```
#include <bits/stdc++.h>
#define MAX_LENGTH 100
using namespace std;

// defining class with overloaded operators
class LARGE_NUMBER
{  private:
       int value[MAX_LENGTH];
       int length;

    public:
       LARGE_NUMBER();
       //  OVERLOADING THE cin and cout operators
       friend std::ostream &operator<<(std::ostream &os,LARGE_NUMBER &num)
       {
          for(int i=num.length-1;i>=0;i--)
          {
             os<<num.value[i];
          }
          return os;
       }
       friend istream & operator >> (istream &in,LARGE_NUMBER &num)
       {
          string str;
         int i1,j1,k1;
         cout<<"Enter a number :";
          cin>>str;
          for(i1=0;str[i1]!='\0';i1++);
         num.length=i1;
         k1=0;
         for(j1=i1-1;j1>=0;j1--)
         {
            num.value[j1]=str[k1++]-48;
         }

       }
       //  Overloading the '+' operator taking a LARGE_NUMBER as a formal parameter
       LARGE_NUMBER operator+( LARGE_NUMBER oper2 )
       {
          LARGE_NUMBER temp_number;
          int carry = 0;
          int c,i;
          if(length>oper2.length)
             c=length;
          else
             c=oper2.length;

          for ( i=0; i<c; i++ )
          {
             temp_number.value[ i ] =value[ i ] + oper2.value[ i ] + carry;
             if ( temp_number.value[ i ] > 9 )
             {
                temp_number.value[ i ] %= 10;
                carry = 1;
             }
             else
                carry = 0;
          }
          if(carry==1)
          {
             temp_number.length=c+1;
             if(temp_number.length>=MAX_LENGTH)
                cout<<"LENGTH IS BEYOND LIMIT..\n";
             else
                temp_number.value[i]=carry;
```

```
66          }
67          else
68          temp_number.length=c;
69          return temp_number;
70       }
71       // Overloading the '-' operator taking
72       LARGE_NUMBER operator-( LARGE_NUMBER oper2 )
73       {
74          LARGE_NUMBER temp_number;
75          int c;
76          if(length>oper2.length)
77             c=length;
78          else
79             c=oper2.length;
80          int borrow_value = 0;
81          if(value[c-1]>=oper2.value[c-1])
82          {
83             for( int i = 0;i <c;i++)
84             {
85                if(borrow_value==0)
86                {
87                   if(value[i]>=oper2.value[i])
88                   {
89                   temp_number.value[i]=value[i]-oper2.value[i];
90                   }
91                   else
92                   {
93                      borrow_value=1;
94                      temp_number.value[i]=value[i]+10-oper2.value[i];
95                   }
96                }
97                else
98                {
99                   if(value[i]-1>=oper2.value[i])
100                  {
101                     temp_number.value[i]=value[i]-1-oper2.value[i];
102                  }
103                  else
104                  {
105                     borrow_value=1;
106                     temp_number.value[i]=value[i]+9-oper2.value[i];
107                  }
108               }
109            }
110            temp_number.length=c;
111         return temp_number;
112         }
113         else
114         {
115            cout<<"\nFirst input number is smaller, enter again .\n"<<endl;
116         }
117
118      }
119      // Overloading the assignment operator .
120      LARGE_NUMBER operator*( LARGE_NUMBER oper2)
121      {
122         cout<<"Assignment operator"<<"\n";
123         if(this == &oper2)
124         {
125            cout<<"Address same"<<"\n";
126            return *this;
127         }
128         else{
129            int *Temporary;
130            Temporary = new int[oper2.length];
131            for(int i=0;i<oper2.length;i++)
132
```

```
            {
                Temporary[i]=oper2.value[i];
            }
            return *this;
        }
    }
};

// A defualt construct to initialize the value of each index to zero and find the length of LARGE_NUMBER

LARGE_NUMBER::LARGE_NUMBER( )
{
   for ( int i = 0; i <MAX_LENGTH; i++ )
        value[ i ] = 0;
   length = MAX_LENGTH-1;
}

int main()
{
   int ch;
   // Declaring objects of Class LARGE_NUMBER

   LARGE_NUMBER number_1,number_2,number_3,result;
   // Getting inputs ..
 while(true){

   printf("\n \n");
   printf("Enter you choice ..\n");
   printf("1.Adding Larger numbers\n");
   printf("2.Subtracting Large numbers\n");
   printf("3.Assigning one large number to another number \n");
   scanf("%d",&ch);

   if(ch==1)
   {
   cin>>number_1;
   cin>>number_2;
   result = number_1+number_2;
   cout<<number_1<<" "<< "+ "<< number_2 << " = " << result<< endl;
   }
   else if(ch==2)
   {
   cin >> number_1;
   cin >> number_2;
   result = number_1 - number_2;
   cout << number_1<< " - " << number_2 << " = "<< result<< endl;
   }
   else if(ch==3)
   {
      cin>>number_1;
      number_3=number_1;
      printf("After assigning it to another number\n");
      cout<< number_3 <<"\n";
   }
   else{
      printf("Enter A Valid Choice choice \n");
      continue;
   }

 }

}
```