CHAPTER-11

USER DEFINED FUNCTIONS

TYPE A : VERY SHORT ANSWER QUESTIONS

1.	**A function's single most important role is to** **(a) give a name to a block of code** **(b) reduce program size** **(c) accept argument and provide a return value** **(d) help organize a program well**
Ans.	(d) help organize a program well
2.	**Define a function. What is the name of one-statement description of a function?**
Ans.	A function is a subprogram that acts on data and often returns a value. The name of one-statement description of a function is PROTOTYPE.
3.	**Function prototype is alternatively called. What is the statement specifically called that invokes a function?**
Ans.	A function prototype is alternatively called function declaration. The statement that invokes a function is specifically called function call.
4.	**What is a function declaration? How is a function declaration different from a function definition?**
Ans.	A function declaration tells the program about the type of the value returned by the function and the number and type of arguments. A function declaration has no body and no code. In other words, a (prototype) declaration introduces a function name to the program. On the other hand, a definition tells the Program, what is the function doing and how is it doing so.
5.	**What are actual and formal parameters of a function?**
Ans.	The parameter that appear in a function call statement i.e., which are passed are actual parameters. The parameter that appear in a function definition i.e., which receive the passed value are formal parameters.
6.	**Where is a function's return type specified? What is the return type of a function that does not return a value? How many values can be returned from a function?**
Ans.	A function's return type is specified first in the function prototype. The return type of a function that does not return a value is void. Only one value can be returned from a function.
7.	**What are global and local prototypes?**
Ans.	Global prototype: If the function's prototype appears outside all other functions in the program file, then it is called global prototype. Local prototype: If the function's prototype appears within another functions in the program file, then it is called local prototype.
8.	**When can a function prototype be omitted?**
Ans.	When the function definition appears before its calling function.
9.	**Construct function prototype for descriptions given below:** **(i) Rarb() take no argument and has no return value.** **(ii) mains() takes a float argument and returns an int.** **(iii) san() takes two double arguments and returns a double.** **(iv) sum() takes an int array and an it value and returns a long result.** **(v) check() takes a string argument and returns an int.**

Ans.	`(i) void Rarb();`
	`(ii) int mains(float);`
	`(iii) double san(double, double);`
	`(iv) long sum(int arr[], int);`
	`(v) int check(char []);`

10.	**What is the condition of using a function in an expression? When a function returns a value, the entire function call can be assigned to a variable. True or False?**
Ans.	Only the functions returning a value can be used in expressions. True.

11.	**Identify the errors in the function prototypes given below:**
	`(i) float average (a,b);`
	`(ii) float mult(int x, y);`
	`(iii) void calc(int a[], s=10);`

```
(iv) void arithop (int a[], int b[], int s=10, int j);
(v) float doer (int, int, float=3.14);
```

Ans.

Error	Correction
(i) datatype of the argument is missing.	float average(int a, int b);
(ii) datatype of the second argument is missing.	float mult(int x,int y);
(iii) datatype of the second argument is missing.	void calc(int a[],int s=10);
(iv) The argument s cannot have a default value unless argument on its right also has its default value.	void arithop(int a[], int b[], int s=10, int j=3);
(v) Third argument for which the default value has been provided is not named.	float doer(int, int, float T=3.14);

12. **When is a default argument value used inside a function?**

Ans. To keep the original copy of the argument value intact.

13. **What is the use of constant arguments?**

Ans. By using constant argument the function cannot modify the values as the values are constant.

14. **Given the function**

```
     int thrice(int x)
     {
          return a*3;
     }
```

Write a main() function that includes everything necessary to call this function.

Ans.

```
void main()
{     int t;
     t=thrice(4);
     cout<<t;
     getch();
}
```

15. **What is the principle reason for passing argument by value?**

Ans. The principle reason for passing argument by value is that the original copy of the argument value remains intact.

16. **What is the principle reason for passing argument by reference? In a function call, what all data items can be passed by reference?**

Ans. The principle reason for passing argument by reference is reflect the changes in original values. In a function call, the arguments which are pass by reference in function declaration are passed by reference.

17. **How are arrays passed in a function? What are the three ways of receiving an array in a function?**

Ans. In C++, an array can be passed as a pointer to an array by specifying the array's name without an index. Following are the three ways of receiving an array in a function:

a) Formal parameters as a pointer.
b) Formal parameters as a sized array.
c) Formal parameters as an unsized array.

18. **If amount is a float array holding 10 elements, then how are amount and amount + 6 different from one another?**

Ans. Question is not from this chapter

19. **Write the function prototypes for the function definition given below:**

```
     double f()
     {
          :
          return 1.0;
     }
```

Ans.

```
double f();
```

20. **Write the function declaration for the definition given below:**

```
int &f1()
{
     int a,b;
     :
     return b;
}
```

Ans.

```
int &f1();
```

21. When can a function appear on the left side of an assignment statement?

Ans. Only the function returning a reference can appear on the left-hand side of an assignment expression.

22. If the return type of a function is missing, what happens? What is the role of a return statement in a function?

Ans. If the return type of a function is missing, it is assumed to be returning int values. The return statement is used for immediate exit from the function or return a value to the calling code.

23. What are the three types of functions in C++?

Ans. The three types of functions in C++ are:
1. Computational functions.
2. Manipulative Functions.
3. Procedural Functions.

24. Write a declaration for a function called fun() that takes two arguments and returns a char. The first argument is int and is not to be modified. The second argument is float with a default value of 3.14159.

Ans.

```
char fun(const int a, float b = 3.14159);
```

25. What is meant by scope? What all kinds of scope is supported by C++?

Ans. The program part in which a particular piece of code or a data value can be accessed is known as its scope.
C++ provides four kinds of scope: local, function, file and class.

26. How is a global prototype different from a local prototype? How is a global variable different from a local variable?

Ans. A local prototype is placed in the body of another function and the function is locally available to the function that declares it whereas, a global prototype is placed outside all the functions and the function is globally available to all the functions.
A local variable is declared inside the function and is locally available within the function, whereas a global variable is declared outside all the functions and is globally available to all the functions.

27. Give the following code segment:

```
float a,b;
int main()
{     char ch;
      :
      {
            int i=0;
            :
      }
}
void f1(char gr)
{     short x,y;
      :
}
```

Write scopes for all the variables mentioned above.

Ans.

<u>Variable</u>	<u>Scope</u>
a,b	-> file scope
ch	-> function scope (main())
i	-> block scope
gr	-> function scope of f1()
x,y	-> function scope of f1()

28. Write a function that interchanges the value of two integers A and B without using any extra variable.

Ans.

```
void swap(int x,int y)
{
      x=x+y;
      y=x-y;
      x=x-y;
      cout<<"\n The swaped value in x = " << x << " and y = " << y;
}
```

29. **Differentiate between CALL by reference and CALL by value.**

Ans. In call by value, the method copies the values of actual parameters, whereas in call by reference, a reference or the address of the original variable is passed.

30. **Write a function which will take a string and returns the world count. Each word is separated by a single space.**

Ans.

```
int word_cnt(char str[50])
{
     int i,c=1;
     i=0;
     while(i<strlen(str))
     {
          if(str[i]==' ')
          {
               while(str[i]==' ')
                    i++;
               c++;
          }
          i++;
     }
     return c;
}
```

31. **Write a function which will take the height of the person in inches and return the height in feet and inches in two separate variables.**

Ans.

```
void convert(int feet)
{
     int f,i;
     f=feet/12;
     i=feet%12;
     cout<<f<<" feet "<<i<<" inches";
}
```

32. **What is the significance of empty parentheses in a function declaration?**

Ans. It shows that the function is not expecting any argument.

TYPE B: SHORT ANSWER QUESTIONS

1. **List the steps you would follow using a function. Answer your question with the help of an example.**

Ans. Before using a function in C++, three things that are required are:

1. Function Declaration to specify the function's interface to the program.
2. Function Definition to tell the program about what and how a function is doing.
3. Function call to invoke the function.

For example,

```
int sum(int a,int b);      //Function Declaration
int sum(int a,int b)       //Function Definition
{
     return a+b;
}
void main()
{
     int res;
     res=sum(5,2);    //Function Call
}
```

2. **What is role of void keyword in declaring functions?**

Ans. Void data type specifies an empty set of values and it is used as the return type for functions that do not return a value. Thus, a function that does not return a value is declared as follows:

void function-name(parameter list);

A function that does not require any parameter can be declared as follows:

type function-name(void);

3. **Describe the different styles of function prototypes in C++ using appropriate examples.**

Ans. A general form of function prototype is as shown below:

type function-name(parameter list);

In a function prototype, the names of the arguments are optional. Following, are some examples of function prototypes:

```
float volume(int a,float b,float c);
float area(float, float);
float power(int m,int n=2);
int sum(const int a,const int b);
int absval(int a);
```

4. **What do you understand by default arguments and constant arguments? Write a short note on their usefulness.**

Ans. C++ allows us to assign default value to a function’s parameter which is useful in case a matching argument is not passed in the function call statement. The default values are specified at the time of function declaration. For example,

```
     float interest(float p, int t, float r=0.10);
```

Constant argument means that the function cannot modify these arguments. In order to make an argument constant to a function, the keyword const is used. For example,

```
     int sum(const int a, const int b);
```

The constant arguments are useful when functions are called by reference.

5. **How is call-by-value method of function involving different from call-by-reference method? Give appropriate examples supporting your answer.**

Ans.

Call By Value	Call by reference
✓ Call by value is used to create a temporary copy of the data which is transferred from the actual parameter in the final parameter.	✓ Call by reference is used to share the same memory location for actual and formal parameters
✓ The changes done in the function in formal parameter are not reflected back in the calling environment.	✓ The changes done in the function are reflected back in the calling environment.
✓ It does not use & sign	✓ It makes the use of the & sign as the reference operator.

Example:

```
void compute (int A, int & B)
{
         A++;
         B++;
        cout<<“The function on display gives “;
        cout<<“A = “<<A<<“&”<<“B=”<<B<<endl;
}
void main( )
{
        int I=50, J=25;
        cout<<“Initial of function call “<<endl;
        cout<<“I=”<<I<<“&”<<“J=”<<J<<endl;
        compute(I,J); cout<<“After the call of the function”<<endl;
        cout<<“I=”<<I<<“&”<<“J=”<<J<<endl;
        getch( );
 }
```

6. **Write a C++ function that intakes two arguments: a character and an integer and prints the character given number of times. If, however, the integer is missing, the function prints the character twice.**

Ans.

```
void fun(char a,int n=2)
{
     for(int i=0;i<n;i++)
     {    cout<<a<<endl;
     }
}
void main()
```

```
{
      fun('A',4);
       getch();
}
```

7. Comment on passing of arrays as arguments to C++ function. Support your answer taking an example.

Ans. C++ does not allow passing an entire array as an argument to a function. It is possible to pass a pointer to an array by specifying the array's name without an index.

A single dimensional array can be passed in functions arguments in three ways: as a pointer, as a sized array or as an unsized array, and all three declaration methods produce similar results because each tells the compiler that an integer pointer is going to be received.

For example,

```
      double getAverage(int arr[], int size)
      {  int    i, sum = 0;
         double avg;
         for (i = 0; i < size; ++i)
         {
             sum += arr[i];
         }
         avg = double(sum) / size;
         return avg;
      }
```

8. Give the output of the following program:

```
#include<iostream.h>
void sumfn(int last)
{     auto int sum=0;
      static int sum2=0;
      for(int i=last;i>=0;i--) sum+=i;
           sum2+=sum;
      cout<<sum<<" "<<sum2<<endl;
}
void main()
{
      for(int i=1;i<11;i++) sumfn(i);
}
```

Ans. Output:

```
1     1
3     4
6     10
10    20
15    35
21    56
28    84
36    120
45    165
55    220
```

9. What is the output of the following program? Justify your answer.

```
#include<iostream.h>
#include<string.h>
void chg(char * &nm)
{
      strcpy(nm,"kush");    //copy "kush" to nm
}
void main()
{
      char name[]="sandeep";
      cout<<name<<"\t"<<endl;
```

```
     chg(name);
     cout<<name<<"\t"<<endl;
     return 0;
}
```

Ans. Output:

```
Sandeep
Kush
```

10. **Explain the output of the following program:**

```
#include<iostream.h>
int &max(int &x, int &y)
{
     if(x>y)
          return(x);
     else
          return(y);
}
void main()
{
     int A=10,B=13;
     max(A,B)=-1;
     cout<<"A="<<A<<"B="<<B<<endl;
     max(B,A)=7;
     cout<<"A="<<A++<<"B="<<B--<<endl;
     max(A,B)=3;
     cout<<"A="<<A<<"B="<<B<<endl;
}
```

Ans. Output:

```
A = 10,     B = -1
A = 7,      B = -1
A = 3,      B = -2
```

11. **Give the output of the following code. Justify your answer.**

```
#include<iostream.h>
int m=5;
void main()
{
     int m=20;
     {
          int m=10* ::m;
          cout<<"m="<<m<<",::m="<<::m<<endl;
     }
     cout<<"m="<<m<<", ::m="<<::m<<endl;
}
```

Ans. Output:

```
m = 50  :: m = 5
m = 20  :: m =5
```

12. **Explain the output of the following program:**

```
#include<iostream.h>
#include<string.h>
void Execute(int &X,int Y=200)
{    int TEMP=X+Y;
     X+=TEMP;
     if(Y!=200)
          cout<<TEMP<<X<<Y<<endl;
}
void main()
{    int A=50,B=20;
```

```
      Execute(B);
      cout<<A<<B<<endl;
      Execute(A,B);
      cout<<A<<B<<endl;
}
```

Ans.

1. On first call of execute() X receives reference of B and as 2nd paramter is missing so default value of Y i.e. 200 is used. Then sum of X i.e. 20 and Y i.e. 200 is stored in TEMP, after that value of TEMP i.e. 220 is add to the value of X i.e. 20 so now the the value of X is 220+20=240. As X holds the reference of B so the value of B also became 240. And variable A doesn't have any operation with so it is same as it was i.e. 50. So the first output is 50 240.
2. Second out is 290 340 240 because the value of B is changed due to the first execute method and its current value is 240 so the if condition returns true because B is passed to Y and if condition Y is not equals to 200 it is 240. Then Temp = X+Y means 50+240=290. A becomes 340 due 290+50. And as there is no calculation with Y it is still 240 because it contains the value of B i.e. 240.
3. Finally cout<<A<<B<<endl; prints the current value of A and B i.e. 340 and 240 as the original values are manipulated by Execute(B); and Execute(A,B);

Output:

```
50   240
290  340  240
340  240
```

13. **Figure out the errors in the following code fragment:**

```
:
int Sum(int A[])
int s=0;
{
      for(int i=0;i<5;++i)
            s+=A[i];
      returns;
}
void main()
{
      int Val[5],R;
      for(int i=0;i<5;++i)
            cin>>Val[i];
      R=Sum(&Val);
      cout<<"Sum="<<R;
}
```

Ans. The global variable declaration should be first statement in the program.

It should be return instead of returns and it should have some value.

There should be no ‘&’ operator I function call.

Following is the correct code:

```
int s=0;
int Sum(int A[])

{
      for(int i=0;i<5;++i)
            s+=A[i];
      return s;
}
void main()
{
      int Val[5],R;
```

```
        for(int i=0;i<5;++i)
                cin>>Val[i];
        R=Sum(Val);
        cout<<"Sum="<<R;
}
```

14. **Write a function that takes a double array name and an array size as arguments and that swaps the first and last value in that array.**

Ans.

```
void swap(double Arr[ ],int n)
{
        int temp;
        temp=Arr[0];
        Arr[0]=Arr[n-1];
        Arr[n-1]=temp;
}
void main()
{
        double Arr[100],n;
        cout<<"Enter number of elements you want to insert ";
        cin>>n;
        for(int i=0;i<n;i++)
        {
                cout<<"Enter element "<<i+1<<":";
                cin>>Arr[i];
        }
        swap(Arr,n);
        cout<<"\nArray after swapping"<<endl;
        for(i=0;i<n;i++)
                cout<<Arr[i]<<" ";
        getch();
}
```

15. **Write a function that takes three arguments: the name of a float array, the array size, and a float value. Have the function set each element of the array to float value?**

Ans.

```
void fun(float arr[],int n,float value)
{
        for(int i=0;i<n;i++)
        {
                arr[i]=value;
        }
}
void main()
{       float arr[20],n,value;
        cout<<"Enter how many elements: ";
        cin>>n;
        cout<<"Enter value:";
        cin>>value;
        fun(arr,n,value);
        for(int i=1;i<=n;i++)
        {
                cout<<"Element "<<i<<": "<<value;
                cout<<endl;
        }
}
```

16. **Write a function that takes an int argument and doubles it. The function does not return a value.**

Ans.

```
void fun(int a)
{       int b=a*a;
        cout<<b;
}
void main()
{    int a;
     cout<<"Enter a: ";
     cin>>a;
     fun(a);
}
```

17. **Write a function that takes two char arguments and returns 0 if both the arguments are equal. The function returns -1 if the first argument is smaller than the second and 1 if the second argument is smaller than the first.**

Ans.

```
int fun(char a,char b)
{
     if(a==b)
          return 0;
     else if(a<b)
          return -1;
     else if(a>b)
          return 1;
}
void main()
{
     char a,b;
     int res;
     cout<<"Enter first character: ";
     cin>>a;
     cout<<"Enter second character: ";
     cin>>b;
     res=fun(a,b);
     cout<<res;
}
```

18. **Write a function to take an int argument and return 0 if the given number is prime otherwise return -1.**

Ans.

```
int prime(int n)
{    int f=0;
     for(int i=2;i<n/2;i++)
     if(n%i==0)
     {    f=1;
          goto lb;
     }
     lb:
     if(f==0)
          return 0;
     else
          return -1;
}
void main()
{    int n,res;
     cout<<"Enter n: ";
     cin>>n;
     res=prime(n);
     if(res==0)
          cout<<"Prime";
     else
          cout<<"Not prime";
}
```

19. **Write a function to receive an int array, its size and a character '+' or '-'. By default, the character should be '+'. For the character '+', the function returns the sum of positive numbers stored in the array and for the character '-', the function returns the sum of negative numbers stored in the array.**

Ans.

```
int funct(int a[], int n, char s = '+')
{
    int sum = 0;
    for(int i = 0; i< n ; i++ )
    {
        if(s == '+')
        {
            if(a[i] > 0 )
                sum+=a[i];
        }
        else
        {
            if(a[i]<0)
            sum+=a[i];
        }
    }
    return sum;
}
void main()
{
    int arr[20], dn;
    cout<<"\n Enter dimension :";
    cin>> dn;
    for(int i = 0; i< dn ; i++)
    {
        cout<<"\n Enter any interger (positive / negetive )";
        cin>> arr[i];
    }
    /* ... Sum of all positive integers in the array ... */
    int s = funct( arr, dn);
    /* .... Sum of all negative integers ..*/
    int s2 = funct(arr, dn, '-');
    cout<< "\n The sum of the positive integers : " << s;
    cout<< "\n The sum of all negative integers : " << s2;
    getch();
}
```

20. **Write a function that takes a character argument and prints it number of times equal to number of times function has been called to the point.**

Ans.

```
void fun(char a)
{
    cout<<a<<endl;
}
void main()
{
    int n;
    char ch;
    cout<<"Enter how many times: ";
    cin>>n;
    cout<<"Enter character: ";
    cin>>ch;
    for(int i=1;i<=n;i++)
    {
        fun(ch);
    }
    getch();
}
```

21. **Write a function that takes two int arguments and returns reference of the odd number out of the two. If both**

the arguments are odd, then the reference of the smaller one is returned.

Ans.

```
int &setodd(int &a,int &b)
{     if((a%2!=0) && (b%2!=0))
      {
            if(a<b)
                  return a;
            else
                  return b;
      }
      else if(a%2!=0)
            return a;
      else
            return b;
}
void main()
{     int a,b,res;
      cout<<"Enter a and b: ";
      cin>>a>>b;
      res=setodd(a,b);
      cout<<"Odd number is: "<res;
}
```

22. **What all kinds of scope is supported in C++?**

Ans. There are four types of scopes provided by C++:

1. Local scope: A local variable scope is restricted to the function that declares the variable.
2. Function scope: The variables declared in the outermost block of a function have function scope.
3. File scope: A name declared outside all blocks and functions has file scope.
4. Class scope: A name of a member has class scope and is local to its class.

23. **Discuss the similarities and differences between global and local variables in terms of their lifetime and scope.**

Ans. **Difference:**

- ✓ The lifetime of the global variable is the program-run whereas, the lifetime of local variable having function scope is the function-run and having block scope is block-run.
- ✓ The scope of global variable is the entire program file whereas, the scope of local variable is the function which declares them.

Similarities:

- ✓ In terms of lifetime and scope there are no similarities.

24. **Write a function having this prototype : int replace(char * string, char ch1, char ch2); Have the function replace every occurrence of ch1 in the string with ch2, and have the function return the number of replacements it makes.**

Ans.

```
int replace(char *string,char ch1,char ch2)
{
      int l,c=0;
      l=strlen(string);
      for(int i=0;i<=l;i++)
      {
            if(string[i]==ch1)
            {
                  string[i]=ch2;
                  c++;
            }
      }
      cout<<"New string: "<<string;
      return c;
}
void main()
```

```
{
        char str1[20];
        char ch1,ch2;
        cout<<" Enter string:\n1: ";
        cin>>str1 ;
        cout<<"Enter character u want to change: ";
        cin>>ch1;
        cout<<"Enter new character: ";
        cin>>ch2;
        int res=replace(str1,ch1,ch2);
        cout<<"Number of replacement: "<<res;
        getch();
}
```

25. **Complete the following function power() by filling up the correct symbols/expressions/variables at places indicated by ________ . The function power() is declared as follows:**

```
        long power(int x,int n);
```

The power() works as given below:

power (0,n) = 0 for each n

power (x,n) = x^n if x=0 and n>0

power (x,0) = 1 for each x

The function definition is as follows:

```
        long power(int x,int n)
        {
                long res=1;
                if(x= _ _ _ _ )
                        res=0;
                else if(n= _ _ _ _ 0)
                        res=1;
                else if(n= _ _ _ _ _ 0)
                for(int i=0;i<n;i++)
                        res*=x;
                else
                {
                        for(int j=0;j>-n;i--\)
                                res*==x;
                        res=1/res;
                }
                return res;
        }
```

Ans.

```
long power(int x,int n)
        {
                long res=1;
                if(x==0)
                        res=0;
                else if(n==0)
                        res=1;
                else if(n>0 && x!=0)
                for(int i=0;i<n;i++)
                        res*=x;
                else
                {
                        for(int j=0;j>-n;j--)
                                res*=x;
                        res=1/res;
                }
```

```
           return res;
     }
```

26. **Write a C++ function to sum n natural numbers starting from a given number.**

Ans.

```
int sum(int N)
{        int S = 0;
         for(int i = 0; i <= N; i++)
                 S += i;
         return S;
}
void main()
{       int n;
        cout << "Enter the number: ";
        cin >> n;
        cout << sum(n) << endl;
}
```

27. **Write a C++ function to find least common divisor of two integers.**

Ans.

```
void least_div(int a,int b)
{    int i=2,k=-1;
      while(k==-1 && i<(a+b)){
         if(a%i==0 && b%i==0)
                k=i;
             i++;

                 }
           if(k!=-1)
                  cout<<"Least common divisor:  "<<k<<"\n";
           else   cout<<"No common divisor\n";
}
Void main()
{    int a,b;
      cout<<"Enter a: ";
      cin>>a;
      cout<<"Enter b: ";
      cin>>b;
      least_div(a,b);
}
```

28. **Write a C++ function that compares two string and returns 0 if the two strings are equal and -1 if the string are unequal.**

Ans.

```
int compare(char str1[], char str2[])
{
     if ( strcmp (str1,str2) == 0)
           return 0;
     else
           return -1;
}
void main()
{
     char str1[20];
     char str2[20];
     int res;
     cout<<" first string:\n1: ";
     cin>>str1 ;
     cout<<" second string:\n 2: ";
     cin>>str2;
     res=compare(str1,str2);
     if(res==0)
           cout<<"Equal strings";
     else
          cout<<"Not equal";
     getch();
}
```

TYPE C : LONG ANSWER QUESTIONS

1. **Write a complete C++ program that reads a float array having 15 elements. The program uses a function reverse() to reverse this array. Make suitable assumptions wherever required.**

Ans.

```
#include<iostream.h>
#include<conio.h>
void rev(float a[], int n)
{
     float t;
     for(int i =0, k = n-1; i< n/2 ; i++, k--)
     {
          t = a[i];
          a[i] = a[k];
          a[k] = t;
     }
}
void main()
{
     float arr[15];
     cout<< "\n Enter 15 real numbers :";
     for(int i = 0; i< 15 ; i++)
     cin>> arr[i];
     clrscr();
     cout<<"\n The original array : \n";
     for( i = 0; i< 15; i++)
     cout<< arr[i] << "  ";
     cout<<"\n";
     rev(arr, 15);
     cout<<"\n The reversed array : \n";
     for( i = 0; i< 15; i++)
      cout<< arr[i] << " ";
     getch();
}
```

2. **Write a complete C++ program that invokes a function satis() to find whether four integers a, b, c, d send to satis() satisfy the equation $a^3 + b^3 + c^3 = d^3$ or not. The function satis() returns 0 if the above equation is satisfied with given four numbers otherwise it returns -1.**

Ans.

```
#include<iostream.h>
#include<conio.h>
int satis(int a, int b, int c, int d)
{
     if( ((a*a*a) + (b*b*b) + (c*c*c)) == d*d*d)
          return 0;
     else
          return -1;
}
void main()
{
     int x, y, z, w;
     cout<< "\n Enter 4 integers : ";
     cin>>x>>y>>z>>w;
     int s = satis(x,y,z,w);
     if(s == 0)
          cout<< "\n The equation is satisfied";
```

```
      else
           cout<< "\n The equation is NOT satisfied";
      getch();
}
```

3. **Write a complete C++ program that uses a function called carea() to calculate area of circle. The function carea() receives radius of float type and return are of double type. The function main() gets a radius value from the user, calls carea(), and display the result. The function carea() is local to main().**

Ans.

```
#include<iostream.h>
#include<conio.h>
double carea(float r)
{
      double ar = (double) ( 3.14 * r * r );
      return ar;
}
void main()
{
      double carea(float r);
      float rad;
      cout<<" \n Enter Radius :";
      cin>>rad;
      double area = carea(rad);
      cout<<"\n The area of the circle of radius " << rad << " unit is " <<
      area << "sq. unit ";
      getch();
}
```

4. **Write a C++ program that uses a function smallo() (that is passed two int argument by value) to receive reference of the smaller value. Then using this reference the smaller value is set to 0. Write a main() function also to exercise this function.**

Ans.

```
#include<iostream.h>
#include<conio.h>
int &smallo(int &a, int &b)
{
      if(a<b)
        return a;
      else
        return b;
}
void main()
{
      int a, b;
      cout<< "\n Enter two numbers :";
      cin >> a >> b;
      cout<< "\n The original values :" << a << "  and " << b;
      smallo(a, b) = 0;
      cout<< "\n The changed values :" << a << " and " << b;
      getch();
}
```

5. **Write a C++ program that uses following functions:**
(i) sqlarge() that is passed two int argument by reference and then sets the larger of the two umbers to its square.
(ii) sum() that is passed an int argument by value and that returns the sum of the individual digits of the passed number.
(iii) main() that exercise above two functions by getting two integers from the user and by printing the sum of the individual digit of the square of the larger number.

Ans.

```
void sqlarge(int &a, int &b)
{
      if(a >b)
```

```
         a = a*a;
      else
         b = b*b;
}
int sum(int x)
{
      int r, s=0;
      while(x > 0)
      {
           r = x % 10;
           s = s+r;
           x = x / 10;
      }
      return s;
}
void main()
{
int num1, num2,num3;
cout<<" \ n Enter a number :";
cin>>num1;
int tot = sum(num1);
cout<< " \n THE SUM OF DIGITS = " << tot;
cout << "\n Enter two numbers : ";
cin>> num2>>num3;
cout<< "\n The two numbers originally are " << num2 << " and " << num3 ;
sqlarge(num2, num3);
cout<< "\n The two numbers after change are " << num2 << " and " << num3 ;
getch();
}
```

6. **Write a C++ program to use the following function:**
(i) display() to display a matrix of size m x n.
(ii) times2() to double each number of the matrix of size m x n.
(iii) main() to read a matrix of size m x n and then to display original matrix and then to display the new matrix formed by doubling its elements.

Ans.

```
#include<iostream.h>
#include<conio.h>
void display(int a[10][10], int m, int n)
{
      for(int i = 0; i< m ; i++)
       {
        for( int j = 0; j< n ; j++)
              cout << a[i][j] << "  ";
         cout<< endl;
       }
}
void times2(int a[10][10], int m, int n)
{
      for(int i = 0; i< m ; i++)
        for( int j = 0; j< n ; j++)
           a[i][j] = a[i][j] * 2;
}
void main()
{
      int mat[10][10], row, col, i, j;
      cout<< "\n Enter total rows :";
      cin>> row;
      cout<<" \n Enter total columns :";
```

```
     cin>> col;
     cout<< "\n Enter the elements for Matrix:";
     for(i = 0; i< row; i++)
       for(j = 0; j< col; j++)
          cin>>mat[i][j];
     cout<<"\n The Original Matrix :";
     display(mat, row, col);
     times2(mat, row, col);
     cout<<"\n The New Matrix :";
     display(mat, row, col);
     getch();
}
```

7. **Write a program uses a function power() to raise a number m to power n. The function takes int values for m and n returns the result correctly. Use a default value of 2 for n to make the function calculate squares when this argument is omitted. Write a main() to get the value of m and n from the user and to display the calculated result.**

Ans.

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
double power(int m, int n = 2)
{
     double res = pow(m,n);
     return res;
}
void main()
{
     int x,y;
     double d;
     cout<< "\n Enter a number  and its power to calculate the result :";
     cin>>x>>y;
     d = power(x,y);
     cout<< "\n The " << x << " to the power " << y <<" is " << d;
     getch();
}
```