem occurs because the total request of both processes exceeds the total number of units for the tape drive and the resource allocation policy is such that it immediately allocates a resource on request if the source is free.

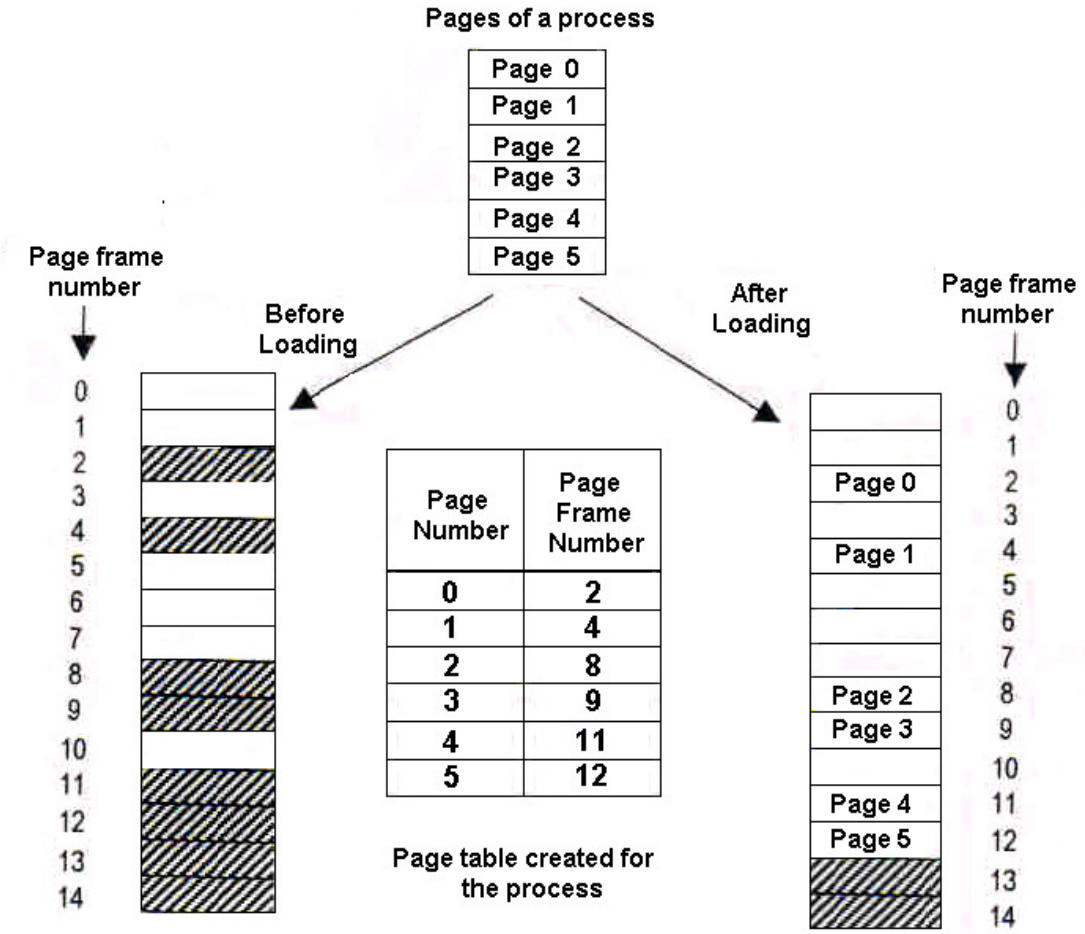
Q.231 What is paging? Explain paging mechanism with the help of an example. **(8)**

Ans: Paging is a memory management scheme, which allows the process's memory to be non-contiguous, thus allowing process to be allocated physical memory wherever free memory blocks are available.

In paging the physical memory of the computer system is partitioned into fixed-sized blocks called frames. The total memory requirement of a process including the data and instructions is partitioned into blocks of same size called pages. When a process is to be loaded into memory, the pages are loaded into the free page frames wherever the page frames are available. A mapping table called the page table is maintained by the operating system to keep track of which page of the process is loaded into which page frame of the memory.

Consider a process, which is partitioned into 6 pages, Page 0 to page 5. The physical memory is partitioned into 15 page frames, 0-14 whose size is the same as the size of the pages of the process. The paging mechanism is illustrated above.

During paging, every address generated by the CPU is divided into two parts, page, number and page offset. The corresponding page frame number from the page number is obtained from the page table. To define the physical memory address that is sent to the memory unit, the page offset is added to the base address of the page frame number.



Status of physical memory partitioned into page frames before loading the process (free page frames are indicated as hashed blocks)

Status of physical memory after loading the process (free page frames are indicated as hashed blocks)

Q.232 It is required to print the current date at the right-bottom and a heading at the top-center of the page. Give steps to achieve this. **(4)**

Ans: To print a date at the right-bottom, the steps are:

- On the insert menu, click Page Number option.
- In the Position box, specify the print the page numbers in the footer at the bottom of the page and alignment to the right.

To insert a header at the top-center the steps are:

- On the View menu, click Header and Footer.
- Insert the heading text and center-align the text.

Q.233 Explain the word wrap feature in MSWORD. **(4)**

Ans: The word-wrap feature automatically moves the last typed word to the next line if it goes past the right margin. That is if the last word cannot fit completely within the defined margins, the word is moved to the next line.

Q.234 A worksheet contains the following details. Give the steps to insert a bar chart of the Amount in all years. **(4)**

	A	B	C	D	E	F
Sales	1981	1982	1983	1984	1985	1986
Amount In crores	50	54	62	78	75	79

Ans: Enter the data into the worksheet.
Select the **Chart option** from the **Insert menu**
Select the **data range** on the worksheet to plot.
Select the **chart option** regarding the type and the placement.
Select the **location** where to place the chart.

Q.235 Explain the functions in a spreadsheet package. **(4)**
i) MIN ii) SUM

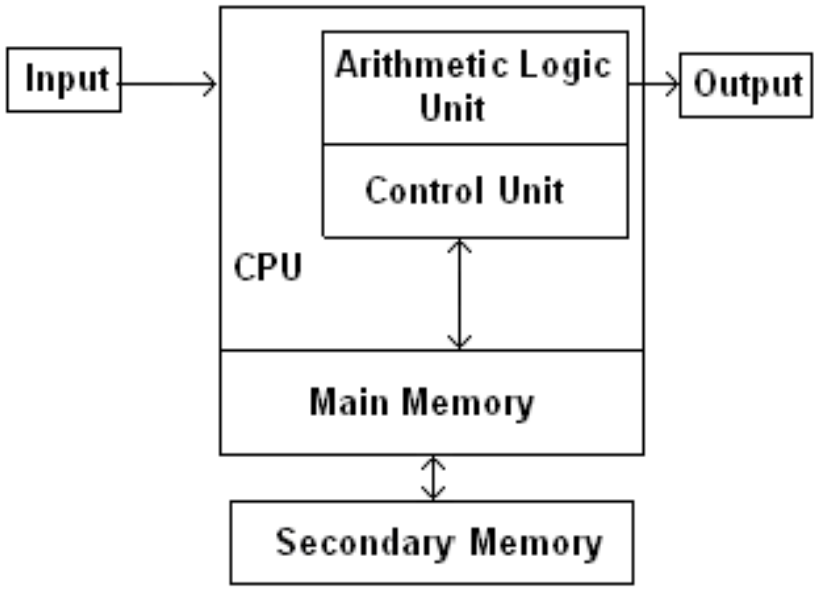
Ans:

i) MIN: This function returns the minimum value among the argument passed. It cannot take more than 30 arguments. Ex. MIN(B2..B9) – finds the minimum of the 8 number, B2 to B9 cells.

ii) SUM: This function calculates the sum of the numbers in a list. Ex. SUM(A1..A4, B1..B4) – sums the values in the ranges, A1..A4 and B1..B4.

Q.236 Draw a block diagram to illustrate the basic organisation of a computer system and explain the functions of the various units. (7)

Ans:



CPU: the main processing unit with abilities to perform arithmetic and logic operations + controlling various tasks and resources.
Main memory: temporarily storing data required by CPU to perform required operations at a given point of time, volatile.
Secondary memory: storing and achieving data, non-volatile
I/O: Managing input/output

Q.237 What do you understand by memory hierarchy? Name the general classes of storage media that might make up a memory hierarchy. **(7)**

Ans:
There is a trade-off to be made while designing the systems architecture in the context of computer memory, among the three key characteristics of memory, namely cost, capacity and access time. The plan is not to rely on a single memory component to technology, but to employ a memory hierarchy, as shown below. Going down the hierarchy, the following thing occurs:

- Decreasing cost per bit
- Increasing Capacity
- Increasing access time i.e. slower memory
- Decreasing frequency of access of the memory by the CPU

Following is a traditional memory hierarchy.

REGISTERS
CACHE
MAIN MEMORY
MAGNETIC DISKS
MAGNETIC TAPE

Q.238 Draw the truth table for the following Boolean expression $F(A,B,C) = A + B \cdot C + A \cdot B$ **(7)**

Ans:

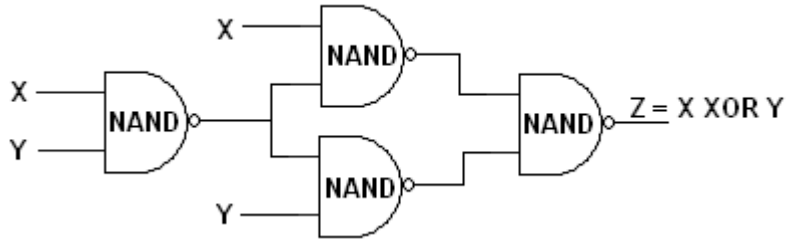
A	B	C	A'	B.C	A.B'	F(A,B,C)
0	0	0	1	0	0	1
0	0	1	1	0	0	1
0	1	0	1	0	0	1
0	1	1	1	1	0	1
1	0	0	0	0	1	1
1	0	1	0	0	1	1
1	1	0	0	0	0	0
1	1	1	0	1	0	1

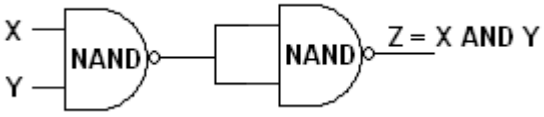
Q.239 What is the difference between a sequential circuit and a combinatorial circuit? What is meant by state of a circuit? **(7)**

Ans :
Logic circuits without feedback from output to the input, constructed from a functionally complete gate set, are combinatorial. Logic circuits that contain no memory (ability to store information) are combinatorial. Those, which contain memory, including flip-flops, are said to be sequential.
The logical structure/situation of a circuit at any given point of time is known as the state of the circuit.

Q.240 What are universal gates? Realize the following gates with the help of universal gates.
(i) XOR (ii) AND

Ans:
NAND and NOR are universal gates because all other gates can be constructed with these two gates.





Q.241 What are the advantages of EBCDIC code?

Ans :EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE (EBCDIC)
Using an 8-bit code, it is possible to represent 256 different characters or bit combinations. This provides a unique code for each decimal value 0 through 9 (for a total of 10), each uppercase and lowercase letter (for a total of 52), and for a variety of special characters. In addition to four numeric bits, four zone bit positions are used in 8-bit code. Each group of the eight bits makes up one alphabetic, numeric, or special character and is called a byte.
In EBCDIC, the bit pattern 1100 is the zone combination used for the alphabetic characters A through I, 11 0 1 is used for the characters J through R, and 111 0 is the zone combination used for characters S through Z. The bit pattern 1111 is the zone combination used when representing decimal digits. For example, the code 11000001 is equivalent to the letter A; the code 11110001 is equivalent to the decimal digit 1. Other zone combinations are used when forming special characters. Not all of the 256 combinations of 8-bit code have been assigned characters.
Since one numeric character can be represented and stored using only four bits (8-4-2-1), using an 8-bit code allows the representation of two numeric characters (decimal digits). Representing two numeric characters in one byte (eight bits) is referred to as packing or packed data. By packing data (numeric
characters only) in this way, it allows us to conserve the amount of storage space required, and at the same time, increases processing speed.

Q.242 Explain the rules of BCD addition.

Ans : The rules for BCD addition: First, the BCD digits are added as if they were two 4-bit binary numbers. When the binary sum is less than or equal to 1001 (decimal 9), the corresponding BCD digit sum is correct. However, when the binary sum is greater than 1001, we obtain an invalid BCD result. The addition of binary 0110 (decimal 6) to the binary sum converts it to the correct BCD representation and also produces an output carry as required.

Q.243 What do you understand by direct and indirect addressing? Explain with the help of examples. **(7)**

Ans : In direct addressing, the address field contains the effective address of the operand: EA = A. So it requires only one memory reference and no calculations. The address space is very limited.
In indirect addressing, the address field refer to the address of a word in the memory which in turn contains a full-length address of the operand.
EA = (A) ( ): contents of

Q.244 What are I/O devices necessary for a computer system. Why are I/O devices very slow compared to the speed of primary storage and CPU. **(7)**

Ans : I/O devices are the interfaces of the computer / CPU to the outside world, be it other systems or human users.

I/O devices are very slow compared to the primary memory or CPU because the data transfer rate of these peripherals is often much slower than that of the memory or CPU, so it is impractical to use the high-speed system bus to communicate directly to a peripheral.
SRAM is static RAM made of semi conductor devices i.e. traditional flip-flop logic-gate configurations, as there is no loss/gradual decay of charge, they do not need any refreshing circuits. DRAMs are dynamic RAMs made of capacitors, which have a tendency to gradual discharge, so they need a refreshing circuit for periodic charge refreshing to maintain data storage.

Q.245 What are the differences between SRAM and DRAM? Explain in brief why a refreshing circuit is needed for DRAM. **(5)**

Ans : SRAM is static RAM made of semi conductor devices i.e. traditional flip-flop logic-gate configurations, as there is no loss/gradual decay of charge, they do not need any refreshing circuits. DRAMs are dynamic RAMs made of capacitors, which have a tendency to gradual discharge, so they need a refreshing circuit for periodic charge refreshing to maintain data storage.

Q.246 Explain the principle of duality in Boolean Algebra. How is it useful? **(5)**

Ans : Duals are opposites or mirror images of original operators or constants. Switching logic using binary operators exhibits duality between the AND and OR operators. The dual of AND is OR and vice versa.
The principles of duality and complementing provide features that are used in the development of DeMorgan's theorems.

Q.247 Give the dual of the following Boolean expressions:
(i) A + B
(ii) A. B + A . B **(4)**

Ans : Use the principles as given above i.e. The dual of AND is OR and vice versa.

Q.248 Compare the characteristics. of impact and non-impact printers with examples. What are digitizers?

Ans : 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.249 What are the basic functions of an operating system?
Ans : Functions include:

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

Q.250 How does DOS manage the file system?

Ans : DOS manages the file system with FAT or File Allocation Table. It is a table that the operating system uses to locate files on a disk. Due to fragmentation, a file may be divided into many sections that are scattered around the disk. The FAT keeps track of all these pieces.
In DOS systems, FATs are stored just after the boot sector.

Q.251 What are serial and parallel devices? Explain with examples

Ans : Serial devices take data bit by bit, parallel devices can take a number of bits at a time through a parallel port.

Q.252 Explain how registers are organized in a processor. Explain the functions of the following registers
(i) Memory Address Register (MAR)
(ii) Memory Buffer Register (MBR)
(iii) Program Counter (PC)
(iv) Accumulator (A)
(v) Instruction Register (IR)
(vi) Input Output Register (I/O - R)

Ans :
(i) Memory Address Register (MAR) specifies the address in memory of the word to be written from or read into MBR.
(ii) Memory Buffer Register (MBR) contains a word to be stored in memory, or is used to receive a word from memory
(iii) Program Counter (PC) contains the address of the next instruction-pair to be fetched from memory
(iv) Accumulator (A) temporarily hold operands and results of ALU operations
(v) Instruction Register (IR) contains the op code instruction being executed
(vi) Input Output Register (I/O – R) temporarily hold I/O buffer data

Q.253 Explain the following terms:
(i) Virtual memory
(ii) Paging
(iii) Plotter

Ans :
(i) Virtual Memory: This is system memory that is simulated by the hard drive. When all the RAM is being used (for example if there are many programs open at the same time) the computer will swap data to the hard drive and back to give the impression that there is slightly more memory.

(ii) Paging: A method of managing virtual memory. When a requested page is not found in main memory, an interrupt occurs. The paging device machine then transfers the requested inactive page to memory. High rates of page swapping can degrade performance.
(iii) Plotter: A mechanical device which produces printout using vector or co-ordinate graphics often by using a pen moved about on rails.

Q.254 Explain the type of files the following extensions depict-.COM, .BAT, .DAT, .EXE, .SYS. In what situations does one need to edit the CONFIG.SYS file?

Ans:
.COM: command files e.g. command.com
.BAT : Batch files
.DAT : data files
.EXE : executable files
.SYS : system files e.g. config.sys
when a systems configuration details need to be changed e.g. the user wants a program to start off automatically when the system starts, or a software installation needs special configuration settings to run properly, config.sys
needs to be changed. .

Q.255 What is meant by booting? Explain the basic booting process.

Ans : During this process the computer will perform a self-diagnostic, reporting any errors it may encounter. Much can be learned about the state of the computer by paying close attention to the series of messages passing across the display during startup.
The basic booting process has the following sequence of operations:
1. Power-On Self Test (POST)
2. System Initialization, CMOS and BIOS Check
   After reading in the CMOS settings the boot sequence will continue with:
   PCI Initialization - If you have a PCI bus the system will now initialize the cards on the bus.
   Configuration Display - The BIOS now tests and displays the system configuration, including the status of:
   CPU
   Floppy drives
   Mouse and keyboard ports
   Serial ports
   Parallel ports
   Cache memory information: common sizes for cache memory are 64 KB, 128 KB, 256 KB, or 512 KB.

3. Loading the Disk Operating System (DOS)
4. Configuring the System with CONFIG.SYS
5. Running Programs at Startup with AUTOEXEC.BAT
6. Running Windows

Q.256 What is the difference between multi-tasking and multi-programming?

Ans : Multiprogramming is a method of running several different programs in a computer apparently at the same time.
Usually on a mainframe - the computer has a number of programs loaded into memory and the operating system switches quickly between them, processing a little bit of each one in turn. The high speed of the processor makes it seem like more than one program is being run at the same time.
Multi-tasking: Two or more programs actually run at the same time. Windows 95 and OS/2 support multitasking. Windows 3.1 supports "task switching" but not multi-tasking. Extended memory is divided up into "virtual machines" that share time on a single processor. With multi-tasking, a computer could be receiving communication via modem in the background while running Excel in the foreground.

Q.257 Explain the following terms with respect to a memory
(i) Access time
(ii) Storage capacity
(iii) Cost per bit of storage
(iv) Word length.

Ans :
(i) Access time: for random-access memory, this is the time it takes to perform a read or write operation, that is, the time from the instant that an address is presented to the memory to the instant that data have been stored or made available for use. For non-random access memory, it is the time it takes to position the read-write mechanism at the desired location.

(ii) Storage capacity: the total number of bits/bytes/words that a memory component can maximally hold, i.e. 16 MB RAM means the RAM chip can hold at the most 16 megabytes of data.

(iii) Cost per bit of storage: the cost of storing one bit of data in a particular type of memory component! technology, i.e. in pure semi-conductor based cache memory, the price of such memory component is very high, so the cost per bit of storage is also very high.

(iv) Word length: the 'natural' unit of memory organisation, can be 16/32/64 bits, length of the word is typically equal to the number of bits used to represent a number and to the instruction length.

Q.258 Explain how cache memory is used to increase the speed of processing of a computer.

Ans : Cache is the fastest memory component available providing a large memory size at the

price of less expensive types of semi conductor memories. It actually speeds up the processing of a computer by acting as a buffer reducing the delay in transfer of data in between the CPU and the slower and larger main memory.

Q.259 Write short notes on the following:
(i) Utility and availability of the chart facilities in Spread sheets.
(ii) Image editing software packages.
(iii) Features of groupware.
(iv) Salient features of Word Processing package.

Ans :(i) Most spreadsheet packages have chart drawing facilities with various analytical options for better visual representation and easier interpretation of spreadsheet data.
A chart consists of two axes. The horizontal (X) axis is called the category axis. The vertical (Y) axis is called the value axis. The chart contains one or more values. Each value is a list of number to be charted. A value entry contains two lists:

- the categories list, represented in the X axis.
- the series list, represented on the Y axis.

The following features are supported: Vertical Bar Chart

- Line Chart
- Pie Chart
- Area Chart
- Stacked Vertical Bar Chart Vertical Bar 3D Chart
- Customizable size, colors, title and legend
- Output HTML image map and tool tips

(ii) Image editing software packages:
There is a huge variety of software for manipulating images in various ways. Much of this software can be grouped under the heading image processing software. Another very important category is what we call image editing software. This group includes painting programs, graphic art packages and so on. They are often useful in conjunction with image processing software packages, insituations where direct immediate interaction with an image is the easiest way of achieving something. For instance, if a region of an image is to be masked out for subsequent image processing, it may be easiest to create the mask using an art package by directly drawing on top of the original image. Art packages also often allow the user to move sections of the images around and brighten or darken selected regions interactively. Few dedicated image processing packages offer the same flexibility and ease of use in this respect.

(iii) Features of groupware:
Groupwares are programs that help people work together collectively while located remotely from each other. Groupware services can include the sharing of calendars, collective writing, e-mail handling, shared database access, electronic meetings with each person able to see and display information to others, and other activities.
It was a term coined by marketers around 1995 to mean "software that facilitates group work," never emerged as a well-defined software category. Today the term is used less and tends to be narrowly identified with three products: Lotus Notes, Microsoft Exchange and

Novel GroupWise.

(iv) Parallel processing:
A computing method that can only be performed on systems containing two or more processors operating simultaneously. Parallel processing uses several processors, all working on different aspects of the same program at the same time, in order to share the computational load.

Q.260 Differentiate between interpreter and compiler. **(3)**

Ans: Interpreter: The language processor that translates (converts) each statement of source program into machine code and executes it immediately before to translate the next statement is called Interpreter. If there is an error in the statement the interpreter terminates its translating process at that statement and displays an error message. The GWBASIC is an example of interpreter.
Compiler: The language processor that translates the complete source program as a whole in machine code before execution is called compiler. The C and C++ compilers are best examples of compilers.
The program translated into machine code is called the object program. The source code is translated to object code successfully if it is free of errors. If there are any errors in the source code, the compiler specifies the errors at the end of compilation. The errors must be removed before the compiler can successfully compile the source code. The object program can be executed a number of times without translating it again.
The Interpreter differs from compiler that translates the entire source program into object program before execution.
The main advantage of Interpreter is that it makes easy to trace out and correct errors in the source program.

Q.261 Discuss how a laser printer works. **(4)**

Ans: Laser printer works as follows:
Charging the drum: When the printer receives a command from the computer's operating system to begin the print process, the photosensitive drum is negatively charged.
Exposing the drum: After the drum is charged, the laser flashes like a strobe light, the beam reflecting from the mirror to the negatively charged drum. As the drum spins, the laser creates the outline of the image on the drum by building a pattern of strongly charged and less-than-strongly-charged areas of negative voltage.
Developing the image: After the image is electro magnetically "written" to the drum, the toner must be applied to the print drum so the image can be transferred to paper. Adjacent to the print drum and the reservoir of toner is a smaller drum called the *developer.* The developer attracts toner powder to itself as it rotates between the toner and the print drum. When the print drum has toner on only the areas of slight negative charge, then the image is ready for transfer to paper.
Transferring the image: Now the registration rollers feed the paper into the printer and over the transfer corona wire, which gives the paper a strong positive charge. As the paper moves beneath the print drum, the weakly charged negative toner particles are strongly

attracted to the positively charged paper. So the image moves onto the paper. The paper continues to move through the assembly, passing over a static eliminator strip that removes all electromagnetic charge from the paper.
Fusing the image: As the paper leaves the print drum, all that holds the toner on the paper is a slight positive charge and a bit of gravity. The *juser* assembly finishes the printing by melting the toner into the paper as the sheet moves between a heated, Teflon coated roller and a rubber roller. The paper is then ejected from the print device.

Q 262 Write the trouble shooting steps if the computer is unable to load DOS. **(2)**

Ans.
1. If the computer is not coming to a MS-DOS prompt, reboot the computer and as the computer is booting, press the F5 key when you see the message "Starting MS-DOS" or the MS-DOS version. This will load the default standard MS-DOS.
2. If you successfully get to a MS-DOS prompt and would like to prevent the computer from loading the program that is preventing you from getting to a MS-DOS prompt, or if you would like to fix possible error messages you may be receiving when booting the computer, edit the autoexec.bat and/or the config.sys files.

Q.263 Differentiate between warm boot and cold boot. **(3)**

Ans: Cold Boot: It means starting from 'off state' or 'power off and than on by using the power button. You perform a cold boot every time you turn on the power switch of your computer. To "boot" the computer means to start it up and reset the memory and BIOS.
Warm Boot: It means restarting the computer using Alt+Ctrl+Del combination or restart command from the startup menu.

Q.264 Give the Principal working of Plotter

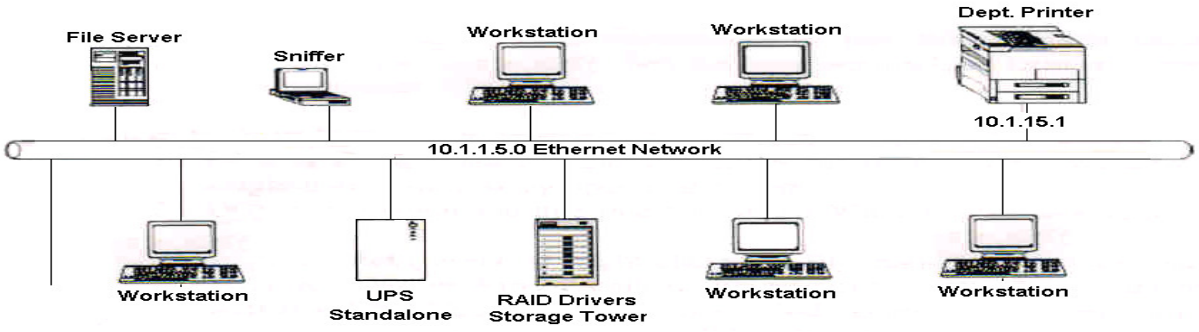
Ans: Plotter: A plotter is an output peripheral device used with a computer, which can be linkened to a printer. However instead of printing text or images, a plotter is more usually used to draw up technical plans and blueprints. Instead of the print cartridge found in a printer, a plotter commonly uses a pen.
With a plotter, a method of moving the paper and penholder are used to allow the pen to be in precisely the right point at the right time. However the order of print is different from a printer. The pen does not start at the top right moving to the left and then down a line. The plotter takes the whole image data, and then calculates a path for the pen. While the path for the pen may look complicated while the pen is in motion, In fact the plotter has calculated the shortest route for the pen, which involves the least amount of crossed lines as possible. Plotters are used to produce designs and graphs.

There are basically two types:
a) Drum Plotters: It is having a drum in which the paper is fixed and it is moved to and fro which makes it in vertical motions. The pens are clamped in the holder with different set of

colors and they are moved in horizontal directions. When the both move simultaneously the graphs or design is produced on the paper.
b) Flatbed Plotters: In this type of plotters the paper is not fixed and set of colored pens are in motion. The computer controls these.

Q.265 Write a short note on computer networking concepts. Show with the help of a diagram how will you connect five computers on a LAN. **(8)**

Ans: 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.
Computer Network can be classified on the basis of their scale as: Local Area (LAN), Metropolitan Area Network (MAN) and Wide Area Networks (WAN).
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:



Q.266 What are the characteristics of following types of operating systems?

(i) Time-sharing
(ii) Client-server
(iii) Multi-tasking
(iv) Multi-programming **(8)**

Ans: (i) **Time sharing:** In time sharing several systems (called as dumb servers having only computer peripherals) are attached to a single dedicated server having own CPU. Dumb servers share the CPU of dedicated server, as they don't have their own CPUs. Every action or command in Time-sharing operating systems is so short that very short span of CPU time is assigned for each user and thus the users at dumb systems have an impression that they have their own CPU though the fact is they share the CPU of dedicated server. Such short periods of time is called as time-slots or time-slices or time-quantum.

(ii) **Client-server**: The operating system has a multi-user session manager to enable multiple client-server sessions on the server and a multi-user stack protocol manager to manage one or more protocol stacks used in communicating with the clients. When a user connects to the server via a first client, the stack protocol manager assigns a first protocol stack to this first client-server connection and the session manager creates a first session for the user. When the user subsequently reconnects to the server using a second client that is different from the first client, the stack manager assigns a second protocol stack to a second client-server connection and the session begins creating a second session for the user. During this latter process, however, the session manager recognizes that the user is affiliated with the first session. The session manager adapts the first session to conform to the system configuration of the second client. The session manager then re-associates the second protocol stack with the reconfigured first session so that the user is returned to his/her original session, even though they logged on from a different client. .

(iii) **Multi-tasking**: Multitasking is a method of running several jobs at a time, now jobs can be either in the form of programs, processes, threads, user performing multi tasks at a time in single pc only. The main idea behind to do so is better CPU utilization. Several jobs are kept in the memory at a time such that when CPU is busy in execution of one job or task then as switches over to other job and make it ready to get its next turn for execution by CPU.

(iv) **Multi-programming**: When multitasking is just talking about executing multiple programs concurrently then the term multitasking term is referred as multiprogramming.

Q.267 Compare DOS and Windows as two operating systems. **(8)**

Ans: DOS Vs Windows:

- DOS is a single-user, single task operating system while Windows is a single-user, multitasking operating system.
- DOS has a command-line interface while Windows native interface is a GUI.
- DOS restricts users to eight-character file name with three-character extensions. While Windows allows file names to contain 255 characters as well as some punctuation marks like periods, commas, and semicolon.

- DOS is a simple operating system while Windows is a complete operating environment. All programs of Windows confirm to a standard way of working.
- DOS supports 2 GB of maximum partition size, while Windows supports 2 TB or more.
- DOS uses FAT 16 file system, while Windows uses FAT 32.
- Server administration is not possible in DOS.
- DOS does not support networking, Windows does.

Q.268 What are real-time systems? What are the differences between application software and system software? **(7)**

Ans: A real-time system (RT) is a system that satisfies the requirement of producing the desired results before a specified deadline. For a Real time system, the correct functioning of the system depends on the results produced and the time at which they are produced. A real-time operating system has well-defined, fixed time constraints. Processing must be done within the defined constraints, or the system will fail. A real time system is considered to function correctly only if it returns the correct result within any time constraints. So the following features are desirable in a real-time operating system:

- Multi-tasking within an operation
- Ability to define the priorities of tasks
- Priority driven or deadline oriented scheduling
- Programmer defined interrupts

There are two categories of Real time systems:
i) Soft RT system: A system whose operation is degraded if results are not preceded according to specified timing requirements.
ii) Hard RT system: A system whose operation is incorrect if results are not produced according to the timing constraints. Catastrophic results will happen then.
Application software and system software:
Application software are designed to perform specific data processing or computational tasks for the user. These programs are specifically designed to meet end-user requirements. For example: spreadsheets, word processors, media players and database applications etc.
While a **System software** is an essential part of computer operations. A System software is any computer software which manages and controls computer hardware so that application software can perform a task. Operating systems, such as Microsoft Windows, Mac OS X or Linux, are prominent examples of system software. The function of the systems software is to manage the resources of the computer, automate its operation and facilitate program development.

Q.269 Differentiate between an Optical Mouse and Mechanical Mouse. **(4)**

Ans: The mechanical mouse is the one that has a trackball on its lower side. The sensors inside this mouse are mechanical. On the contrary, in an optical mouse the movements of the mouse are detected through laser and there are no mechanical movements in it. It is also much quicker than the mechanical one.

Q.270 What is a digitizer? Explain the working of a digitizer. **(7)**

Ans: A Digitizer is a device, which converts analog information into a digital form. You can easily do your signatures using this pen or styles or pen on the digitizer and it is the job of digitizer or change your signatures in bit map and send them to computer for storage. From there, whenever needed the same shape can be regenerated. Digitizers are very accurate devices. They are available in many sizes. The most common sizes are 6 into 8 inch and 12 into 18 inch. The cost of digitizer increases with increase in size. Thus bigger size digitizers are very costly.

The styles used with digitizer looks like an ordinary pen. You can hold it in the same way and can make artistic strokes on the digitizer using it. Whatever strokes you will apply on the digitizer, the same can be seen on the screen.

Working of an Digitizer:

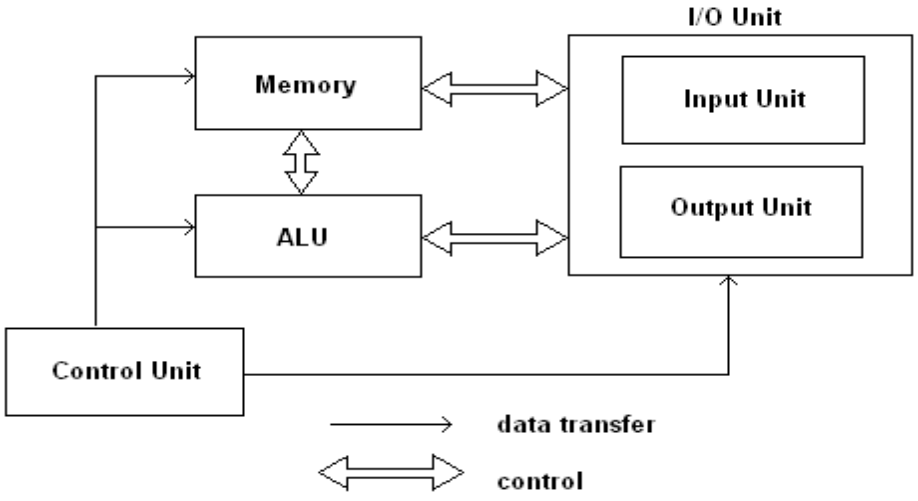
In case of a digitizer each position on the tablet relates to a specific position on the screen. So it traces the existing drawing more accurately and it can easily create original drawing such as architectural drawing with precise dimensions. The styles draws directly on the tablet and its movements are captured and translated into a corresponding drawing on the computer.

A puck or cursor can also be used instead of styles. In both the cases the exact positions of drawing device is detected by the tablet in terms of coordinates and is sent to the computer.

Q.271 With the help of a block diagram, explain the basic organization of a computer system. **(8)**

Ans: Basic Computer Organization: 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 and ROM.

Q.272 Give three advantages of windows over DOS operating environment. **(6)**

Ans: There are numerous advantages for windows users and programmers alike over the older DOS text based environment. They are:
1. Standardized graphical user interface
2. Multitasking capability
3. OOP approach in programming
4. Memory control
5. Hardware Independence
6. Use of dynamic link libraries (DLL)

Biggest Different is Graphical user interface and Multimedia feature both are support by Window, not MS-DOS. <Also see Ans7a, June 2004>

Q.273 What is an operating system? Explain how throughput, turnaround time and response time are used to measure the overall performance of a computer system. **(6)**

Ans: Operating system is an integrated set of programs that
a. Controls the resources of a computer system.
b. Provides its user an interface that is easier to use as compared to a bare machine.
There are several possible purposes of an operating system: To provide an environment for a computer user to execute programs on computer hardware in a convenient and efficient manner. To allocate the separate resources of the computer as needed to solve the problem given. The allocation process should be as fair and efficient as possible.
Throughput means number of processes completed / time unit.
Turnaround Time is the average time elapsed from when process is submitted to when it has completed.
Response Time - Average time elapsed from when process is submitted until useful output is obtained.

Typically, Utilization and Throughput are traded off for better Response Time. Response time is important for OS's that aim to be user-friendly. In general, we would like to optimize the average measure. In some cases, minimum and maximum values are optimized, e.g. it might be a good idea to minimize the maximum response time.

Q.274 What is an MICR device? Where is it primarily used? **(2)**

Ans:MICR refers to Magnetic Ink Character Recognition, and is a special type of printing used at the bottom of cheques, to be read by computers. It contains the cheque number, bank routing code, and account number.

Q.275 What are the main functions provided by most Operating Systems? **(6)**

Ans: Main functions of an operating system are:
1. Process management: A process is a program in execution. It is the job which is currently being executed by the processor. During its execution a process would require certain system resources such as processor, time, main memory, files etc. OS supports multiple processes simultaneously. The process management module of the as takes care of the creation and termination of the processes, assigning resources to the processes, scheduling processor time to different processes and communication among processes.
2. Memory management: It takes care of the allocation and deallocation of the main memory to the various processes. It allocates main and secondary memory to the system/user program and data. To execute a program, its binary image must be loaded into the main memory.
OS decides: which part of memory are being currently used and by whom, which process to be allocated memory, Allocation and deallocation of memory space.
3. I/0 management: This module of the as co-ordinates and assigns different I/O devices namely terminals, printers, disk drives, tape drives etc. It controls all I/O devices, keeps track of I/O request, issues command to these devices.
4. File management: Data is stored in a computer system as files. The file management module of the as would manage files held on various storage devices and transfer of files from one device to another. This module takes care of creation, organization, storage, naming, sharing, backup and protection of different files.
5.Scheduling: The as also establishes and enforces process priority. That is, it determines and maintains the order in which the jobs are to be executed by the computer system. This is so because the most important job must be executed first followed by less important jobs.
6.Security management: This module of the OS ensures data security and integrity. That is, it protects data and program from destruction and unauthorized access. It keeps different programs and data which are executing concurrently in the memory in such a manner that they do not interfere with each other.
7. Processor management: as assigns processor to the different task that must be performed by the computer system. If the computer has more than one processor idle, one of the process waiting to be executed is assigned to the idle processor.