```cpp
#include <iostream>
using namespace std;

//function prototype
double *resize(double *Array, int &arraysize, int newarraysize, int count);
void currentArray(double *currArray, int &arraysize, int &count);
void newhighestaddress(double *updateArray, int array_capacity, double *pmax);
void newlowestaddress(double *updateArray, int array_capacity, double *pmin);

 int main()
 {
     int count; //tracks the number of

     //size of array given by user
     int asize = 0;

     cout << "Enter size of array: ";
     cin >> asize;

     //new size of array
     int newsize = 0;
     //newsize++;

     //pointer to double variable
     double *array = nullptr;

     //allocate an array with size given by user
     array = new double[asize];

     //store in array with current size
     currentArray(array, asize, count);

     //store in array with updated size
     array = resize(array, asize, newsize, count);


     delete []array;
     array = nullptr;
     return 0;
 }

//old array
void currentArray(double *currArray, int &arraysize, int &count)
{
int i = 0; // holds number of iterations

     while(i<arraysize)
     {
        cout << "Enter a number into subscript[" << i << "] of array: " ;
        cin >> *(currArray + i);
        i++;
     }

     //display contents of array using for loop
 /*
     cout << "Numbers entered are: ";

     for(int i=0; i<arraysize; i++)
            {
            cout << *(currArray + i) << " " ;
            }
        cout << endl;

     //display the number of iteration
*/
     count = i;
```

```
 67       cout << endl;
 68       cout <<"Iteration number: " << count << endl;
 69
 70
 71  }
 72  //updated array
 73  double *resize(double *Array, int &arraysize, int newarraysize, int count)
 74  {
 75      double *p_max;
 76      double *p_min;
 77      double *NewArray = nullptr;
 78
 79      //allocate new array with updated size
 80      NewArray = new double[arraysize + newarraysize];
 81
 82          //copy the contents from fold Array to NewArray
 83          for(int i=0; i <arraysize; i++)
 84          {
 85              NewArray[i] = Array[i];
 86          }
 87
 88          //add a new number in array
 89          cout <<"Please enter element number " << count << ": ";
 90          double add_element;
 91          cin >> add_element;
 92
 93          //update new size of Array
 94          arraysize = arraysize + newarraysize;
 95          int capacity = arraysize;
 96
 97          //points to new added value
 98          cout <<"Updating p_max to address " << &p_max << ", now pointing to value: "
<< add_element << endl;
 99          cout <<"Read number: " << add_element << endl;
100
101          //deallocate memory space from old array
102          delete []Array;
103          Array = nullptr;
104
105          //add new number into new Array
106          for(int i=0; i<capacity; i++)
107              {
108               //assign new element to new array via pointer
109               *(NewArray+(capacity - 1)) = add_element;
110              }
111
112          //display updated contents of array
113          cout << "Current array elements: ";
114          for(int i=0; i<capacity; i++)
115              {
116               //display elements via pointer
117                cout << *(NewArray + i) << " " ;
118              }
119          cout << endl;
120          //display lowest element
121          newlowestaddress(NewArray,capacity,p_min);
122          cout << endl;
123          //display highest element
124          newhighestaddress(NewArray,capacity,p_max);
125
126          return NewArray;
127  }
128
129  //get lowest element in array
130  void newlowestaddress(double *updateArray, int array_capacity, double *pmin)
131  {
```

```
 double lowest = 0;


  lowest = *(updateArray); //assign subscript 0 as highest


    for(int i=1; i<array_capacity; i++)
            {
             if( *(updateArray + i) < lowest)
                lowest = *(updateArray + i);
            }
        //display lowest element and its variable address
        cout <<"Current minimum = " << lowest << " at address " << pmin ;
}

//get highest element in array
void newhighestaddress(double *updateArray, int array_capacity, double *pmax)
{

 double highest = 0;

  highest = *(updateArray); //assign subscript 0 as highest

    for(int i=1; i<array_capacity; i++)
            {
             if( *(updateArray + i) > highest)
                highest = *(updateArray + i);
            }
        //display lowest element and its variable address
        cout <<"Current maximum = " << highest << " at address " << pmax ;
}
```