Theoretical Part

Q1.what is the difference b/w function declaration and its definition .

Give a short program to demonstrate.

Answer: **Function Declaration:**

The function declaration is also called prototype.

Program:

```
int add();
```

only declare a datatype of function

Function Definition:

Program:

```
int add(){
int a = 10; int b=20;
int c= a+b;
}
```

The body of function is called definition

It include * declarator type of function(parameter list)

Body of function {

Set of statements}

Note: the value pass in definition is calles parameters

The value passed to a function in a call is called arguments.

Question 2: Where can be the declaration of function placed?

Answer: The function declaration is model of the function.It provides model to a function.

The following information is provided to compiler for declaraing a function

The name of Function

The type of data returned by the function

The number and types of parameters used in the function.

So now the important question is that where the declaration of function is placed

The declaration of function is placed after the header file and before the main function.

Because compiler include the header file in program the read the function prototype provided by user. And then go to main function. If function is not declare before the main function instead after a main function then the program shows error. So its necessary to declare a program before main function and the definition of function will be provided after main function the program will work correctly.

Question 4: What is differene b/w passing a parameter by the value and by refernce?

Answer: Passing parameters by value:

In pass by value we make a copy of variables and make changes in that copy and original variable is not changed throughout a program.

```
Void display(int);
Int main(){void display( int y){
        y=10;
        cout<<y;
```

```
}
int main(){
        int j=3;
        display(j);
  return 0;
}
```

The function display 10 because actual value is not change .

Passing value by reference:

Passing arguments by reference is that we store a address of argument .

And in main function we give a refrence of the parameter address. By that the changes occur in actual variable .

```
#include<iostream>
using namespace std;
void display(int & x){      //store a address  of x
        x++;
        cout<<x;
}
int main(){
        int i=2;
        display(i);       the I value is assigned to address of x.
return 0;
}
```

Question 5: Which operator is used in passing parameters by reference?

Answer: The address operator is used to passing parameters by reference.In this we pass the arguments indirectly by giving a address reference.This address is reference . and for that purpose we use address operator.

Program:

```
#include<iostream>

using namespace std;

void display(int & x){      //store a address  of x

        x++;

        cout<<x;

}

int main(){

        int i=2;

        display(i);       the I value is assigned to address of x.



  return 0;


}
```

Program for both pass by value and reference.

```
#include<iostream>
using namespace std;
void display(int & x, int y){   // the first parameter is passed by reference and
        x++;                                    second is passed by value
        y=10;
        cout   <<x<<" "<<y;
}
int main(){
        int i=2,j=3;

        display(i,j);              // the value of i is becomes 3 because I =2 and in function its increments
```

and j values is still 10. Because it display value of value.

```
 return 0;

}
```

Question 6:

Int fun(int ,float)

In this function is declared because datatype , name and type and no of arguments list are given

Fun(2,4.5)

In this function is called .we give value to a function and call in the body of main function.

Programming Part

```
#include<iostream>
#include<cstring>
using namespace std;
 leapyear(){      int year;
          cout<<"Enter current Year\n";
           cin>>year;
          if(year%4==0)
                    if(year%100==0)
                     if(year%400==0)
                              cout<<year<<" is leap year";
                    else
                     cout<<year<<" is not leap year";
           }else
          cout << year << " is a leap year.";
```

```
  } else
            cout << year << " is not a leap year.";
            return 0;
            }
 }
int minimum(int,int,int,int){      //Question 14
          int a , b,c,d;
          cout<<"Enter four integers value: ";
          cin>>a>>b>>c>>d;
          if(a<b)
          cout<<a<<" is minimun. \n";
          else if(c<d)
          cout<<c<<" is minimum . \n";
   else if(b<a)
          cout<<b<<" is minimum . \n";
          else if(d<c)
          cout<<d<<" is minimum . \n";
}
    transformArray(){     //Question 17
 int arr[10];
 for(int i=0;i<5;i++)
 cin>>arr[i];
 int temp=0;

  for(int i=0;i<=5/2;i++){
          int temp = arr[5-i-1];
          arr[5-i-1] = arr[i];
          arr[i]=temp;
```

```
  }
   cout<<" Array ";
   for(int i=0;i<5;i++)
   cout<<"\t"<<arr[i];
}
  strcopy(){
        char a[20]= {" Strings in C++ \n"};
        cout<<a;
        char b[30] ;
        cout<<"This is a copy of string : ";
        cout<<strcpy(b,a);
  };
```

//Question 18

Question 19:

```
  #include <iostream>
#include<conio.h>
using namespace std;

int main()
{
        int length = 0;
        char string[20];
        cout<<"Enter string :\t";
        gets(string);

        for (int i = 0 ; string [i] != '\0'; i++)
        {
                length ++;
```

```
        }
        cout<<"Lenth of string '"<<string<<"' is "<<length;
        getch ();
        return 0;
}
```

```
    ReverseOfString(){              //question 20
      char rev[10] ={"I am iman"};
        cout<<rev<<"\nReverse of This string is :  ";

        for(int i=10;i>=0; i--)
           cout<<rev[i];
         };
         boolexample(){   //Question 21
        int x;
        (x-5!=5)&&(x-5==5);
        cout<<x;   //Answer is 0.
}
```

```
#include<iostream>     //Question 22
using namespace std;
main(){
  int i;
```

```
 i = 1||0&&(4>=3+2) ;
 cout<<i;
}
```

```
 #include<iostream>
```

//Question 23

```
using namespace std;
main(){
int *p= &44;
cout<<p;
```

FIRST OF ALL A VALUE ASSIGNED TO A VARIABLE

THEN THIS VARIABLE VALUE IS STORED IN NEW VARIABLE BY ACCESSING THE ADDRESS

THEN THE NEW VARIABLE POINT TO THE OLD VARIABLE ADDRESS AND ACCESS THE VALUE THROUGH ADDRESS.

Correct code

```
 int var=44;
       p= &var;
       cout<<*p;


}
```