|  |
| --- |
| **Data Structures and Algorithms**  ***Section*: BSCE2021 Assignment # 2 *Total marks*: 100**  ***Name*** : ***\_NIMRA MAQBOOL\_ Roll number* : \_\_BSCE21012\_\_** |

***Submission:***

• *Email instructor or TA if there are any questions. You cannot look at others’ solutions or use others’ solutions, however, you can discuss it with each other. Plagiarism will be dealt with according to the course policy.*

*• Submission after due time will not be accepted.*

**There should be a Report explaining your code and highlighting the results. Follow this naming convention for your report RollNumber\_Assignment#.pdf e.g BSCE21001\_Assignment1.pdf.**

**TASK:**

Previously, you implemented the Arraylist class which expands in size every time data is added to it and contracts in size every time data is removed from it. you implement the following function

append(int data);

prepend(int data);

insertAtIndex(int data, int index)

deleteAtEnd();

deleteAtStart();

deleteAtIndex(int index);

display();

menu();

Now implement the following functions:

findMin();

findMax();

findRepeatCount();

search(int);

findRepeatCountAndIndexes(int);

selectionSort();

The findMin function returns the minimum element of ArrayList and the findMax function returns the Maximum element of ArrayList. Search function will find the element and return its index. findRepeatCount function will count the repeated element in the list. findRepeatCountAndIndexes function will not only count the repeated element but provide its indexes. Handle all corner cases. To insert an element in an array, take user input. Make a menu-driven program. Just call the menu function in the main.

|  |
| --- |
| **Function.h:**  #include <iostream>  using namespace std;  class ArrayList { //making class private:  int sizeOfArray; //declaring  int Data; //declaring  int \*array;  int NoOfElement; public:  ArrayList(int s, int d) {  sizeOfArray = s; //copying  Data = d;  NoOfElement = 0; //copying  array = new int[sizeOfArray];  for (int i = 0; i < sizeOfArray; i++) {  array[i] = 0; //setting values to zero  }  }   void add() {  for (int i = NoOfElement; i < sizeOfArray; i++) {  cout << "ENTER DATA = ";  cin >> array[i]; //taking input  NoOfElement++;  }  cout<<"\nARRAY BEFORE ANY OPTION IS GIVEN AS = ";  }   void append() {  cout << "\nENTER ELEMENT TO INSERT = " ;  cin >> Data;  sizeOfArray = sizeOfArray + 1; //increasing size  int \*array2 = new int[sizeOfArray]; //making a new dynamic array  for (int i = 0; i < sizeOfArray; i++) {  if (i == sizeOfArray - 1) { // if iterator is 1 index less than the size then put the respective data at it  array2[i] = Data;  }//copying  else {  array2[i] = array[i]; //copy the rest of array  }  }  delete[]array; //deleting  array = array2; //copying address  array2 = NULL; //null it  cout << "\nTHE APPENDED ARRAY IS GIVEN AS = ";  }   void prepend() {  sizeOfArray = sizeOfArray + 1;  int \*array2 = new int[sizeOfArray]; //making a new dynamic array  for (int i = 0; i < sizeOfArray; i++) {  array2[i] = array[i]; //copying  }  delete[]array; //deleting  array = array2; //copying address  array2 = NULL; //null it  cout << "\nENTER ELEMENT TO INSERT = " ;  cin >> Data;  for (int i = sizeOfArray - 1; i > 0; i--) { //applying loop to copy the items to next index  array[i] = array[i - 1];  }  array[0] = Data; //putting the item to zero index  cout << "\nTHE PREPENDED ARRAY IS GIVEN AS = ";  }   void InsertAtIndex(int index) {  sizeOfArray = sizeOfArray + 1;  int \*array2 = new int[sizeOfArray]; //making a new dynamic array  for (int i = 0; i < sizeOfArray; i++) {  array2[i] = array[i]; //copying  }  delete[]array; //deleting  array = array2; //copying address  array2 = NULL; //null it  cout << "\nENTER THE INDEX = " ;  cin >> index;  cout << "\nENTER ELEMENT TO INSERT = " ;  cin >> Data;  if (index <= sizeOfArray && index >= 0) { //checking if the index is less than size and is not negative  for (int i = sizeOfArray - 1; i >= index - 1; i--) { //applying the loop from size -1 to index -1 and decrementing  array[i + 1] = array[i]; //then copying the term/element to next index  }  array[index] = Data; //putting the index to the respective index  } else {  cout << "\nYOU HAVE ENTERED INVALID INDEX." << endl;  exit(3);  }  cout << "\nTHE ARRAY AFTER INSERTING THE ELEMENT AT "<<index<<" INDEX = ";  }   void deleteAnElementFromLast() {  sizeOfArray = NoOfElement - 1;  int \*array2 = new int[sizeOfArray]; //making a new dynamic array  for (int i = 0; i < sizeOfArray; i++) {  array2[i] = array[i]; //copying  }  delete[]array; //deleting  array = array2; //copying address  array2 = NULL; //null it  cout << "\nARRAY AFTER DELETING ELEMENT FROM THE END = ";  }   void deleteAnElementFromStart() {  sizeOfArray = NoOfElement - 1;  int \*array2 = new int[sizeOfArray]; //making a new dynamic array  for (int i = 0; i < sizeOfArray; i++) {  array2[i] = array[i+1]; //copying  }  delete[]array; //deleting  array = array2; //copying  array2 = NULL; //null it  cout << "\nARRAY AFTER DELETING AN ELEMENT FROM START = ";  }   void deleteAnElementFromSpecificIndex(int index) {  int \*array2 = new int[sizeOfArray]; //making a new dynamic array  for (int i = 0; i < sizeOfArray; i++) {  array2[i] = array[i]; //copying  }  delete[]array; //deleting  array = array2; //copying  array2 = NULL; //null it  cout << "\nENTER INDEX = ";  cin >> index;  sizeOfArray = sizeOfArray - 1; //decreasing the size  if (index <= sizeOfArray && index >= 0) { //checking if the entered index is  for (int i = index; i < sizeOfArray; i++) { //applying loop  array[i] = array[i + 1]; //copying the next index to previous one  }  cout << "\nARRAY AFTER DELETING AN ELEMENT FROM THE "<<index<<" INDEX IS GIVEN AS = ";  }else{  cout<<"\nYOU HAVE ENTERED INVALID INDEX."<<endl;  cout<<"YET THE LAST INDEX IS BEING DELETED."<<endl;  }  }   void display() {  for (int i = 0; i < sizeOfArray; i++) {  cout << array[i] << " "; //displaying array elements  }  cout << endl;  }  void findMin(){  int min;  min=array[0];  for(int i=0;i<sizeOfArray;i++){  if(array[i]<min){  min=array[i];  cout<<"AT "<<i;  }  }  cout<<"INDEX THE MINIMUM NUMBER IS = "<<min<<endl;  }  void findMax(){  int max;  max=array[0];  for(int i=0;i<sizeOfArray;i++){  if(max<array[i]){  max=array[i];  }  }  cout<<"THE LARGEST NUMBER IN THE ARRAY = "<<max<<endl;   }  void findRepeatedCount() {  int count=0;  for (int i = 0; i < sizeOfArray; i++) {  for (int j = i + 1; j < sizeOfArray; j++) {  if (array[i] == array[j]) {  cout << "THE REPEATED NUMBER IS = " << array[i] << endl;  count++;  }  }  }  if(count==0){  cout<<"THERE IS NO REPEATED ELEMENT IN THE ARRAY.."<<endl;  }  }   void search(int number){  int count=0;  for(int i=0;i<sizeOfArray;i++){  if(array[i]==number){  cout<<"THE NUMBER IS AT "<<i<<" INDEX IN THE ARRAY."<<endl;  count++;  }  }  if(count==0){  cout<<"THERE IS NO SUCH ELEMENT IN ARRAY."<<endl;  }  }  void findRepeatedCountAndIndex() {  int count=0;  for (int i = 0; i < sizeOfArray; i++) {  for (int j = i + 1; j < sizeOfArray; j++) {  if (array[i] == array[j]) {  cout << "THE REPEATED NUMBER IS = " << array[i] << endl;  count++;  cout << "THE NUMBERS WERE PRESENT AT INDEXES " << i << " AND " << j << endl;  }  }  }  if(count==0){  cout<<"NO REPETITION IN ARRAY."<<endl;  }  }  void selectionSort(){  int temp;  for(int i=0;i<=sizeOfArray-1;i++){ //applying loop till the size-1 starting from 0  int min=i;  for(int j=i+1;j<sizeOfArray;j++){ //applying inner loop starting from i to greater than zero  if(array[j]<array[min]){ //comparing  min=j;  }  temp=array[min]; //swapping  array[min]=array[i];  array[i]=temp;   }  }  }   void menu() {  int opt; //declaring  cout << "CHOOSE OPTIONS." << endl;  cout << "1.APPEND." << endl;  cout << "2.PREPEND." << endl;  cout << "3.ADD AT A SPECIFIC INDEX." << endl;  cout << "4.DELETE TO THE LAST OF ARRAY." << endl; //displaying the options to choose  cout << "5.DELETE AT THE FIRST INDEX OF ARRAY." << endl;  cout << "6.DELETE TO SPECIFIC INDEX OF ARRAY." << endl;  cout<<"7.FIND THE MINIMUM ELEMENT FROM THE ARRAY."<<endl;  cout<<"8.FIND THE MAXIMUM ELEMENT FROM THE ARRAY."<<endl;  cout<<"9.FIND THE REPEATED ELEMENTS."<<endl;  cout<<"10.FIND THE NUMBER FROM ARRAY."<<endl;  cout<<"11.FIND THE INDEXES OF REPEATED ELEMENTS IN ARRAY."<<endl;  cout<<"12.SELECTION SORT."<<endl;  cout << "13.EXIT." << endl;  cin >> opt; //taking option to call  if (opt == 1) {  add();  display(); //calling functions  append();  display();  }  if (opt == 2) {  add();  display(); //calling functions  prepend();  display();  }  if (opt == 3) {  add();  display(); //calling functions  int index;  InsertAtIndex(index);  display();  }  if (opt == 4) {  add();  display(); //calling functions  deleteAnElementFromLast();  display();  }  if (opt == 5) {  add();  display(); //calling functions  deleteAnElementFromStart();  display();  }  if (opt == 6) {  add();  display();  int index;  deleteAnElementFromSpecificIndex(index); //calling functions  display();  }  if(opt==7){  add();  display();  findMin();  }  if(opt==8){  add();  display();  findMax();  }  if(opt==9){  add();  display();  findRepeatedCount();  }  if(opt==10){  add();  display();  int number;  cout<<"ENTER NUMBER TO SEARCH = ";  cin>>number;  search(number);  }  if(opt==11){  add();  display();  findRepeatedCountAndIndex();  }  if(opt==12){  add();  display();  selectionSort();  display();  }  if (opt == 13) {  cout << "YOU CHOOSE TO EXIT.." << endl; //displaying the msg  exit(3);  }  }  };   * **In the findmin function, I have declared a min named variable and then I have applied a loop starting from 0 to size and then I have applied an if condition comparing the I and min (0) index then I have displayed the min number with index.** * **In findmax function, I have just changed the comparision from min to I then displayed the max number.** * **In repeat count function, I have applied 2 loops 1 starting from zero to size-1 index and the second one starting from i+1 to size index then comparing that if element at I and j are equal then display that element else display that there is no repetition.** * **In search function, I have applied a loop starting from zero to size and then I have compared the number taken from user with every element in the array and if it matches then I have iterated count element else I have displayed that there is no such number in the array.** * **In repeat count with index, I have displayed the indexes as well as the number which is repeating.** * **In selection sort I have applied 2 loops starting from 0 to size-1 and initializing min with i and the second one from i+1 to size and then checking that if index j element is less than min element then store j in min and then check if the min is not equal to I then swap the array[i] and array[min].** * **Then I called all the functions in menu function having if statements in it.**   **Main.cpp:**  #include <iostream> #include "Functions.h" using namespace std; // int main() {  ArrayList A(5,0);  A.menu();  // Call all functions one by one here to test the output   return 0; }  **OUTPUTS:** |