Set –I

Ans 1	fabs()	math.h	isalnum()	ctype.h
	sqrt()	math.h	getch()	conio.h

Note do not use #include< > with header file only name is required if you use that marks will be deducted

Q2.

Line 3	**missing {**	Correct code **{**
Line 5	cout **>>** "Input an integer ?";	Correct code is cout**<<**
Line 6	cin>>n and **n** not declared before use	Correct code it should be declared in line 4 or before the use
Line 9	cout<< "Sum="<<**s**<<endl; **s** not declared	Correct code is cout<<"Sum="<<**sum**<<endl

Q3.
1
3
5
7
9
11

Same output with loop code

while loop

```
void main()
{
        int x=1
        while(x<=11)
        {
        cout<<x<<endl;
        x=x+2;
        }
}
```

for loop

```
void main()
{
        for( int x=1;x<=11x=x+2)
        {
        cout<<x<<endl;
        }
        }
```

Q4. **Comment**: Non executable statements of a C++ program are called Comments. Comments are also known as Remarks. A Comment is completely ignored by a compiler. No code is generated for a Comment.

Single line Comment: also known as C++ style Comments. Single Line Comment starts with pair of forward slash (//) and till the end of line is considered as a Comment. Examples of Single Line Comment are given below:

```
// single line comment
```

Multi-line comment: also known as C style comments. Multi-line comment start with forward slash and star (/*) and with star and forward slash (*/). Examples of Multi-Line Comment are given below:

```
/* multi-line comments
    comment in C style
*/
```

Q 4 B

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.
Example While loop	Example Do While 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.

Q 4 C

Run-time error:	**Logical error**:
Syntactically correct statement performs illegal operation during execution of a program is called Run-Time errors. Illegal operation is performed when the program encounters unexpected data. Run-Time errors are triggered when **running** the program.	An error in program design or program implementation that does not prevent your program from compiling, but causes it to do something unexpected.
Examples of Run-Time errors are given below: a) Division by zero (0) b) Square root of a negative number c) Logarithm of zero (0) or negative number d) Inputting character / string value when integer or floating point value is expected	Examples of Logical errors are given below: a) Variables with incorrect or unexpected values b) Accumulator or counter not initialised c) Incorrect placement of braces (curly brackets) for a block d) Missing parenthesis when parenthesis are required e) In a logical expression using = instead of ==

5(a)
```
#include<iostream.h>
void main()
{
  int n, f=1,a=1;
  float x, sum=0.0,
  cout<<"Input an integer? ";cin>>n;
  cout<<"input the value of x"; cin>>x
  while (a<=n)
  {
  f=f * a;
  p=p * x* x;
  sum=sum + p/f;
  a=a+1;
  }
  cout<< "Sum of the Series = " <<sum;
}
```

5 (b)
```
#include<iostream.h>
void main()
{
  int n;
  cout<<"Input an integer? ";cin>>n;
  int k=2, prime=1;
  while (k<n && prime==1)
  {
      if (n%k==0)
          prime=0;
      k++;
  }
  if (prime==1)
      cout<<n<<" Prime Number"<<endl;
  else
      cout<<n<<" Composite Number"<<endl;
   }
```

Q5 (c)
```
#include<iostream.h>
  void main()
  {
          int n, digit=0, sum=0, prod=1;
          cout<<"Input an integer? ";cin>>n;
          while (n!=0)
          {
          int r=n%10;
          if(r%2 = =0)
          sum+=r;
          else
          prod*=r;
          n/=10;
  }
  cout<<"Sum of Even digits="<<sum<<endl;
  cout<<"Product of odd digits="<<prod<<endl;
}
```

Set-II

Q1	log10()	math.h	isalnum()	ctype.h	gets()	stdio.h
	endl	iostream.h	sqrt()	math.h	toupper()	ctype.h

Q1b	Line 1 # include iostream.h	Correct code #include**<**iostream.h**>**
	Line 2 **void()**	correct code void **main()**
	Line 5 cin> n;	correct code cin**>>**n
	Line 6 for(int c**==**2;c<=2*n;c+=2)	correct code for(int c**=**2;c<=2*n;c+=2)
	Line 7 sum ++ c;	correct code sum **+=** c;
	Line 8 cout<<**'**sum=**'**<<sum<<endl;	correct code cout<< **"**sum=**"** <<sum<<endl;

Q2a Output
12,26,46
312,1196,358

Q2b Output
2 2,2
4,6,8
8,14,22
16,30,52
128,158,210

Q2c Output
35,36,38
37,36,36

Q3a default , break

Q3b + and –

Q3c 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.
While loop is example of entry controlled loop

Q3d **Syntax error**: error committed when the syntax of the language (grammar of the language) is violated. Examples of Syntax errors are given below:

a) Typographical mistakes
b) Omitted semicolons or coma
c) References to undeclared variables
d) Wrong number or type of parameters passed to a function
e) Call to undefined function

Syntax errors are detected by the compiler. Syntax errors are also known as **Compile-Time** errors because the errors are flagged by the compiler during compilation time

Q4a

switch case	Ternary operator

switch(x<=y)
{
case 0: lo=y;
break;
case 1: lo=x;
break;
}
cout<< "Minimum Value="<<lo<<endl;
	lo= (x<=y) ? x:y;
	cout<<"Minimum Value ="<<lo<<endl;

4 b

switch case	Ternary operator

switch(year%4==0 && year%100!=0)
{
case 0:
cout<< “Not a leap Year”;
break;
case 1:
cout<< “Leap Year”;
}
	cout<<(year%4==0 && year%100!=0)**?** “Leap year” **:** “Not a Leap Year”;

5 a) char **1 byte** int **4 bytes** float **4 bytes** double **8 bytes**

5b) float does not support any modifiers. Double support only long

5c) a) float or double b) int c)float or double d)float or double

4d)**Comment**: Non executable statements of a C++ program are called Comments. Comments are also known as Remarks. A Comment is completely ignored by a compiler. No code is generated for a Comment.

Single line Comment: also known as C++ style Comments. Single Line Comment starts with pair of forward slash (//) and till the end of line is considered as a Comment. Examples of Single Line Comment are given below:

```
// single line comment
```

Multi-line comment: also known as C style comments. Multi-line comment start with forward slash and star (/*) and with star and forward slash (*/). Examples of Multi-Line Comment are given below:

```
/* multi-line comments
    comment in C style
*/
```

5e) **Typecasting**: converting data from one type to another type temporarily, inside the processor (CPU). Examples of Type casting are given below:

```
#include<iostream.h>
void main()
{
    int m, n;
    cout<<"Input 2 integers? ";
    cin>>m>>n;
    double r1=double(m)/n;
    double r2=(double)m/n;
    cout<<r1<<','<<r2<<endl;
}
```

6a) Try Yourself

6b)
```
#include<iostream.h>
void main()
{
int n,  S1=0,S2=0, a=1;
cout<<"Input an integer? ";cin>>n;
while (a<=2*n-1)
        {
        S1=S1+a*a;
        S2=S2+S1;
        a=a+2;
        }
cout<<"Sum of Series is :"<<S2;
}
```

6c)
```
#include<iostream.h>
void main()
{
int n, f=1,a=1;
float x, sum=1.0,p=1.0,
cout<<"Input an integer? ";cin>>n;
cout<<"input the value of x"; cin>>x
        p=x;
        while (a<=n)
        {
        f=f * a;
        p=p * x;
        if(a%2!=0)
        {
        sum=sum + p/f;
        }
        a=a+1;
        }
cout<< "Sum of the Series = " <<sum;
  }
```

6d)
```
#include<iostream.h>
void main()
{
int n;
cout<<"Input an integer? ";cin>>n;
int t=n, s=0;
while (n!=0)
{
int digit=n%10;
s+=digit*digit*digit;
n/=10;
}
if (s==t)
cout<<t<<" Armstrong Number"<<endl;
else
cout<<t<<" Not Armstrong Number"<<endl;
}
```

Set-III

Q1 a) tolower()→ ctype.h tan()→ math.h isdigit → ctype.h
Cin → iostream.h labs()→ math.h M_PI→ math.h

b)

Line 1 #include<**stream.h**>	Correct code	#include<**iostream.h**>
Line 2 void main**[ ]**	Correct Code	void main**()**
Line 5 cin>>**n**;	Correct Code	**n should be declared before use**
Line 6 for(int x=1;x<=n;**1+=x**)	Correct Code	for(int x=1;x<=n; **x+=1**)

C) output
73,50,34
510,280,157

d) 1,1
9,10
36,46
100,146
225,371

e) 10881099

Q2. i) (cos(x)+sec(x))/sin(x)-cosec(x) ii) (sqrt(a)+sqrt(b))/(a*a+b*b)
iii) `log10(x) + log10(y)/log10(z)`

b)
```
int num;
cout<<"Input an integer between 0 and 6?";cin>>num;
switch(num)
{
case 0: cout<<"Zero"<<endl;break;
case 1:
case 3:
case 5: cout<<"Odd Number"<<endl; break;
case 2:
case 4:
case 6: cout<<"Even Number"<<endl; break;
default:
     cout<<"Invalid Number"<<endl; break;
}
```

c) **if-else**
```
int x ,y,max;
cout<<"Input an two integer?";cin>>x>>y;
if(x>y)
{
     max=x;
}
else
{
max=y;
}
cout<<" Maximum Value  ="<<max<<endl;
```

Ternary Operator

```
int x ,y,max;
cout<<"Input an two integer?";cin>>x>>y;
max=(x>y)? x: y;
cout<<" Maximum Value  ="<<max<<endl;
```

d)
```
int n,p=1;
cout<<"Input an integer?";cin>>n;
for(int c=1;c<=n;c++)
     p*=c++;
cout<<p<<endl;
```

i) if value of n= -7 output will **1** **ii)** if value of n= 5 output will **15**

3 a) any three of below given rules

Rules for naming a C++ variable (identifier)

1. Variable name should start with an **alphabet (letter) or an underscore.**
2. Variable name may contain more than one character. Second characters onwards we may use only alphabets or digit or underscore.
3. No special characters are allowed in a variable name except underscore.
4. A variable name in C++ is case sensitive. Uppercase and lowercase letters are distinct. `Sum`, `sum`, `SUM` and `sUm` are treated as four different variable names in C++.
5. A variable name cannot be a keyword. **`void`**, **`char`**, **`int`**, **`float`**, **`double`**, **`if`** and **`else`** are incorrect variable names because they are keywords.

b) three Incorrect Identifier are

1) **long** because it is a keyword
2) comp-sc Identifier can not contain any special character (-)
3) 2ndfloor identifier can not start with number
4) cell# Identifier can not contain any special character (#)

C) i) 30.0/4→ float/double ii)"30.4/4"→ String iii) 30/4→ int iv) '4'→ char

d) Mention Two Differences between data type float & type double

float	**double**
Storage allocation 4 bytes or 32 bits	Storage allocation 8 bytes or 32 bits
`float` does not support any type modifiers	Data type **`double`** supports only **`long`**.

```
e) unsigned char A;
   signed char  B;
   short int  n1;
   long int n2;
   signed int n3;
   unsigned int n4;
```

Q4a) **Token:** Building block of a program is called a token. It is also called program element. Tokens of a C++ program can be classified as Keyword, Identifier, Constant, Operator, String and Comment.

Keyword	**Built-in identifier**
It is component of a program which has special meaning for the C++ compiler	It is name of built-in functions, constants, variables, classes and structure. To use built-in identifier we need appropriate header
A keyword cannot be redefined.	**Built-in identifier can be redefined**.

B) I) two unary operator other than + and – are ++ and –

II) RIGHT TO LEFT OPERATOR **?:** (TERNARY OPEARTOR) UNARY OPERATOR **+ , - , !, ++, --**
III) && , ||
IV) * , /, %, ==, !=, <=,>=

c) **Typecasting**: converting data from one type to another type temporarily, inside the processor (CPU). Examples of Type casting are given below:

```
#include<iostream.h>
void main()
{
    int m, n;
    cout<<"Input 2 integers? ";
    cin>>m>>n;
    double r1=double(m)/n;
    double r2=(double)m/n;
    cout<<r1<<','<<r2<<endl;
}
```

D) I) mychar>= ‘0’ && myhar <= ‘9’
ii) number%2!=0 && number%5!=0
iii) marks> 0 && marks<100
iv) alpha= ‘A’ || alpha= ‘E’ || alpha= ‘I’ || alpha= ‘O’ || alpha= ‘U’

Note: Do not use if else statement to show the logical expression marks will be cut if used with if –else only logical statement should be written like above
If(mychar>=0 && myhar <= 9) is worng way and marks will be cut
Correct way is writing simply mychar>=0 && myhar <= 9

Q5 a)
```
#include<iostream.h>
#include<conio.h>
void main()
{
int local,INT, ibill,bill;
cout<<"Enter the No of Local Calls:";
cin>>local;
cout<<"Enter the No of International Calls:";
cin>>INT;
ibill=INT*50;
if(local>0 &&  local<=100)
bill=250;
else if(local>100 && local<=250)
bill=250+(local-100)*3 +10;
else if(local>250 && local<=500)
bill=250+ 150*3 +(local-250)*4+ 25;
else if(local>500)
bill=250+ 150*3 + 250*4+ (local-500)*5 +75;
else
cout<<" ERROR";
cout<<" YOUR LOCAL CALL AMOUNT IS--> "<<bill<<endl;
cout<<" YOUR INTERNATIONAL CALL AMOUNT IS--> "<<ibill<<endl;
cout<<" YOUR MONTHLY BILLT IS--> "<<bill+ibill;
getch();
}
```

b)
```
#include<iostream.h>
#include<math.h>
void main()
{
double a, b, c;
cout<<"Coefficient of x^2? "; cin>>a;
cout<<"Coefficient of x  ? "; cin>>b;
cout<<"Constant Term     ? "; cin>>c;
double d=b*b-4*a*c;
if (d==0)
{
double x=-b/(2*a);
cout<<"Real and equal roots"<<endl;
cout<<"Two root are "<<x<<" and "<<x<<endl;
}
else
if (d>0)
{
double x1=(-b+sqrt(d))/(2*a), x2=(-b-sqrt(d))/(2*a);
cout<<"Real and distinct roots"<<endl;
cout<<"Two root are "<<x1<<" and "<<x2<<endl;
}
else
cout<<"Complex roots"<<endl;
}
```

c)
```
#include<iostream.h>
#include<math.h>
void main()
{
int n;
double x,s=1, p=1, f=1;
cout<<" Enter the Value of x?";cin>>x;
cout<<" Enter the Value of n?";cin>>n;
for (int k=2; k<=2*n; k+=2)
{
p*=x;
f*=k*(k-1);
s+=p/f;
}
```

d)
```
#include<iostream.h>
void main()
{
int n;
cout<<"Input an integer? ";cin>>n;
int t=n, m=0;
while (n!=0)
{
m=10*m+n%10;
n/=10;
}
if (t==m)
cout<<t<<" Palindromic integer"<<endl;
else
cout<<t<<" Not Palindromic integer"<<endl;
}
```

10)
```
#include<iostream.h>
void main()
{
}
int n, digit=0;
cout<<"Input an integer? ";cin>>n;
int t=n;
while (n!=0)
{
int r=n%10;
digit++;
n/=10;
}
cout<<"Inputted Number is="<<t<<endl;
cout<<"number of digits="<<digit<<endl;
}
```

11)
```
#include<iostream.h>
void main()
{
}
int n, s=0;
cout<<"Input an integer? ";cin>>n;
int t=n;
while (n!=0)
{
int r=n%10;
s=s+r;
n/=10;
}
cout<<"Inputted Number is="<<t<<endl;
cout<<"sum of digits="<<s<<endl;
}
```

12)
```
#include<iostream.h>
void main()
{
}
int n, s=0;
cout<<"Input an integer? ";cin>>n;
int t=n;
while (n!=0)
{
int r=n%10;
s=s+r*r;
n/=10;
}
cout<<"Inputted Number is="<<t<<endl;
cout<<"sum of square of digits="<<s<<endl;
}
```

13)
```
#include<iostream.h>
void main()
{
}
int n, s=0;
cout<<"Input an integer? ";cin>>n;
int t=n;
while (n!=0)
{
int r=n%10;
s=s+r*r*r;
n/=10;
}
cout<<"Inputted Number is="<<t<<endl;
cout<<"sum of cube of digits="<<s<<endl;
}
```

14) #include<iostream.h>
```
void main()
{
}
int n, p=1;
cout<<"Input an integer? ";cin>>n;
int t=n;
while (n!=0)
{
int r=n%10;
s=s*r;
n/=10;
}
cout<<"Inputted Number is="<<t<<endl;
cout<<"Product of digits(including 0)="<<s<<endl;
}
```
15) #include<iostream.h>
```
void main()
{
}
int n, p=1;
cout<<"Input an integer? ";cin>>n;
int t=n;
while (n!=0)
{
int r=n%10;
if(r!=0)
{
s=s*r;
}
n/=10;
}
cout<<"Inputted Number is="<<t<<endl;
cout<<"Product of digits(excluding 0)="<<s<<endl;
}
```
16) #include<iostream.h>
```
void main()
{
}
int n,
double s=0;
cout<<"Input an integer? ";cin>>n;
int t=n;
while (n!=0)
{
int r=n%10;
if(r!=0)
{
s=s+1/r;
}
n/=10;
```

```
    }
    cout<<"Inputted Number is="<<t<<endl;
    cout<<"sum of reciprocal of digits(excluding 0)="<<s<<endl;
    }
```

19)
```
#include<iostream.h>
void main()
{
int n;
cout<<"Input an integer? ";cin>>n;
int t=n, s=0;
while (n!=0)
{
int digit=n%10;
s+=digit*digit*digit;
n/=10;
}
if (s==t)
cout<<t<<" Armstrong Number"<<endl;
else
cout<<t<<" Not Armstrong Number"<<endl;
}
```

20
```
#include<iostream.h>
void main()
{
int n;
cout<<"Input n? "; cin>>n;
int f1=1, f2=1;
cout<<f1<<endl;
cout<<f2<<endl;
for (int k=3; k<=20; k++)  //
{
}
int f3=f1+f2;
cout<<f3<<endl;
f1=f2;
f2=f3;
cout<<s<<endl;
}
```

22)
```
#include<iostream.h>
void main()
{
int n;
cout<<"Input an integer? ";cin>>n;
int k=2, prime=1;
while (k<n && prime==1)
{
if (n%k==0)
prime=0;
k++;
}
```

```
if (prime==1)
cout<<n<<" Prime Number"<<endl;
else
cout<<n<<" Composite Number"<<endl;
}
```