# Answer on Question#37989- Programming, C++

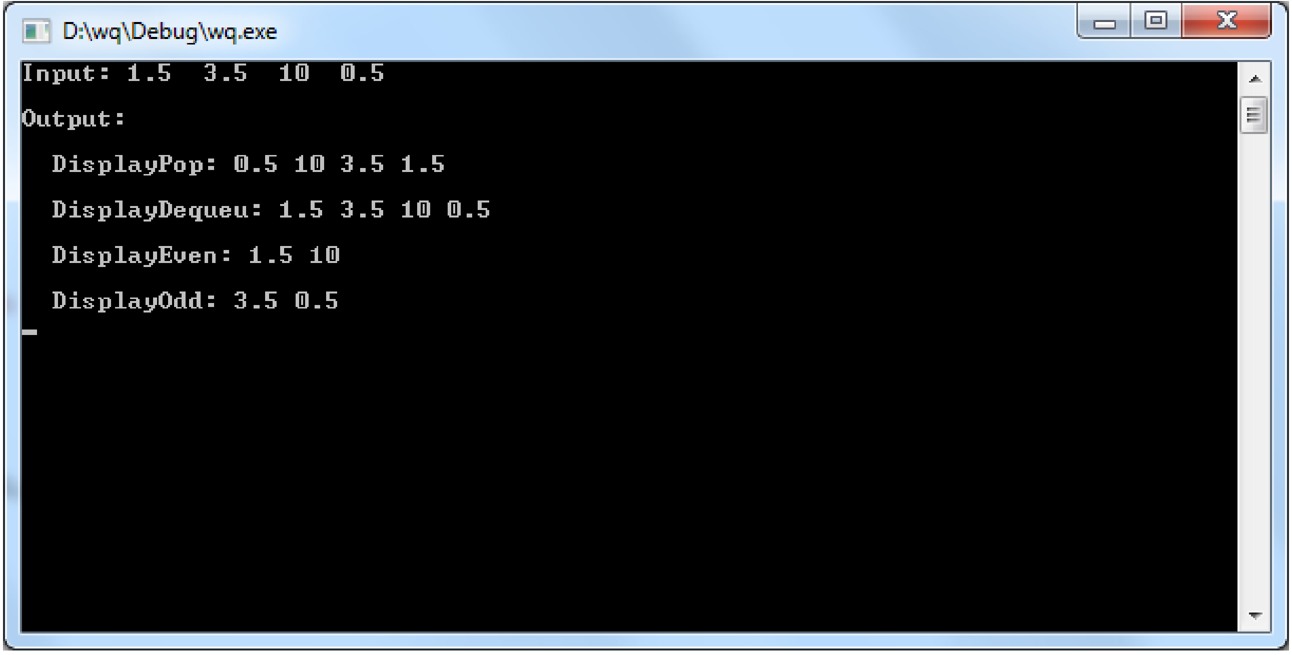
1. Doubly Linked List
   Write a C++ code that implements a doubly linked list structure that contains float elements, and then do the following:
   1. Implement the necessary member functions.
   2. Implement a function DisplayPop(…) that displays the inserted elements in a way like the pop function in the stack, Example: input= 1.5 , 3.5, 10, 0.5 output= 0.5, 10, 3.5, 1.5
   3. Implement a function DisplayDequeue(…) that displays the inserted elements in a way like the dequeue function in the queue, Example: input= 1.5 , 3.5, 10, 0.5 output= 1.5, 3.5, 10, 0.5
   4. Implement a function DisplayEven(...) that displays only the even elements inserted by the user. i.e.: element number 0, element number 2, etc.
   5. Implement a function DisplayOdd(...) that displays only the odd elements inserted by the user. i.e.: element number 1, element number 3, etc.

**Solution.**

```
#include <stdlib.h>
#include <iostream>

using namespace std;

struct Node
{
     double x;     //The value of x will be transmitted to the list
     Node *Next,*Prev; //Pointers to the next and previous addresses of list items
};

 class List   //Create a data type List
 {
     Node *Head,*Tail; //Pointers to the start address and the end of the list
 public:
     List():Head(NULL),Tail(NULL){};
     ~List();
        void DisplayPop();
        void DisplayDequeu();
        void DisplayEven();
        void DisplayOdd();

     void Add(double x);
 };

List::~List()
 {
     while (Head)
     {
         Tail=Head->Next;
         delete Head;
         Head=Tail;
     }
 }

 void List::Add(double x)
```

```
{
  Node *temp=new Node;
  temp->Next=NULL;
  temp->x=x;

  if (Head!=NULL)
  {
      temp->Prev=Tail;
      Tail->Next=temp;
      Tail=temp;
  }
  else
  {
      temp->Prev=NULL;
      Head=Tail=temp;
  }
}

void List::DisplayPop()
{
       cout<<endl<<"  DisplayPop: ";
    Node *temp=Tail;

    while (temp!=NULL)
    {
    cout<<temp->x<<" ";
    temp=temp->Prev;
    }
    cout<<"\n";
}

void List::DisplayDequeu()
{
       cout<<endl<<"  DisplayDequeu: ";
    Node *temp=Tail;

    temp=Head;
     while (temp!=NULL)
    {
    cout<<temp->x<<" ";
    temp=temp->Next;
    }
    cout<<"\n";
}

void List::DisplayEven()
{
       cout<<endl<<"  DisplayEven: ";
       Node *temp=Tail;
       int Even=0;

       temp=Head;
    while (temp!=NULL)
    {
              if(Even%2==0)
              {
          cout<<temp->x<<" ";
              }
    temp=temp->Next;
       Even++;
    }
    cout<<"\n";
}

void List::DisplayOdd()
```

```
 {
        cout<<endl<<"  DisplayOdd: ";
     Node *temp=Tail;
        int Odd=0;

        temp=Head;
     while (temp!=NULL)
     {
               if(Odd%2!=0)
               {
           cout<<temp->x<<" ";
               }
     temp=temp->Next;
        Odd++;
     }
     cout<<"\n";
 }

int main ()
{
 system("CLS");
 List lst;

 cout<<"Input: 1.5  3.5  10  0.5"<<endl<<endl;
 lst.Add(1.5);
 lst.Add(3.5);
 lst.Add(10);
 lst.Add(0.5);

 cout<<"Output: "<<endl;
 lst.DisplayPop();
 lst.DisplayDequeu();
 lst.DisplayEven();
 lst.DisplayOdd();

  system("pause");
  return 0;
}
```



```
Input: 1.5  3.5  10  0.5

Output:

  DisplayPop: 0.5 10 3.5 1.5

  DisplayDequeu: 1.5 3.5 10 0.5

  DisplayEven: 1.5 10

  DisplayOdd: 3.5 0.5
_
```