|  |  |  |
| --- | --- | --- |
| American University of SharjahCollege of Engineering Department of Computer Science & Engineering  P. O. Box 26666  Sharjah, UAE |  | **Lab Instructor:** Eng. Sameer Alawnah  **Office:** EB2-101  **Phone**: 971-6-5152974  **e-mail**: salawnah@aus.edu  **Semester**: Fall 16 |

**CMP 220L – Introduction to Computer Science II**

**Lab 3**

**Note: The good programmer adds comments to his/her code. Add comments to your code.**

**Question 1:**

Do the following steps:

1. Create a vector of integers (vector1)
2. Push back 17 random integers to vector1
3. Print all integers in vector1
4. Erase first 7 elements.
5. Print all integers in the modified vector1.

#include<iostream>

#include<vector>

using namespace std;

void main()

{

vector<int>vector1;

for (int i = 0; i <= 17; i++)//push back 17 elements

{

vector1.push\_back(rand());

}

cout << "integers in vector 1 = " << endl;

for (int i = 0; i < vector1.size(); i++)//print

{

cout << vector1[i] << endl;

}

vector1.erase(vector1.begin()+0, vector1.begin()+7);//erase first 7 elements

cout << "integers in modified vector 1 = " << endl;

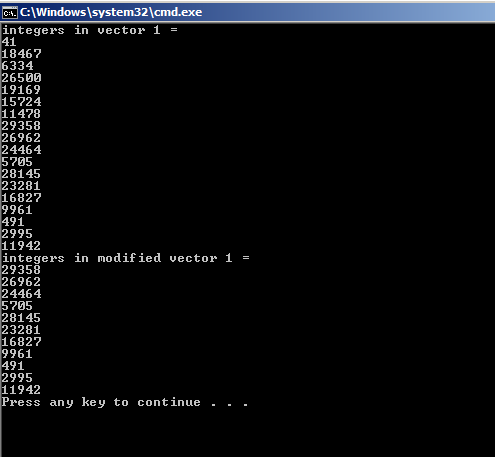
for (int i = 0; i < vector1.size(); i++)//print modified

{

cout << vector1[i] << endl;

}

}



**Question 2:**

An input file used as data source for a telephone directory program, each line represent one contact and consists of the first name, followed by the last name and finally the phone number.

Write the following functions:

* readFile that takes the file name as input and references to 3 vectors fName<string>, lName<string> int your function should read the file given by filename and fill the vectors with their values.
* searchByFirstName, that takes the fName vector and a target string to search for. Your function should return the index in which the fName contains the target string ( first occurrence only). If the target is not found your function should return -1. For example: if the target is “hma” and the fName at index 7 is “Ahmad” then your function should return 7.
* searchByLastName, will do the same functionality as searchByFirstName, but will search in lName vector.
* searchByPhone, will take the phone vector and the target as inputs and return the first occurrence of the target in the vector, -1 if the target not found.

Your main() should read the file inputs.txt save it into vectors using the readFile function. It should also continuously ask the user for the required action from the following actions list:

* 1. Search by First Name
* 2. Search by Last Name
* 3. Search by Phone
* 4. Show the last contact
* 5. Delete the last contact
* 6. Exit.

For the first 3 actions, your program should ask the user for the target and print the first name, last name and phone for the found contact, error message otherwise.

In Action 4, you should print the last found contact if exists, error message otherwise

In Action 5, you should print the last found contact if exists, error message otherwise

The last action is to exit the whole program.

Good Luck ☺

#include<iostream>

#include<vector>

#include<string>

#include<fstream>

using namespace std;

void readFile(string filename, vector<string>&fName, vector<string>&lNam, vector<int>&phone);

int searchbyFirstName(vector<string>fName, string search);

int searchbyLastName(vector<string>lName, string search);

int searchbyPhone(vector<int>phone, int search);

void main()

{

string filename;

vector<string>fName;

vector<string>lName;

vector<int>phone;

string firstname;

string lastname;

int phonenumb;

int index;

cout << "please enter file name";

cin >> filename;

readFile("input.txt", fName, lName, phone);

cout << "Search by first name:" << endl;

cin >> firstname;

index = searchbyFirstName(fName, firstname);

cout << "Search by last name:" << endl;

cin >> lastname;

index = searchbyLastName(lName, lastname);

cout << "Search by phone:" << endl;

cin >> phonenumb;

index = searchbyPhone(phone, phonenumb);

for (int i = 0; i <= index; i++)

{

if (index == i)

{

cout << fName[index] << " " << lName[index] << " " << phone[index] << endl;

}

}

cout << "Last contact is: " << fName[index] << " " << lName[index] << " " << phone[index] << endl;

fName.erase(fName.end() - 1);

lName.erase(lName.end() - 1);

phone.erase(phone.end() - 1);

exit(1);

}

void readFile(string filename, vector<string>&fName, vector<string>&lName, vector<int>&phone)

{

ifstream in(filename);

string firstname;

string lastname;

int phonenumb;

in >> firstname;

in >> lastname;

in >> phonenumb;

while (!in.eof())

{

fName.push\_back(firstname);

lName.push\_back(lastname);

phone.push\_back(phonenumb);

in >> firstname;

in >> lastname;

in >> phonenumb;

}

in.close();

}

int searchbyFirstName(vector<string>fName, string search)

{

for (int i = 0; i < fName.size(); i++)

{

if (fName[i].find(search)!=-1)

return(i);

}

return(-1);

}

int searchbyLastName(vector<string>lName, string search)

{

for (int i = 0; i < lName.size(); i++)

{

if (lName[i].find(search) != -1)

return(i);

}

return(-1);

}

int searchbyPhone(vector<int>phone, int search)

{

for (int i = 0; i < phone.size(); i++)

{

if (phone[i].find(search) != -1)

return(search);

}

return(-1);

}