

Tarea #2  
Layla Tame  
A01192934

## **SOLUCIÓN 1**

```
//  
// main.cpp  
// Tarea 2. Solución 1  
//  
// Created by Layla Tame on 1/26/19.  
// Copyright © 2019 Layla Tame. All rights reserved.  
//  
  
#include <iostream>  
#include <fstream>  
#include <chrono>  
using namespace std;  
  
//Programa que carga un archivo de texto a un arreglo y busca una  
matrícula comparando dato por dato.  
  
int main()  
{  
    int aMatr[154733], iSize = 0, iMat, iComp = 0;  
    ifstream fMatriculas;  
    fMatriculas.open("matriculas.txt");  
    bool bFound = false;  
  
    //Cargar archivo a arreglo de enteros.  
    while(!fMatriculas.eof())  
    {  
        fMatriculas >> aMatr[iSize];  
        iSize++;  
    }  
    fMatriculas.close();  
  
    cout << "Ingresa matricula: ";  
    cin >> iMat;  
  
    //Ciclo que permite buscar tantas matriculas como el usuario  
    desee  
    while (iMat != -1)  
    {  
        clock_t t1 = clock();  
        //Comparar todos los datos
```

```

        for (int i=0; i<iSize; i++)
        {
            iComp++;
            if(aMatr[i] == iMat)
            {
                bFound = true;
                break;
            }
        }
        t1 = clock() - t1;

        if (bFound == true)
        {
            cout << "Se encontró la matrícula " << iMat <<
endl;
        }
        else
        {
            cout << "No se encontró la matrícula " << iMat <<
endl;
        }

        cout << "Comparaciones: " << iComp << endl;
        cout << "Tiempo de ejecución: " << t1 << endl;

        cout << "Ingrese matrícula o para terminar ingrese -1: ";
        cin >> iMat;
        iComp = 0;
    }
}

```

## SOLUCIÓN 2

```

//
// main.cpp
// Tarea 2. Solución 2
//
// Created by Layla Tame on 1/28/19.
// Copyright © 2019 Layla Tame. All rights reserved.
//

```

```

#include <iostream>
#include <fstream>
#include <chrono>
#include <vector>
#include <algorithm>
#include <math.h>
using namespace std;

//Programa que carga un archivo de texto a un arreglo dinámico, ordena
los datos de menor a mayor y utiliza búsqueda binaria para encontrar
un dato

int iComp = 0;
clock_t t1;

//Función de búsqueda binaria del arreglo de matrices
int binSearch(vector<int> vMat, int iStart, int iFinish, int
iMatComp)
{
    int iIndex;
    if (iFinish >= iStart)
    {
        int iMiddle = (iStart + iFinish)/ 2;

        if (vMat[iMiddle] == iMatComp)
        {
            iComp++;
            iIndex = iMiddle;
        }

        else if (vMat[iMiddle] > iMatComp)
        {
            iComp++;
            iIndex = binSearch(vMat, iStart, iMiddle - 1,
iMatComp);
        }

        else if (vMat[iMiddle] < iMatComp)
        {
            iComp++;
            iIndex = binSearch(vMat, iMiddle + 1, iFinish,
iMatComp);
        }
    }
}

```

```

    else
    {
        iIndex = -1;
    }
    return iIndex;
}

int main()
{
    int iMatr, iSize=0, iMatComp;
    vector<int> vMat;
    ifstream fMatriculas;

    fMatriculas.open("matriculas.txt");
    while (!fMatriculas.eof())
    {
        fMatriculas >> iMatr;
        vMat.push_back(iMatr);
        iSize++;
    }
    fMatriculas.close();
    //Ordenar el vector en forma ascendente
    sort(vMat.begin(),vMat.end());

    cout << "Ingrese matrícula: ";
    cin >> iMatComp;

    while (iMatComp != -1)
    {
        t1 = clock();

        if (binSearch(vMat, 0, iSize-1, iMatComp) == -1)
        {
            cout << "No se encontró la matrícula " << iMatComp
<< endl;
        }
        else
        {
            cout << "Se encontró la matrícula " << iMatComp <<
endl;
        }
    }
}

```

```

        t1 = clock() - t1;

        cout << "Comparaciones: " << iComp << endl;
        cout << "Tiempo de ejecución: " << t1 << endl;

        cout << "Ingrese matrícula o para terminar ingrese -1: ";
        cin >> iMatComp;
        //Reiniciar la cuenta de comparaciones realizadas
        iComp = 0;
    }
}

```

### SOLUCIÓN 3

```

//
//  main.cpp
//  Tarea 2. Solución 3
//
//  Created by Layla Tame on 1/30/19.
//  Copyright © 2019 Layla Tame. All rights reserved.
//

#include <iostream>
#include <fstream>
#include <chrono>
#include <array>
#include <list>
using namespace std;

//Programa que carga datos de un archivo de texto a una tabla de
hashing e implementa una búsqueda sobre la misma

//Regresa los ultimos tres dígitos de la matricula
int hashFunction(int iMat)
{
    return iMat % 1000;
}

int main()
{
    ifstream fMatriculas;
    int iMat, iComp = 0, iIndex, iMatComp;
    array<list<int>, 1000> iMatr;
    bool result = false;
}

```

```

fMatriculas.open("matriculas.txt");

//Cargar archivo al arreglo
while (!fMatriculas.eof())
{
    fMatriculas >> iMat;
    iMatr[hashFunction(iMat)].push_back(iMat);
}
fMatriculas.close();

cout << "Ingrese matrícula: ";
cin >> iMatComp;

while (iMatComp != -1)
{
    iIndex = hashFunction(iMatComp);
    clock_t t1 = clock();

    //Mientras existan números en la lista
    while (!iMatr[iIndex].empty())
    {
        iComp++;
        if (iMatr[iIndex].front() == iMatComp)
        {
            result = true;
            break;
        }
        //Eliminar el elemento de la lista
        else
        {
            iMatr[iIndex].pop_front();
        }
    }

    t1 = clock() - t1;

    if (result == true)
    {
        cout << "Se encontró la matrícula " << iMatComp <<
endl;
    }
    else

```

```
        {
            cout << "No se encontró la matrícula " << iMatComp
<< endl;
        }

        cout << "Comparaciones: " << iComp << endl;
        cout << "Tiempo de ejecución: " << t1 << endl;

        cout << "Ingrese matrícula o para terminar ingrese -1: ";
        cin >> iMatComp;
        //Reiniciar la cuenta
        iComp = 0;
        result = false;
    }
}
```