

Set I

1. a) Name the header file for the following built-in functions:
i) `log()` → `math.h` ii) `exp()` → `math.h` iii) `getch()` → `conio.h`
iv) `isalnum()` → `ctype.h`
- b) Give the output of the following program:
14,23,43
1514,3743,6743
- c) i) **3**
6
9
12
15
ii) Obtain the same output by using **for** loop.

```
#include<iostream.h>
#include<conio.h>
void main()
{
    for( int x=3;x<=15;x=x+3)
    {
        cout<<x<<endl;
    }
    getch();
}
```
2. a) i) Write all the possible C++ statements to decrement value of a variable x by 1.
x=x-1;
--x;
x--;
x-=1;
ii) Write a logical expression to check that a character variable alpha contains lowercase alphabet (without using `islower()` function).
alpha >= 'a' && alpha <= 'z'
- b) i) `char, int, float, double`
ii) `int, char`
iii) `void float.`
- c) What is type casting? With suitable example show two ways of type casting.
Converting data from one to another type temporarily inside the CPU. Two ways type casting:
cout<<int('A')<<endl; Or, cout<<(int)'A'<<endl;
- d) Name the escape sequences given below:
i) `'\b'` bell ii) `'\n'` new line
3. a) In a **switch-case** statement, which are the optional components?
default and break

b) i) What is the difference between entry controlled loop and exit controlled loop?

Entry Level Loop	Exit Level Loop
If the control statement or control condition is present or checked before the execution of the body of the loop, then it is called entry controlled loop.	If the control condition is present after the execution of the body unconditionally for the first time, then it is called exit controlled loop.
EG:- entry controlled loop: int i=0;int n=10; while(i<n) { i++; } here the condition is checked using while statement and the entry is gained based on it.the statement i++ is executed 10 times	EG;-exit controlled loop; int i=0;int n=10; do { i++; }while(i<n); here the statement i++ is executed unconditionally for the first time and then condition statement is added.

ii) Name one entry controlled loop and one exit control loop.

Example Entry level → while loop	Example Exit level → do while Loop
----------------------------------	------------------------------------

c) i) What is conditional operator? What is the other name for conditional operator?

Conditional operator is used in place of if-else statement Conditional operator needs three operands and two (2) operators. Conditional operator is also known as Ternary operator

Rule: Condition? Action1: Action2

Condition or Logical Expression is evaluated and if the Condition is TRUE then Action1 executed otherwise Action2 is executed.

ii) `int osum=0, esum=0, num;`
`for (int k=1; k<=20; k++)`
`{`
`cin>>num;`
`(num%2==1)? osum+=num : esum+=num;`
`}`
`cout<<osum<<esum;`

4. a) i) Mention two differences between local variable and global variable.

Global Variable	Local Variable
• A variable created immediately after the header files and before any block	• A variable created inside a block (block of if-else, loop and function block)
• Visible throughout the program – main() function and user defined functions	• Visible inside the block and blocks nested below
• Longevity is as long as the program is active	• Longevity is as long as the block is active
• Default value of a global variable is Zero	• Default value of a local variable is garbage

ii) When is scope resolution operator necessary with global variable?

Inside a block, if there is a local variable and global variable having same name then the local variable is preferred to the global variable. To use both the local variable and global variable we need scope resolution operator (::) with the global variable name.

b) i) Write two differences between actual parameter and formal parameter.

Actual Parameter	Formal Parameter
• Parameter used in function invocation	• Parameter used in function definition
• Actual parameter may be a variable or an expression or a constant	• Formal parameter is always variable (or an alias)

ii) State any two differences between value parameter and reference parameter.

Value Parameter	Reference Parameter
• Copy of actual parameter	• Alias of actual parameter
• Change in value parameter does not change actual parameter	• Change in reference parameter, updates actual parameter
• Transfer of data is one way, from calling function to called function	• Transfer of data is two ways, from calling function to called function and vice-versa
• Actual parameter may either be a variable or an expression or a constant	• Actual parameter can only be a variable, it cannot be constant or expression

c) i) Differentiate between function prototype and function definition.

A function prototype also called function declaration is a function header (declarator) terminated by a semi-colon where a function header consists of function name, return value of a function and optional list of parameters.

Example: `double factorial(int);`

A function definition is the complete function, that is, header and the body. A function declaration must appear above any use of the function's name. But function's definition, when listed separately from the function's declaration, may appear anywhere outside the body or block of main() function.

Example:

```
double factorial(int n)
{
    double fact=1;
    for (int k=1; k<=n; k++)
        fact*=k;
    return fact;
}
```

ii) With an example, differentiate between calling function and called function.

A function which invoke other function within it is called calling function.

A Function which Invoked inside calling function is called function

Either function prototype or function definition of called function must be declared before calling function otherwise compiler give error message.

Values passed from calling function to called function through parameter of called function

Example

```
#include<iostream.h>
double factorial(int n)
{
    double fact=1;
    for (int k=1; k<=n; k++)
        fact*=k;
    return fact; }
```

```

void main() // calling function  main is a calling function
{
int m;
cout<<"Input an integer? ";
cin>>m;
double f=factorial(m); // factorial is called function
cout<<m<<"!="<<f<<endl;
}

```

5. Write C++ program for the following:

- a) Test whether an integer passed as parameter to a function **int** checkpalindrome(), is a Palindromic number or not. Function returns a value either 1 (one) when the integer is Palindrome or 0 (zero) when the integer is not Palindrome.

```

int checkpalindrome (int n)
{
    int m=0, temp=n;
    while (n!=0)
    {
        m=10*m+digit+n%10;
        n/=10;
    }
    return(m==temp)
}

```

Test whether an integer input by user is a Palindromic number or not **without function**

```

#include<iostream.h>
void main()
{
    int n;
    cout<<" Enter the Number"
    cin>>n;
    int m=0, temp=n;
    while (n!=0)
    {
        m=10*m+digit+n%10;
        n/=10;
    }
    if(m==temp)
    cout<< "Palindromic Number";
    else
    cout<< "Not Palindromic Number";
}

```

- b) Write a program to generate the series: $1 - \frac{x}{1!} + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots (-1)^n \frac{x^n}{n!}$

```

#include<iostream.h>
void main()
{
    int n, f=1,a=1;
    double x, sum=1.0,p=1.0;
    cout<<"Input an integer? ";cin>>n;
    cout<<"input the value of x"; cin>>x;
    double t=x;
    while(a<=n)
    {
        p=p*a;
        sum+= (-1)* t/p;
        t=(-1)*x*t;
        a=a+1;
    }
    cout<< "Sum of the Series = " <<sum;
}

```

- c) Calculate HCF and LCM of two integers passed as parameters to function `hcf lcm()`. Return value of the function **void**. If one of the parameters is non-positive then display an error message “HCF & LCM do not exist”. The HCF and LCM of two integers is to be displayed inside the function.

```
void hcf lcm(int a, int b)
{
    if (a<=0 || b<=0)
        cout<<" HCF /LCM not possible";
    else
    {
        int r, p=a*b;
        do
        {
            r=a%b;
            a=b;
            b=r;
        }while (r>0);
        cout<<"HCF="<<a<<endl;
        cout<<"LCM="<<(p/a)<<endl;
    }
}
```

Without function

```
#include<iostream.h>
void main( )
{
    int a,b;
    cout<< "Two Nos";
    cin>>a>>b;
    if (a<=0 || b<=0)
        cout<< " HCF /LCM not possible";
    else
    {
        int r, p=a*b;
        do
        {
            r=a%b;
            a=b;
            b=r;
        }while (r>0);
        cout<<"HCF="<<a<<endl;
        cout<<"LCM="<<(p/a)<<endl;
    }
}
```

- d) Generate and display all the Prime numbers between 2 and n (including 2 and n), where n is passed as a parameter to the function. Return value of the function is **void**. Name of the function is `genprime()`.

```
void generateprime(int n)
{
    for(int k=2; k<=n; k++)
    {
        int x=2, prime=1;
        while(x<n && prime==1)
        {
            if (n%x==0)
                prime=0;
            x++;
        }
        if (prime==1)
            cout<<k<<endl;
    }
}
```

Without function

```
#include<iostream.h>
void main( )
{
    int n;
    cout<< " Enter the No:";
    cin>>n;
    for(int k=2; k<=n; k++)
    {
        int x=2, prime=1;
        while(x<n && prime==1)
        {
            if (n%x==0)
            {
                prime=0;
            }
            x++;
        }
        if (prime==1)
            cout<<k<<endl;
    }
}
```