|  |
| --- |
| **Data Structures and Algorithms**  ***Section*: BSCE2021 Assignment # 1 *Total marks*: 100**  ***Name*** : ***\_NIMRA MAQBOOL\_\_ Roll number* : \_BSCE21012\_\_** |

***Submission:***

• *Email instructor or TA if there are any questions. You cannot look at others’ solution or use others’ solution, 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 :**

Make an Arraylist class which contain attributes like data array, size of that array and following methods:

append(int data);

prepend(int data);

insertAtIndex(int data, int index)

deleteAtEnd();

deleteAtStart();

deleteAtIndex(int index);

display();

menu();

Append function take data from user and add that at the end of the list. Prepend function takes data from the user and adds that at the beginning of the list. deleteAtEnd function deletes the last element of the array. The array must be dynamically created. 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 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.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) {  cout << "YOU CHOOSE TO EXIT.." << endl; //displaying the msg  exit(3);  }  }  };  #endif //MAIN\_CPP\_FUNCTIONS\_H  **Append:**   * **I have taken the element from the user then I have increased the size, after that I have made a new array dynamically.** * **After applying the loop from zero to size of array having iteration of 1, I have applied the check that if iterator is 1 less than size of array then copy that element to that index other wise copy the rest of array.**   **Output:**    **PREPEND:**   * **I have taken the data from user and created array dynamically of new increased size then I have copied the array into the new array and deleted the previous one then copied its address to the other array.** * **After that I have applied loop starting from size -1 to 0 and decrementing the array and moved the elements to right and putted the element at zero index.**   **OUTPUT:**    **ADD TO A SPECIFIC INDEX:**   * **After getting the index and the element I have applied a check to check that that the entered index is not greater than size.** * **Then we put a loop starting from size -1 to index -1.** * **Then I moved the items to the right.** * **Then copied the items at the index.**   **OUTPUT:**    **DELETE AN ELEMENT FROM THE LAST:**   * **I just decreased the size of the array and then displayed it.**   **OUTPUT:**  **Text  Description automatically generated**  **DELETE AN ELEMENT FROM THE START:**   * **Reduced the size then I have moved the array from right to left of the array.**   **OUTPUT:**  **Text  Description automatically generated**  **Delete an element from the specific index:**   * **After getting the index I have applied a check to check that that the entered index is not greater than size.** * **Then applied the loop from index to size and moved the array from right to left and decreased the size.**   **OUTPUT:** |