Important Instructions to examiners:
1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance. (Not applicable for subject English and Communication Skills).
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

Q 1. Attempt any TEN of the following: **20 M**

a) Enlist any four keyword used in C.
(Any four keywords ½ mark each)
Ans:

auto	const	double	float	int	short	struct	unsigned
break	continue	else	for	long	signed	switch	void
case	default	enum	goto	register	sizeof	typedef	volatile
char	do	extern	if	return	static	union	while

b) What is operator precedence and associativity?
(Meaning of operator precedence – 1Mark , associativity – 1 Mark)
Ans:
Operator precedence means the sequence in which operations are done. It means operators of higher precedence are evaluated first.
Associativity gives their evaluation order. When an expression with more than one operators having equal priority, is executed, it works according to associativity.
OR

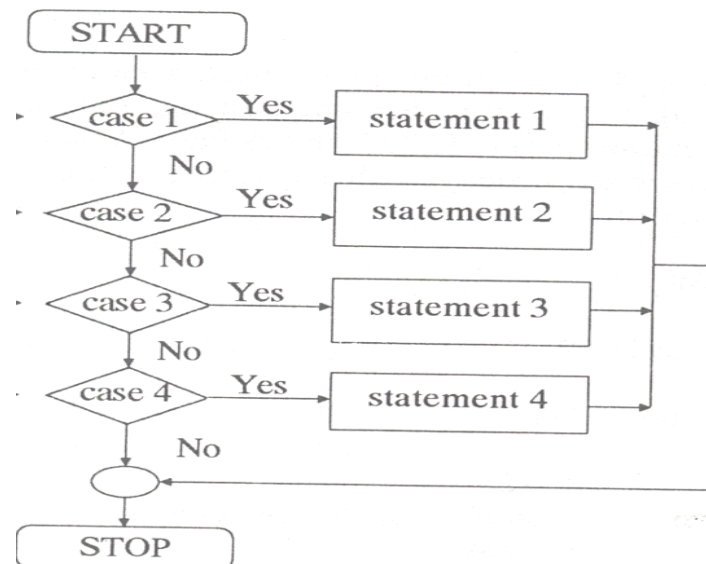| Category	Operator	Associativity

Postfix	() [] -> . ++ - -	Left to right
Unary	+ - ! ~ ++ - - (type)* & sizeof	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right

c) Draw the flow-chart of switch statement.
(Correct flow chart 2 Marks)
Ans:

d) Write syntax and use of i) strcmp( ) ii) strcpy( )
(Use of each function – ½ Mark, Syntax of each – ½ Mark)
Ans:
strcmp( ):- This function compares two strings identified by arguments and returns zero if both strings are equal, otherwise it returns the difference between ASCII values of first non matching character pair from the strings.
          Syntax: strcmp (string1, string2);

strcpy( ):- Function strcpy() copies the content of one string to the other string. It takes two arguments.
          Syntax: strcpy(destination string ,source string);
e) Write output of following program:
(Correct output – 2Marks)

Ans:

```
#include<stdio.h>
void main()
{
int i;
for(i=-1;i<=10;i++)
{
if(i<5)
continue;
else
break;
printf("Hello \n\n");
}
}
```

Output : No output (Nothing will be displayed on screen as break statement takes the control out of the loop).

f) List different types of functions.
Ans:
 (1 Mark each)
Types of functions:
   1) Built in functions
   2) User defined functions.
OR
(1/2 Mark each)
Types of functions
1) Function with no arguments and no return value.
2) Function with arguments and no return value.
3) Function with no arguments and return value.
4) Function with arguments and return value.

g) State the advantage of function.
(Any one correct advantage – 2Marks)
Ans:

The program may become too large and complex and as a result the task of debugging, testing and maintaining a program becomes difficult. So if program is divided into different functions,

then each function may be independently coded and later can be combined into a single unit. Functions are easy to understand, debug and test. Hence the task becomes easy.

h) What is the use of (-›) operator with pointers?
(Use of operator- 2 Marks)
Ans:
To access members of structure with pointer variable, we use arrow **-›** in between pointer variable and structure member.
       Syntax : ptr **-›** member of structure

i) Write the syntax of do-while loop.
(Correct syntax – 2Marks)
Ans:
do
{
Statements;
.
.
}
while(condition);

j) Define the term (i) identifier (ii) token
(Definition of identifier – 1Mark, token – 1Mark)
Ans:
(i) Identifier:
Identifier is a user-defined name and consists of a sequence of letters and digits. It refers to the names of variables, functions and arrays.
e. g. main, amount

(ii)Token:
In a program, the smallest individual unit is known as Token.
e.g. keyword, constants

k) State any two differences between while and do-while statement.
(Any two differences 1 Mark each)

Ans:

while statement	do while statement
while loop checks the condition first and then executes statements.	do while loop first executes statements and then checks the condition
If expression is true then only statements inside block are executed otherwise loop terminates.	If expression is false then also at least once statements inside block are executed.

Syntax:
while (expression)
{
 Statements
 }
	syntax:
	do
	{
	Statements
	} while (expression);

l) Find error in following program and justify it:
1. **#include<stdio.h>**
2. **void main()**
3. **{**
4. **int x;**
5. **int xx[3]={111,222,333,444};**
6. **for(i=0;j<3;i++)**
7. **printf("%f",a[i]);**
(Any two errors – 1 Mark each)
Ans:
Line 4 : it should be i instead of x because undeclared but used integer variable is i and not x.
Line 5 : one element from the array should be less as the array size is given as 3.
Line 6 : j should be replaced by I as I is an index variable and not j.
Line 7 : It should be %d instead of %f because it is an integer array as well as array name
          should be xx and not a.
Line 8 : a closing curly bracket is missing at the end to close main() function.

Corrected Program: (*not required, it is given for reference*)
1. #include<stdio.h>
2. void main()
3. {
4. int i;

5. int xx[3]={111,222,333};
6. for(i=0;i<3;i++)
7. printf("%d",xx[i]);
8. }

Q. 2 Attempt any FOUR of the following: **16 M**
a) Explain?: operator with suitable example.
(Explanation 2 Marks, Example 2Marks)
Ans:
Conditional Operator (Ternary Operator):
It takes the form "? :" to construct conditional expressions.
The operator "? :" works as follows:
Syntax:
exp1? exp2 : exp 3 ;
Where exp1, exp2 and exp3 are expressions.
exp1 is evaluated first, If it is true, then expression exp2 is evaluated.
If exp1 is false, exp3 is evaluated.
Example:
int a=10,b=5,x;
x=(a>b) ? a : b;
here x will take value 10 because condition given is if a>b.

b) Explain any four bit wise operators used in C with example.
(Any four bitwise operators with example– 1Mark each)
Ans:
Bitwise operators:
Bitwise OR – |
It takes 2 bit patterns, and performs OR operations on each pair of corresponding bits. The
following example will explain it.
                    1010
                    1100

        OR   1110
Bitwise AND – &
It takes 2 bit patterns, and perform AND operations with it.
                      1010
                      1100

```
                -------
    AND     1000
                -------
```

The Bitwise AND will take pair of bits from each position, and if only both the bit is 1, the result on that position will be 1. Bitwise AND is used to Turn-Off bits.

Bitwise NOT

One's complement operator (Bitwise NOT) is used to convert each "1-bit to 0-bit" and "0-bit to 1-bit", in the given binary pattern. It is a unary operator i.e. it takes only one operand.

```
1001
NOT   0110
-------
```

Bitwise XOR ^

Bitwise XOR ^, takes 2 bit patterns and perform XOR operation with it.

```
        0101
        0110
        ------
XOR     0011
        ------
```

Left shift Operator – <<

The left shift operator will shift the bits towards left for the given number of times.

int a=2<<1;

Right shift Operator – >>

The right shift operator will shift the bits towards right for the given number of times.

int a=8>>1;

c) Enlist any four types of arithmetic operators used in c and give one example of each.
(List of any four operators – 2 Marks , Example of each – ½ Mark)
Ans:

Operator	Example (considering a=10, b=2)
+ (plus)	Addition of two numbers. Eg : c=a+b; Ans : c= 12
(minus)	Subtractions of two numbers. Eg : c=a-b; Ans c= 8
* (product)	Multiplication of two numbers. Eg: c=a*b; Ans=20
/ (division)	Dividing one number by other.

	Eg: c= a/b; Ans=5
% (modulus)	Dividing one number by other and get remainder of the division. Eg : c=a%b; Ans = 0

d) State any four advantages of using flow-chart.
(Any four advantages – 1Mark each)
Ans:
1. Flowchart is the tool which helps to represent logic of the program pictorially because it is always more effective to visualize something graphically than to describe with words.
2. Flowcharts explain a process clearly through symbols.
3. It shows logical flow of the program accurately.
4. It is an important tool for planning and designing a new system.
5. It makes sure that no logical path is left incomplete without any action being taken.
6. It provides an overview of the system and also demonstrates the relationship between various steps.

e) Write a program to print following pattern:
```
1       2       3       4
5       6       7
8       9
10
```
(Correct logic – 2Marks, correct syntax – 2Marks)
Note: Any other relevant logic shall be considered.
Ans:

```
# include <stdio.h>
# include <conio.h>
main ()
{
int i, j,k=1;
clrscr();
for ( i=4; i>=1; i--)
{
for(j=1; j<=i; j++)
{
printf("%d ",k);
```

```
k++;
}
printf("\n");
}
}
```

f) State any four differences between if and switch statement.
(Any four differences – 1 Mark each)
Ans:

if statement	switch statement
1. 'if' statement checks any number of conditions and takes the action for true case and false case.	1. 'switch case' can be used when there are multiple options and only one is to be selected.
2. 'if' works even with the other comparison operators like >,<,>=,<=	2. 'Switch' statement operates only on equality of values.
3. 'if' does not have any provision to handle default case.	3. 'Switch' has a default case.
4. It does not require any statement like 'break' to come out of the block.	4.Every case needs a break statement to come out of switch block once a condition is matched

5. Syntax :
 if (condition)
 {
 …statement;
 .
 }
	5. Syntax:
	 switch(expression)
	 {
	   case value1:
	       statement;
	       break;
	   case value2:
	       statement;
	       break;
	       .
	       .
	 default:
	       statement;
	 }

Q.3 Attempt any FOUR of the following: **16M**

a) Write a program to display whether entered number is positive integer or not.
(Correct logic – 2Marks, correct syntax – 2Marks)
Note: Any other relevant logic shall be considered.
Ans:

```
 #include<stdio.h>
#include<conio.h>
void main()
{
int no;
clrscr();
printf("\n Enter a number");
scanf("%d",&no);
if(no>0)
printf("Number is positive");
else
printf("Number is not positive");
getch();
}
```

b) Write a program to take input as a number and reverse it by using while loop.
(Correct logic – 2Marks, correct syntax – 2Marks)
Note: Any other relevant logic shall be considered.
Ans:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int no,rem,rev=1;
clrscr();
printf("\n Enter number");
scanf("%d",&no);
while(no>=1)
{
rem=no%10;
printf("%d",rem);
no=no/10;
}
getch();
}
```

c. State any four control statements.

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)
SUMMER – 2016 EXAMINATION

Subject Code: 17212   <u>**Model Answer**</u>   Page No: 11 / 26

(List of any four control statements - 1Mark each)
Ans:
Control statements in c:-
1. If statement.
2. Switch statement
3. Break statement
4. Continue statement
5. Go to statement
6. For loop
7. While loop
8. Do-while loop

d. write a program to print Fibonacci series.
(Correct logic – 2Marks, correct syntax – 2Marks)
Note: Any other relevant logic shall be considered.
Ans:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c,i,no;
clrscr();
printf("\n Enter number of elements");
scanf("%d",&no);
a=0;
b=1;
printf("%d\t%d\t",a,b);
for(i=0;i<no-2;i++)
{
c=a+b;
printf("%d\t",c);
a=b;
b=c;
}
getch();
}
```

e. Explain else-if ladder with flow-chart.
(Explanation 2Marks, Flowchart 2Marks)
Ans:

else if ladder is used to take a multipath decision. It is used in a program when there are more than one conditions are involved.

Syntax:
if (condition 1)
statement -1;
else if (condition 2)
statement -2;
else if (condition 3)
statement -3;
else if(condition n)
statement -n;
else
default statement;
statement-x;

The above construct is known as else if ladder.
The conditions are evaluated from the top to the bottom.
As soon as the true condition is found, the statement associated with it is executed and the control is transferred to the statement-x.
When all the conditions become false, then the final else containing the default statement will be executed.

Flowchart:
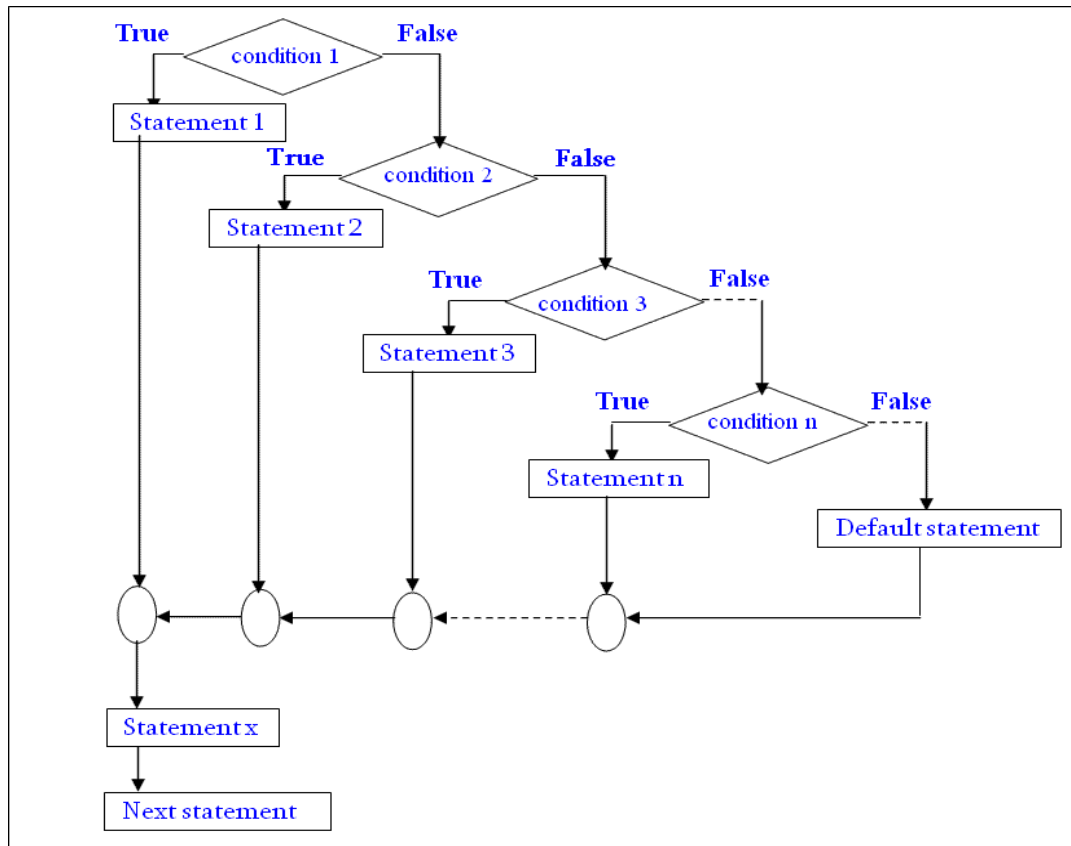
f. Write a program to display all even numbers from 1 to N.
(Correct logic – 2Marks, correct syntax – 2Marks)
Note: Any other relevant logic shall be considered.
Ans:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int N,i;
clrscr();
printf("\n Enter value for N");
scanf("%d",&N);
for(i=1;i<=N;i++)
{
if(i%2==0)
printf("%d\t",i);
}
getch();
```

}

Q4. Attempt any FOUR of the following   **16 M**
a) Explain the declaration and initialize of character array.
(Declaration of character array 2Marks, initialization of character array any one method 2 Marks)
Ans:
In C character array is used to represent string. String is a group of characters.
Declaration of character array:
Syntax:-
char character_array_name[size];
char is a data type, character_array_name represents string name, size determines the number of characters in the character array. When compiler assigns a character string to a character array, it automatically stores a NULL character ('\0) at the end of the string.
Example:-
char name[10];

Initialization of character array:
Compile time initialization:-
Character array can be initialized when it is declared.
Character array can be initialized with any of the following methods:
- char city[7]="Mumbai";
  city character array is 7 elements long as Mumbai contains 6 characters and 1 space for NULL character.
- char city[10]="Mumbai";
  city character array is declared as 10 characters long where Mumbai occupy 6 characters and remaining space is initialized with '\0' characters.
- char city[7]={'M','u','m','b','a','i','\0'};
  city character array is 7 elements long as Mumbai contains 6 characters and 1 space for NULL character.
- char city[ ]={'M','u','m','b','a','i','\0'};
  character array can be initialized without specifying size. In this case, the size of the array will be determined automatically, based on the number of elements initialized. In the example size of character array city is 7.

Run time initialization:-
An array can be initialized using scanf() function at run time.
         int a[5];
         for (i=0;i<5;i++)
         scanf("%d",&a[i]);

b) How the size of an array is calculated? Explain with suitable example.
(Description of size calculation 2Marks, any suitable example-2Marks)

Ans:
Size of an array is calculated with respect to two parameters:
1. Memory (number of bytes) required by the data type of the variable.
2. Number of elements declared as size in square bracket along with variable name.

Size of an array=memory required by data type * number of elements inside array variable.

Example:-
                int a[10];
Memory (number of bytes) required by the data type int- 2 bytes
Number of elements in array variable a -10
Size of a=2*10=20 bytes

c) Write a program for addition of two 3 x 3 matrix.
(Correct logic – 2Marks, correct syntax – 2Marks)

Note: Any other relevant logic shall be considered.

Ans:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[3][3],b[3][3],c[3][3],i,j;
clrscr();
printf("Enter first matrix elements:\n");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
scanf("%d",&a[i][j]);
}
}
printf("\nEnter second matrix elements:\n");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
scanf("%d",&b[i][j]);
}
}
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
c[i][j]=a[i][j]+b[i][j];
```

```
}
}
}
printf("\n\nAddition of two matrices is:");
for(i=0;i<3;i++)
{
printf("\n\n");
for(j=0;j<3;j++)
{
printf("%d\t",c[i][j]);
}
}
getch();
}
```

d) Explain recursion with example.
(Description of recursion 2Marks, any suitable example 2Marks)
Ans:

Recursion means a function calls itself. A recursive function contains a function call to itself inside its body. A recursive function is called from main ( ) function for the first time. After that every time function is called from its body.
Example:-

```
void main ( )
{
display ( );   // first call to display function
}
void display( )        ←
{                                  //Recursive call to display function
display ( );  _____|
}
```

e) Write a program to exchange value of two integer numbers using function.
(Correct logic – 2Marks, correct syntax – 2Marks)
Note: Any other relevant logic shall be considered.
Ans:

```
#include<stdio.h>
#include<conio.h>
void exch(int a,int b);
void main()
{
int a,b;
clrscr();
```

```
printf("Enter value of a and b");
scanf("%d%d",&a,&b);
printf("\n Value before exchange:");
printf("\n a=%d \t b=%d",a,b);
exch(a,b);
getch();
}
void exch(int a,int b)
{
int t;
t=a;
a=b;
b=t;
printf("\n Value after exchange:");
printf("\n a=%d \t b=%d",a,b);
}
```

f) Write a program to declare structure stationery having data member, name, quantity and cost. Accept and display this information for five items.
(Correct logic – 2Marks, correct syntax – 2Marks)
Note: Any other relevant logic shall be considered.
Ans:

```
#include<stdio.h>
#include<conio.h>
struct stationery
{
char name[20];
int quantity,cost;
}s[5];
void main()
{
int i;
clrscr();
printf("\n Enter information");
for(i=0;i<5;i++)
scanf("%s%d%d",&s[i].name,&s[i].quantity,&s[i].cost);
printf("\n Display information");
for(i=0;i<5;i++)
printf("%s%d%d",s[i].name,s[i].quantity,s[i].cost);
getch();
}
```

Q.5 Attempt any FOUR of the following **16 M**
a) Explain strlen() and strcat() string handling function from standard library.
(Explanation of strlen() 2 Marks, strcat() 2 Marks)
Ans:
strlen()-This library function counts the length of the string.
Syntax : strlen(string1);
Example :
i=strlen(st1);

strcat()-This library function concatenates a string onto the end of the other string.
Syntax : strcat(string1,string2);
Example: strcat(str1,str2);
Here, str2 will be concatenated at the end of the str1.

b) Write a program to sort array element in ascending order.
(Correct logic – 2Marks, correct syntax – 2Marks)
Note: Any other relevant logic shall be considered.
Ans:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int arr[10];
int i,j,temp;
clrscr();
for(i=0;i<10;i++)
{
scanf("%d",&arr[i]);
}
for(i=0;i<9;i++)
{
  for(j=i+1;j<10;j++)
  {
   if(arr[i]<arr[j])
   {
   temp=arr[i];
```

```
  arr[i]=arr[j];
  arr[j]=temp;
  }
}
}
printf("\nSorted array elements :\n");
for(i=0;i<10;i++)
printf("%d ",arr[i]) ;
getch();
}
```

c) Explain call by reference function with example
(Explanation 2Marks, Example 2 Marks)
Ans:

The **call by reference** method of passing parameter to a function copies the address of an parameter into the formal parameter.

Inside the function, the address is used to access the actual parameter used in the call. It means the changes made to the formal parameters affect the actual parameters.

Example :
```
#include <stdio.h>
void swap(int *a,int *b);
void main()
{
  int num1,num2;
num1=5;
num2=10;
swap(&num1,&num2);
  printf("Number1 = %d\n",num1);
  printf("Number2 = %d",num2);
  return 0;
}
void swap(int *a,int *b)
{
  int temp;
  temp=*a;
  *a=*b;
  *b=temp;
```

}

d) What is array of structure? List any two differences between array and array of structure.
(*Explanation of array of structure 2 Marks, Any 2 relevant differences 1 Mark each*)
Ans:
Array of Structure is collection of structure. Structure is used to store the information of one particular object but if want to store many objects then array of structure is used.
Example -
struct student
{
　　　int roll_no;
　　　char name[10];
}s[5];

Following are differences:
➢ Array stores similar data type elements whereas array of structure stores variables of structure where each structure variable contains different data type elements.
➢ Example of array: int a[10];
　Example Array of structure:
　struct book
　{
　char name[10] ;
　float price;
　};
　struct book b[100];

e) State four storage class. Explain any one.
(*List of four storage classes 2 Marks, Explanation of any one 2 Marks*)
Ans:
There are four storage classes in C:
　1. Automatic
　2. Static
　3. External
　4. Register

Features	Automatic	Register	Static Storage Class	External Storage

	Storage Class	**Storage Class**		**Class**
Keyword	auto	register	static	extern
Initial Value	Garbage	Garbage	Zero	Zero
Storage	Memory	CPU register	Memory	Memory
Scope	scope limited, local to block	scope limited, local to block	scope limited, local to block	Global
Lifetime	limited life of block, where defined	limited life of block, where defined	value of variable persist between different function calls	Global, till the program execution
location	Memory	Register memory	memory	memory
Example	void main() { auto int i; printf("%d",i); } OUTPUT 124	void main() { register int i; for(i=1; i<=5 ; i++); printf("%d ",i); } OUTPUT 1 2 3 4 5	void add(); void main() { add(); add(); } void add() { static int i=1; printf("\n%d",i); i=i+1; } OUTPUT 1 2	void main() { extern int i; printf("%d",i); int i=5 } OUTPUT 5

f) How structure members are access? Give example.
(Explanation 2 Marks, any example 2 Marks)

Note: Any method of accessing structure members shall be considered.
Ans:
Structure members are accessed with the structure variable and (.) operator.
In order to assign a value to a structure member, the member name must be linked with the **structure** variable using dot (.) operator also called **period** or **member access** operator.

Example – Consider book is a structure. Structure contains members as name and price. b1 is variable of a structure.
struct Book
{
 char name[15];
 int price;
} b1 ;

b1.price=200;   //b1 is variable of Book type and price is member of Book
In the above statement b1 is a structure variable and price is a member of that structure.
Price is accessed using (.) operator preceded with variable name.

Q.6 Attempt any FOUR of the following   16 M
 a) Explain global and local variable with example.
(*Explanation with example global variables 2 Marks, local variables 2 Marks*)
Ans:
Global Variables are declared before the main function. Global Variables can be accessed in any function in a program. Global Variables are alive till the end of the program.
Example of global variable.
//program to find the sum of two numbers
#include<stdio.h>
int a,b,result;   //declaration of global variables
void main()
{
....
}
Here a, b and result are global variables which are declared before the main function.

Local variables are declared inside a function. Local Variables cannot be accessed outside the function. Local Variables are alive only for a function.
Example of local variable

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)
SUMMER – 2016 EXAMINATION

Subject Code: 17212   **Model Answer**   Page No: 23 / 26

```
//program to add any two integers
#include<stdio.h>
void main()
{
int a,b,sum; // declaration of local variable
....
}
```
Here a, b and sum are local variables which are declared in main function.

b) Define pointer. How it declared and initialised?
(Definition 2Marks, Declaration 1Mark, Initialization 1Mark)
Ans:
A pointer is a variable which stores memory address of another variable which is of similar data type.

Declaration:
Syntax:
datatype *pointer_variable_name;
Example:
int *p;
Initialization:
Syntax:
pointer_variable_name = &variable_name;
Example:
int a;
int *p;
p=&a;

c) Write a program to show use of array of pointers.
(Correct use of array of pointer in a program 4 Marks)
Note : Any correct program shall be considered.
Ans:
```
# include<stdio.h>
void main()
{
int *a[4];
int i=31,j=5,k=19,l=71,m;
```

```
arr[0]=&i;
arr[1]=&j;
arr[1]=&k;
arr[1]=&l;
for (m=0; i<=3; m++)
 printf("%d", *(arr[m]);
}
```

d) State any two advantages and disadvantages of pointer.
(Any 2 correct advantages 1 Mark each, any 2 correct disadvantages 1Mark each)
Ans:
Advantages:
1. Pointers are more efficient in handling arrays and data tables.
2. They can be used to return multiple values from a function via function arguments.
3. Pointers permit references to functions and thereby facilitating passing of functions as arguments to other functions.
4. The use of pointer arrays to character strings results in saving of data storage space in memory.
5. Pointers allow C to support dynamic memory management.
6. Pointers reduce length and complexity of programs.
7. They increase the execution speed and thus reduce the program execution time.
Disadvantages:
1. If it contains an incorrect value it can lead to a problem when used.
2. When you use this incorrect pointer to read a memory location, you may be reading a incorrect garbage value then error may be occurred in the program.
3. Pointers are slower than normal variables.
4. If pointers are updated with incorrect values, it might lead to memory corruption.

e) State four arithmetic operations perform on pointer with example.
(List of any four arithmetic operations on pointer 1Mark each)
Note: any valid example of arithmetic operation shall be considered
Ans:
int * i;
i++;
In the above case, pointer will be of 2 bytes. And when we increment it, it will increment by 2 bytes because int is also of 2 bytes.

float * i;
i--;
In this case, size of pointer is still 2 bytes. But now, when we decrement it, it will decrement by 4 bytes because float is of 4 bytes.

int *a,*b,*c;
*a=10;
*b=20;
*c=*a * *b;
printf("%d" , *c);
 Here, Normal multiplication operation is done on pointer variables.
int *a,*b,*c;
*a=10;
*b=20;
*c=*a + *b;
printf("%d" , *c);

Here, Normal addition operation is performed on pointer variables.

f) Explain use of pointer in array.
(Explanation of use of pointer in array 4 Marks)
Ans:
We can declare a pointer of type int to point to the array arr.
int *p;
p = arr;
or
p = &arr[0]; //both the statements are equivalent.
Now we can access every element of array arr using p++ to move from one element to another.
We can use a pointer to point to an Array, and then we can use that pointer to access the array.
Example:
include<stdio.h>
void main()
{
int i;
int a[5] = {1, 2, 3, 4, 5};
int *p = a; // same as int*p = &a[0]
for (i=0; i<5; i++)

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)
SUMMER – 2016 EXAMINATION

Subject Code: 17212   **Model Answer**   **Page No: 26 / 26**

```
{
 printf("%d", *p);
 p++;
}
```

In the above program, the pointer *p will print all the values stored in the array one by one.

----------------------************------------------------