

4IT Erasmus+Project 2024-1-RO01-KA121-VET-000214264
Beneficiary Colegiul National Gheorghe Vranceanu, Bacau, Romania

Sending organization: Colegiul National ”Roman-Vodă” Roman, Romania

MOBILITY PORTFOLIO
Paphos, Cyprus

Mobility period:
PAPHOS, CYPRUS, 13.04.2025-27.04.2025 – practice period 14.04.2025-25.04.2025

Receiving

Practice tutor: Lazaros Lazarou

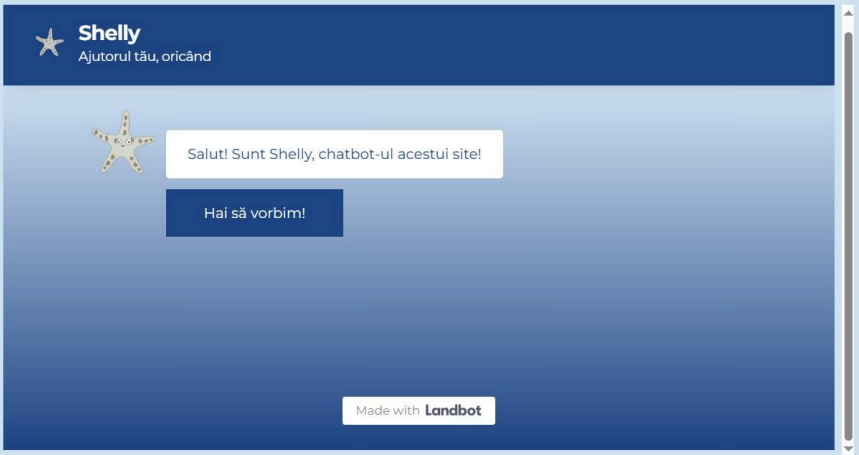
Participant: CIUȘTEA IOANA–ALEXANDRA

DISCLAIMER: The European Commission’s support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Google Sites Website - “Secțiunea Ioanei”



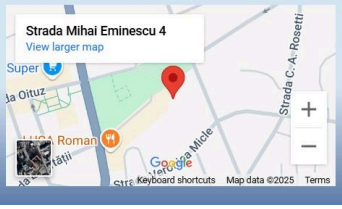




Secțiunea Ioanei
Pagina de pornire Pagina 1 Pagina 2 Galerie interactivă







Colegiul Național „Roman-Vodă”
Strada Mihai Eminescu 4
Roman 611028
Neamț, România


Secțiunea Ioanei
Pagina de pornire
Pagina 1
Pagina 2
Galerie interactivă
PAGINA 1
IOANA CIUȘTEA

Cine *este* Ioana Ciuștea?



Ciuștea Ioana-Alexandra

- Născută pe 10 decembrie 2007
- Elevă în clasa a XI-a A la Colegiul Național „Roman-Vodă”
- Voluntar în asociația People to People International
- Pasionată de informatică, lingvistică, filozofie

Pagina aceasta conține o introducere: cine sunt și de ce doresc să particip la proiectul 4IT, aspirațiile mele.

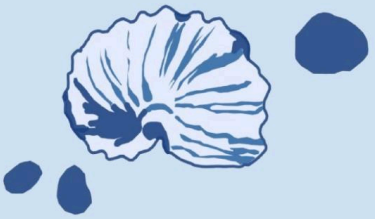
Următoarea pagină va fi dedicată obiectelor, cărților, micilor părți ale lumii care îmi aduc inspirație și fericire, care mă fac cine sunt. O numesc pagină a „odelor”. Cred că este cel mai bun mod de a prezenta într-un fel profund personalitatea și pasiunile mele.

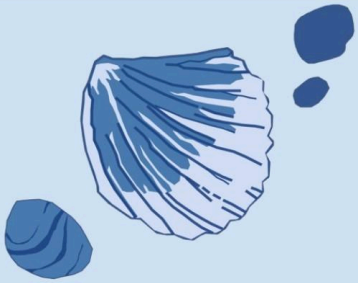
Secțiunea Ioanei

Pagina de pornire **Pagina 1** Pagina 2 Galerie interactivă

Sunt elevă la Colegiul Național „Roman-Vodă” din clasa pregătitoare. Pot spune cu toată inima că acest mediu m-a dezvoltat în moduri colosale, mi-a deschis nenumărate căi, a format o mare parte din cine sunt eu. Mi-am dezvoltat pasiunile pentru informatică, lectură și fizică în cadrul școlii.

Cu toate cele ce au fost oferite de școală, mereu am tânjit spre ceva mai mult, spre ceva mai mare. Dorința de a afla mai multe despre orice subiect nou apare de fiecare dată în mintea mea. Am început să mă afund în literatură, am descoperit filosofia. Astrofizica a devenit, de asemenea, incitantă.

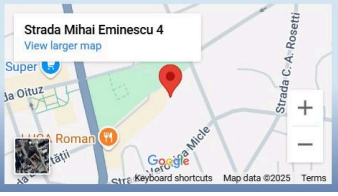




Totuși, informatica a ocupat constant un loc principal în viața mea de când am avut șansa de a o studia. Mi s-a părut mereu, și încă cred asta, deși pare a fi un clișeu, că informatica este viitorul. Vreau să cultiv cunoștințele mele în acest domeniu, mă găsesc căutând și absorbind informații despre orice limbaj, program, detaliu descoperit. Vreau să mă dezvolt necontenit, vreau să mențin pasiunea aceasta și să o extind printr-o posibilă carieră în domeniul informatic.

Tocmai din acest motiv doresc să particip la proiectul 4IT - este cea mai bună șansă pe care un elev cu aspirații ca ale mele o poate avea. Prin 4IT, pot fixa cunoștințele pe care le dețin deja în legătură cu site-urile web, pot sedimenta noi tehnici pentru a transforma ideile mele în realitate. Mi-ar demonstra că o carieră IT este într-adevăr atractivă prin activități și practică în companii, m-ar ghida spre o alegere bazată în experiențe la care am luat personal parte.



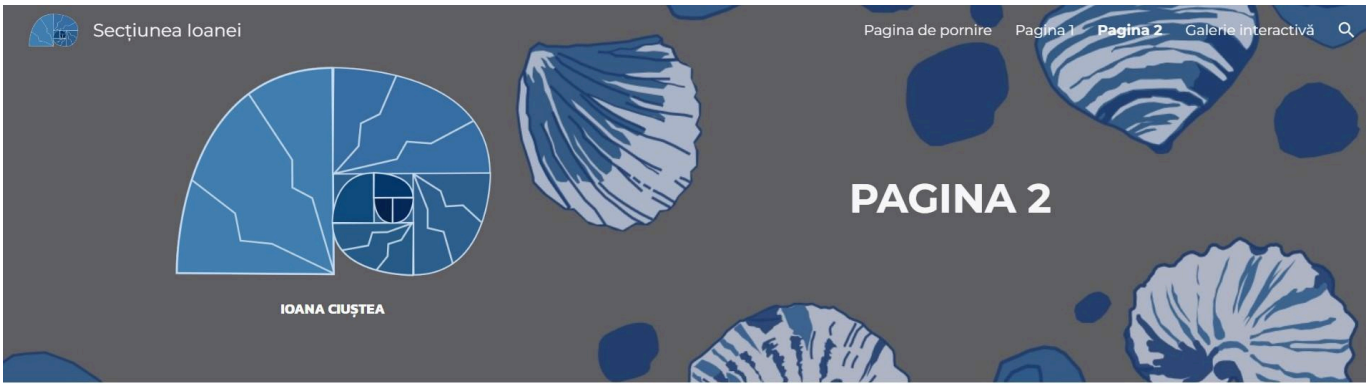


Colegiul Național „Roman-Vodă”

Strada Mihai Eminescu 4

Roman 611028

Neamț, România

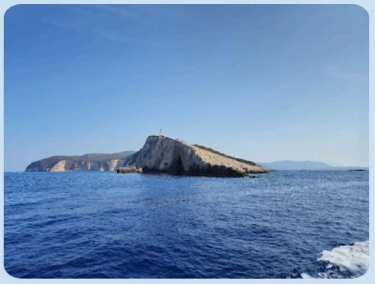

Secțiunea Ioanei
Pagina de pornire
Pagina 1
Pagina 2
Galerie interactivă
PAGINA 2
IOANA CIUȘTEA

Odă pbinfo.ro

Contul meu de pbinfo, Ciustealoana, a fost adus pe lume la 16 septembrie 2022, ora 18:08. Și pot spune cu mare mândrie că la momentul actual, am peste 1200 de probleme rezolvate pe el.

O mare și excelentă sursă de glume în cercul de prieteni, contul meu pbinfo a devenit, pentru mine, o parte din cine sunt. Consider că reflectă perfect pasiunea mea pentru informatică și perseverența mea. Ocupă chiar locul 33 în „Top 100" din județul Neamț. Momentan...

Sunt o mare susținătoare a rezolvatului de probleme pe pbinfo, cel puțin susțin că pe mine mă ajută foarte mult. Cantitatea de dopamină și serotonină secretată de creierul meu încrețit de la problemele de pe acest site atunci când reușesc să obțin 100 de puncte la o problemă este excepțională.



Secțiunea Ioanei
Pagina de pornire Pagina 1 **Pagina 2** Galerie interactivă

Odă CSS



De când am început să învăț HTML, în clasa a opta, CSS mi s-a revoluționar.

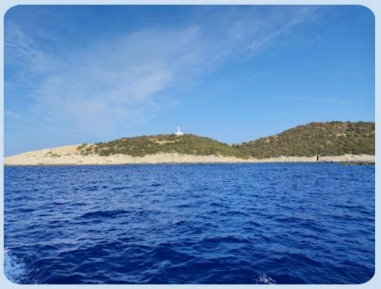
Simplul, sau nu foarte simplul, fapt că folosind CSS pot face ceva atât de complex ca o pagină web frumoasă și plăcută estetic mi se pare o mare putere. Dacă știi să îl folosești, dacă ai câțiva așI în mânecă, CSS deschide nenumărate uși spre transpunerea ideilor complexe direct pe site-ul pe care îl creezi. Sunt fascinată de tot ceea ce poți face cu HTML și CSS și mereu nerăbdătoare să descopăr lucruri noi.

Sunt atât de multe trucuri pe care le poți folosi și cred cu tărie că dacă te documentezi, pentru că există foarte multe surse informative, CSS este un „game-changer".

Odă cărții „Străinul" de Albert Camus

Odă cărții „Străinul” de Albert Camus

Am înfulecat „Străinul” în două ore, într-o zi de vară când era mult prea cald afară pentru a ieși din casă. Poate și din motivul acesta al vremii, am rezonat cu ea în moduri în care nu am mai reușit să o fac cu alte cărți. A fost prima carte cu care am intrat în contact scrisă de Albert Camus și m-a impresionat cel mai mult dintre toate cărțile pe care le-am citit până astăzi.



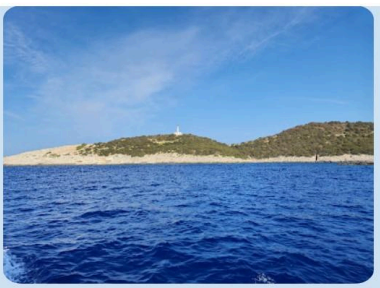
Scena care declanșează acțiunea m-a marcat, pentru că de fiecare dată când mă întorc la ea o pot recepta mai mult și mai mult prin propriile simțuri. O simt în țesuturile mele, simt și eu cum îi intră personajului principal soarele care se reflectă de pe revolver în ochi, cum se încinge sângele în ei. Simt raza de lumină alb-portocalie care îmi obturează vederea.

Cred cu tărie că modul în care Camus compune este genial și majoritatea scenelor și pasajelor sunt extrem de bine scrise. Nu mă consider, momentan, adeptă a „absurdului”, sau a oricărei alte ideologii, dar sunt un mare admirator al lucrărilor lui Camus.

Secțiunea Ioanei

Pagina de pornire Pagina 1 **Pagina 2** Galerie interactivă

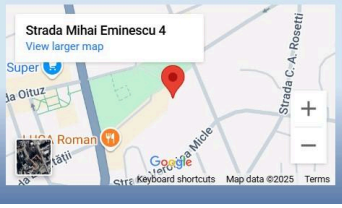
afară pentru a ieși din casă. Poate și din motivul acesta al vremii, am rezonat cu ea în moduri în care nu am mai reușit să o fac cu alte cărți. A fost prima carte cu care am intrat în contact scrisă de Albert Camus și m-a impresionat cel mai mult dintre toate cărțile pe care le-am citit până astăzi.



Scena care declanșează acțiunea m-a marcat, pentru că de fiecare dată când mă întorc la ea o pot recepta mai mult și mai mult prin propriile simțuri. O simt în țesuturile mele, simt și eu cum îi intră personajului principal soarele care se reflectă de pe revolver în ochi, cum se încinge sângele în ei. Simt raza de lumină alb-portocalie care îmi obturează vederea.

Cred cu tărie că modul în care Camus compune este genial și majoritatea scenelor și pasajelor sunt extrem de bine scrise. Nu mă consider, momentan, adeptă a „absurdului”, sau a oricărei alte ideologii, dar sunt un mare admirator al lucrărilor lui Camus.





Colegiul Național „Roman-Vodă”

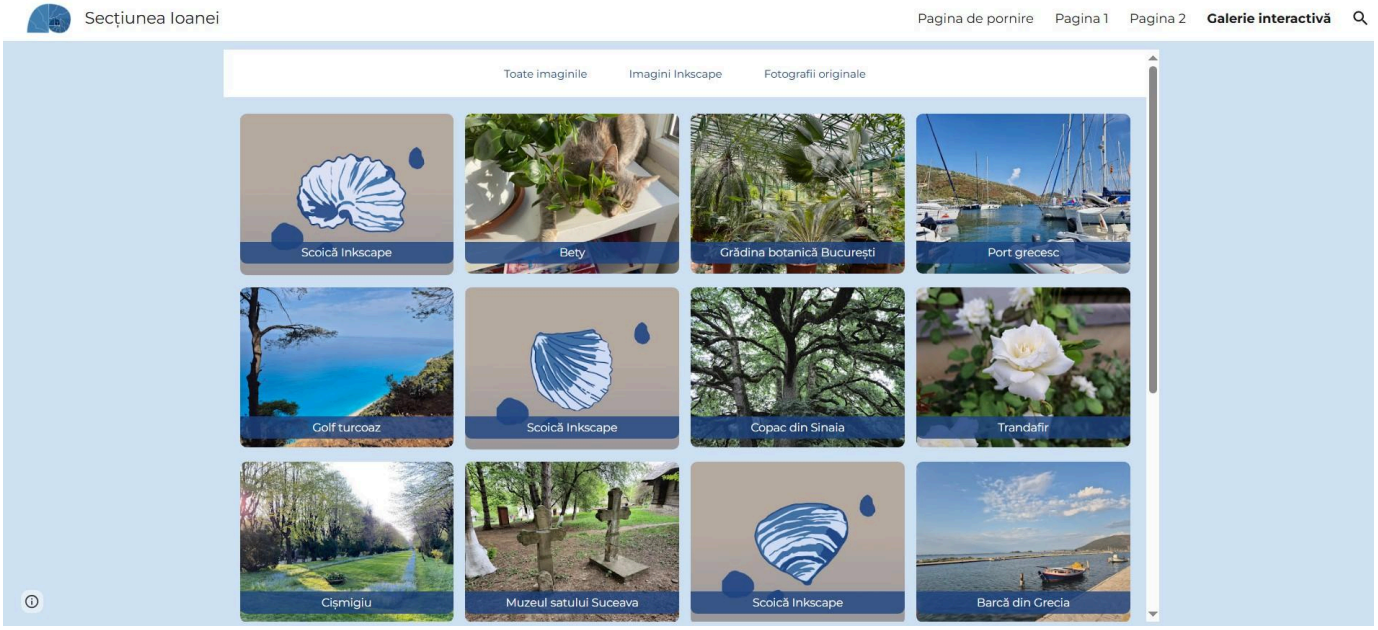
Strada Mihai Eminescu 4

Roman 611028

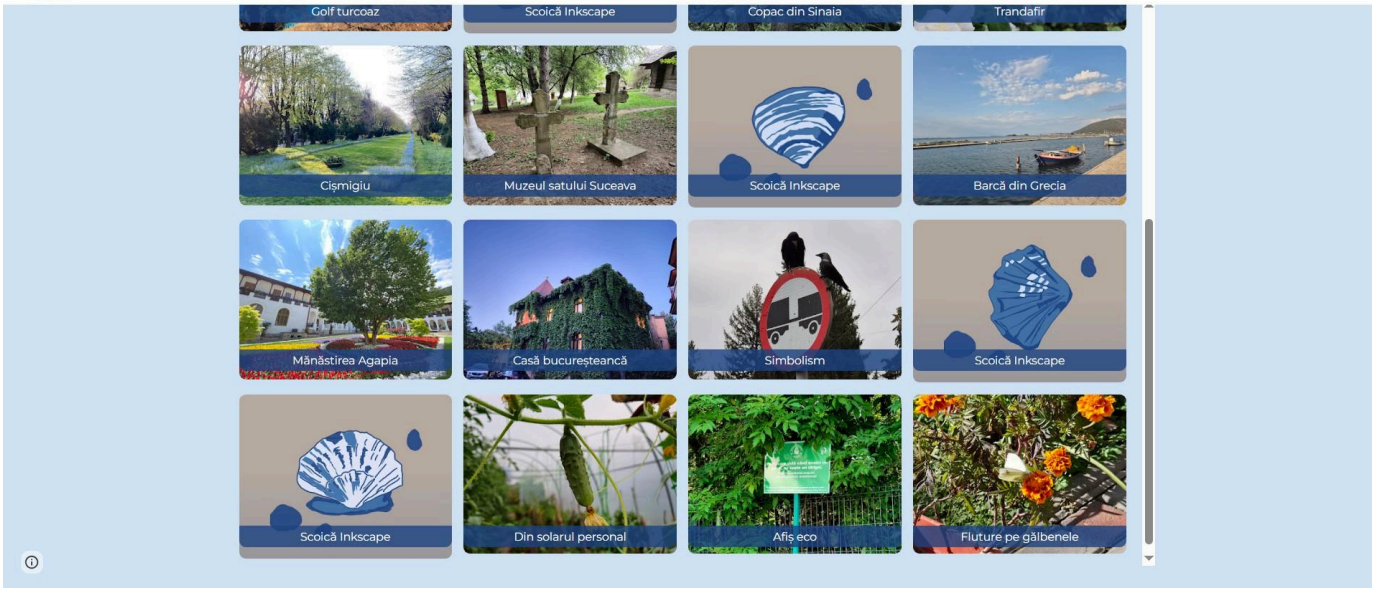
Neamț, România

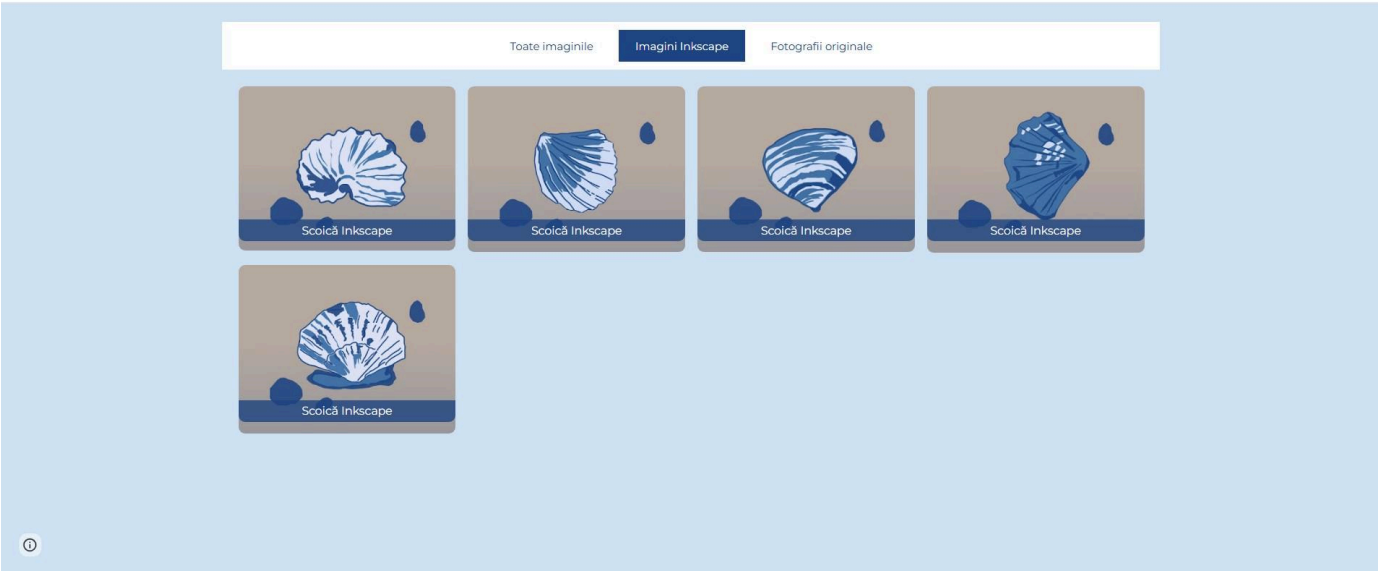


PAGINA PRINCIPALĂ PAGINA 1 PAGINA 2 GALERIE INTERACTIVĂ

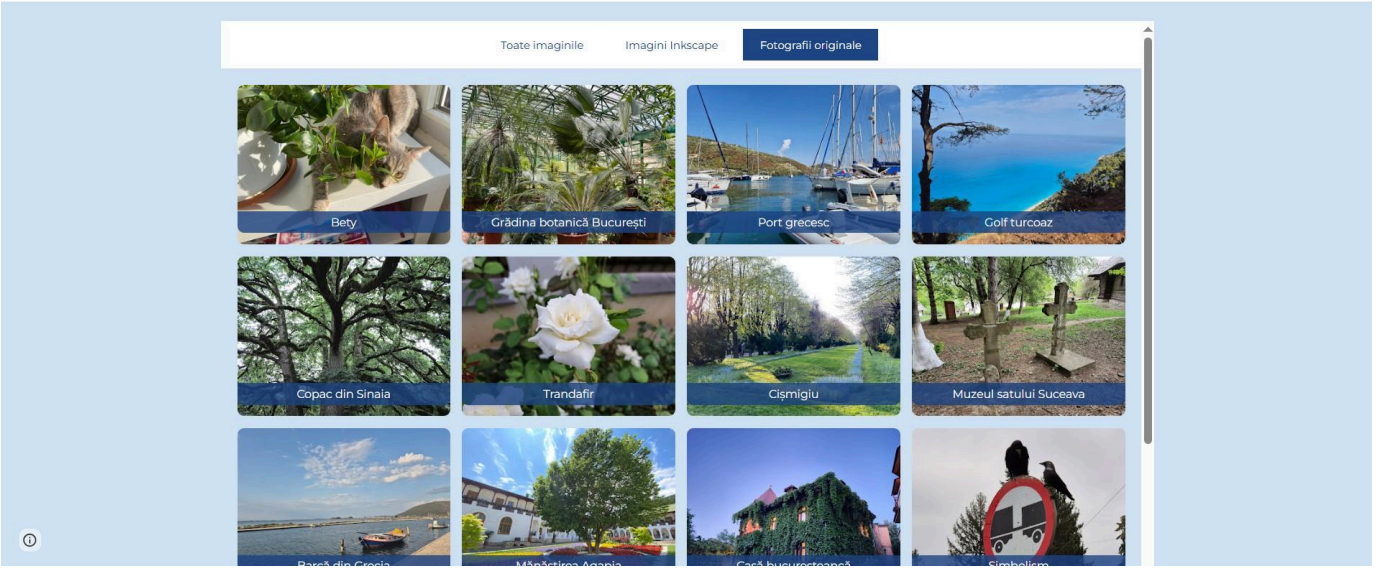


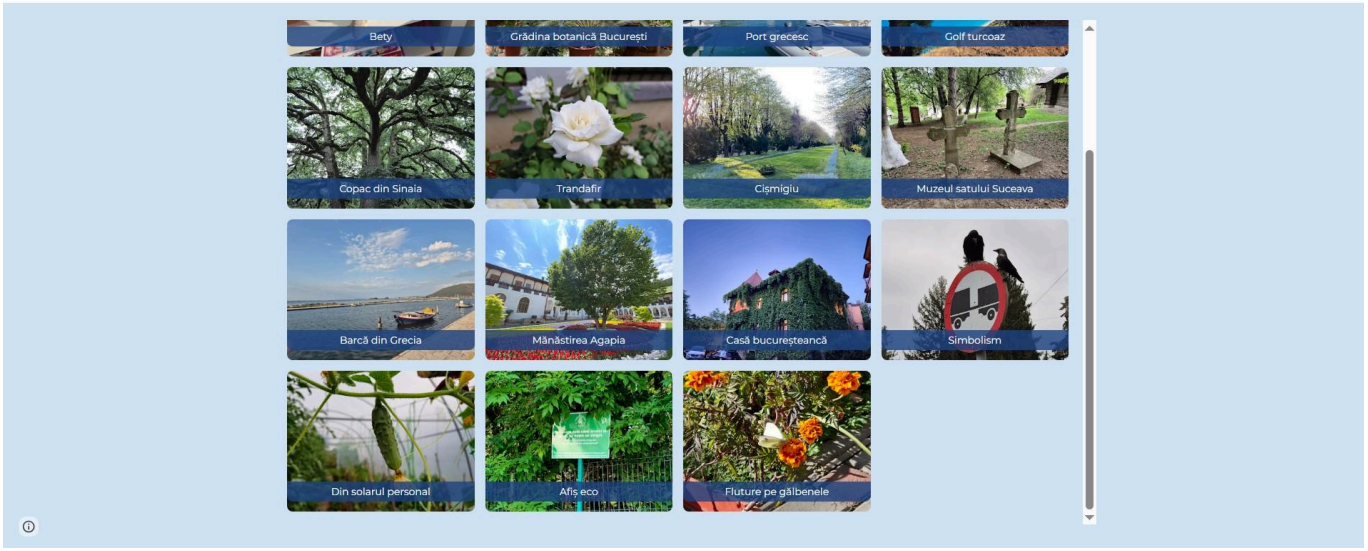
Secțiunea Ioanei
Pagina de pornire Pagina 1 Pagina 2 **Galerie interactivă**





Secțiunea Ioanei
Pagina de pornire Pagina 1 Pagina 2 **Galerie interactivă**



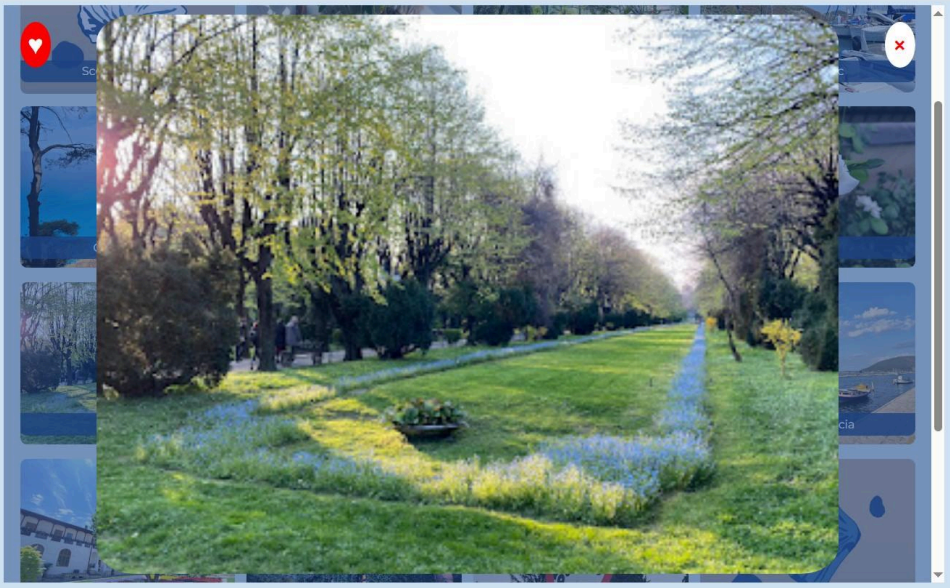


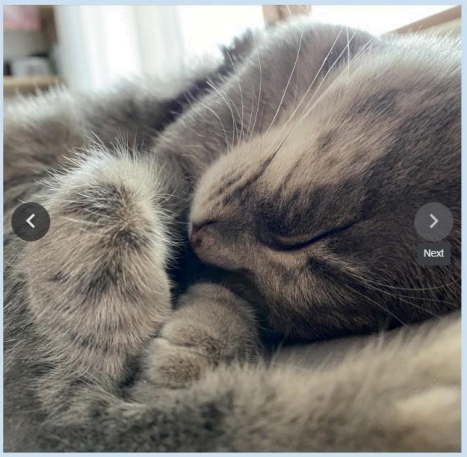
Secțiunea Ioanei
Pagina de pornire Pagina 1 Pagina 2 **Galerie interactivă**



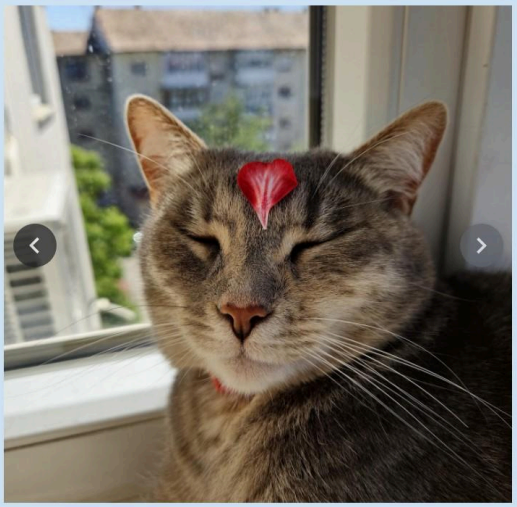


Secțiunea Ioanei
Pagina de pornire Pagina 1 Pagina 2 **Galerie interactivă**



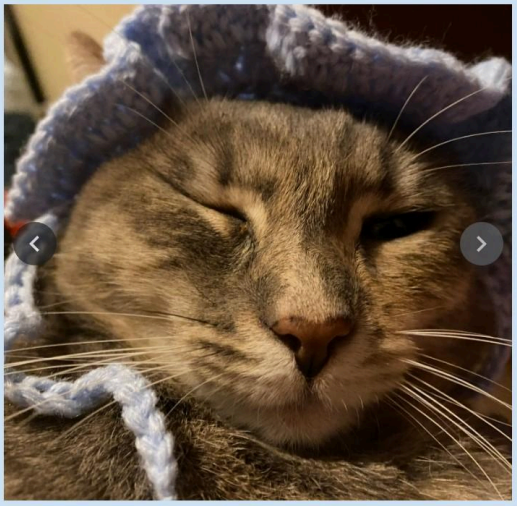


Secțiunea Ioanei
Pagina de pornire Pagina 1 Pagina 2 **Galerie interactivă**



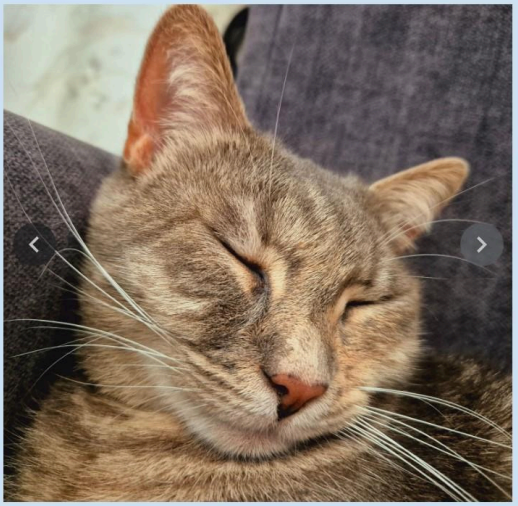
Secțiunea Ioanei

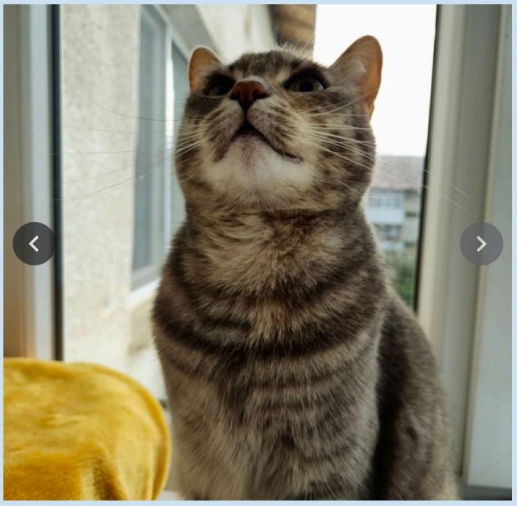
Pagina de pornire Pagina 1 Pagina 2 **Galerie interactivă**



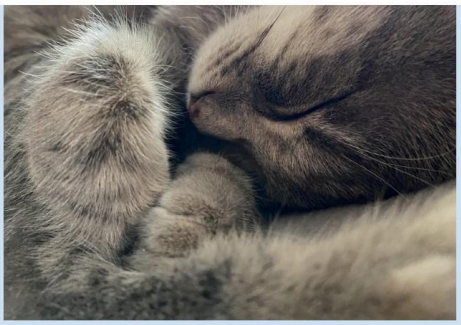
Secțiunea Ioanei

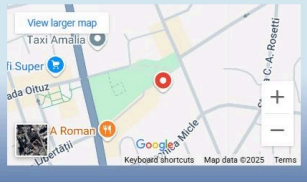
Pagina de pornire Pagina 1 Pagina 2 **Galerie interactivă**





Secțiunea Ioanei
Pagina de pornire Pagina 1 Pagina 2 **Galerie interactivă**





Colegiul Național „Roman-Vodă”
Strada Mihai Eminescu 4
Roman 611028
Neamț, România

“Cum să-ți găsești un loc de muncă”





Platforme de recrutare

Ciuștea Ioana-Alexandra



Platforme de recrutare

Oferta de platforme de recrutare a crescut considerabil în ultimii 10 ani. Fiecare platformă are specificul ei:

1. *Platforme individuale de recrutare*
2. *eJobs, BestJobs* = site-uri de volum
3. *LinkedIn, Facebook* = platforme sociale
4. *MediJobs, DevJobs* = platforme specializate
5. *Indeed, Jooble* = agregatoare de joburi


Platforme de recrutare
Cum folosesc platformele de recrutare?
• Crearea unui profil
◦ Poză de profil
◦ Contact
◦ Studii, skill-uri relevante
• Adăugarea CV-ului
• Networking
• Aplică pentru un loc de muncă!
31

Produse realizate în cadrul mobilității

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

int main()
{
   //1. pointers - manipulating memory - basics
   int age; //integer
   char b; //character
   int *aPtr; //pointer, stores address of an integer
   char *bPtr; //stores address of a character
   age = 5;
   aPtr = &age; //& = address of; takes the address of age and saves it in aPtr - now aPtr knows where age is located
   cout << age << "\n"; //5
   cout << aPtr << "\n"; //0x75AB34
   cout << *aPtr << "\n"; //5; * = dereferencing the pointer
   for(int i = 1; i <= 10; i++)
   {
      cout << aPtr << "\n";
      aPtr++; //increments, adds 4 bytes every time
   }
   //2. dynamic array
   int *dynamicArray; //pointer to an integer, going to be the dynamic array = pointer which can manipulate integers
   int arrSize = 0; //good practice to set to 0, because we are working with memory
   cout << "Enter size of the array: ";
   cin >> arrSize;
   dynamicArray = new int[arrSize]; //dynamically allocate memory for 50 integers
   for(int i = 0; i < arrSize; i++) //fill the array with 50 integers
      dynamicArray[i] = i + 1;
   for(int i = 0; i < arrSize; i++) //display the contents of the array
      cout << dynamicArray[i] << " - ";
   cout << "\n";
   delete dynamicArray; //release the dynamically allocated space back to the OS
}
```

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

int main()
{
   int arrSize = 0, n = 0, nTemp = 0, x;
   char *a, *temp;
   cout << "How many numbers will you enter? ";
   cin >> arrSize;
   a = new char[arrSize];
   do
   {
      cout << "Enter number " << n << " -> ";
      cin >> x;
      if(x < 65) cout << "Number is too small.\n";
      else if(x > 90) cout << "Number is too big.\n";
          else
          {
             a[n] = (char)x;
            n++;
          }
      if(n >= arrSize)
      {
         cout << "Out of space! Would you like to enter more? Enter 0 to exit. ";
         cin >> nTemp;
         if(nTemp > 0)
         {
            arrSize = arrSize + nTemp;
            temp = new char[arrSize];
            for(int i = 0; i < n; i++)
               temp[i] = a[i];
            delete [] a;
            a = temp;
         }
      }
   } while(n < arrSize);
   cout << "Now printing the resulting characters: \n";
   for(int i = 0; i < arrSize; i++)
      cout << a[i] << "\n";
   return 0;
}
```

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

int main()
{
   int arrSize = 0, n = 0, nTemp = 0, x;
   char *a, *temp; //a and temp are each a pointer to an integer
   cout << "How many numbers will you enter? "; //ask the used to input the wanted size of the
array
   cin >> arrSize; //read the size of the array
   a = new char[arrSize]; //dynamically create an array with the number the user gave
   do
   {
      cout << "Enter number " << n << " -> ";
      cin >> x; //user enters the numbers
      if(x < 65) cout << "Number is too small.\n"; //if the number is less than 65, don't save
      else if(x > 90) cout << "Number is too big.\n"; //if the number is more than 90, don't save
         else
         {
            a[n] = (char)x;
            cout << "Number saved!\n"; //save the number if between 65 and 90
            n++; //increase the counter after saving the number
         }
      if(n >= arrSize) //we need to make the dynamic array bigger if the counter is greater than
arrSize
      {
         cout << "Out of space! How many more numbers would you like to enter? Enter 0 to exit. ";
//let the user know he is out of space
         cin >> nTemp; //reads how many numbers the user wants to add to the array
         if(nTemp > 0) //if nTemp is 0, exit; if nTemp is bigger, increase the size of the array
         {
            arrSize = arrSize + nTemp; //increase the previous size
            temp = new char[arrSize]; //create the new array
            for(int i = 0; i < n; i++) //copy values to new array
               temp[i] = a[i];
            delete [] a; //deletes the array from memory
            a = temp; //now points to a new array
         }
      }
   } while(n < arrSize); //while the user wants to input numbers
   cout << "Now printing the resulting characters: \n";
   for(int i = 0; i < arrSize; i++) //print the letters in the dynamic array
      cout << a[i] << "\n";
   delete a;
   return 0;
}
```

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

//command line arguments - very important

int main(int argc, char *argv[]) //argc = argument count;
{
   cout << "You have entered " << argc << " arguments:\n";
   //Using a while loop to iterate through arguments
   int i = 1;
   while(i < argc)
   {
      cout << "Argument " << i + 1 << ": " << argv[i] << "\n";
      i++;
      // /h /s /t
      if(strcmp(argv[i], "/h") == 0)
      {
         //write code to show help
      }
      if(strcmp(argv[i], "/s"))
      {
         //write code to implement /s option
      }
      if(strcmp(argv[i], "/t"))
      {
         //write code to implement /t option
      }
   }
   return 0;
}
```

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

//collect information about students - first name and telephone number; ask the user to search for a name and we will give back the phone number
// !to solve a big problem, break it down into smaller pieces
/*
convert this program in order to be able to not only read information but also search for a name and a phone number
*/

void search_name(string *name, string *number, int arrSize, string n)
{
   bool ok = false;
   for(int i = 0; i < arrSize && !ok; i++)
      if(name[i] == n)
      {
         ok = true;
         cout << n << "'s number is: " << number[i] << "\n";
      }
   if(!ok) cout << "Couldn't find a student named " << n << ".\n";
}

void search_number(string *name, string *number, int arrSize, string n)
{
   bool ok = false;
   for(int i = 0; i < arrSize && !ok; i++)
      if(number[i] == n)
      {
         ok = true;
         cout << "The student whose phone number is " << n << " is named " << name[i] << ".\n";
      }
   if(!ok) cout << "Couldn't find a student with the phone number " << n << ".\n";
}

int main()
{
   int arrSize = 0;
   string *name, n; //pointers to a string
   string *number; //pointers to a string
   cout << "How many students' information would you like to collect? "; //ask the user to input the number of students
   cin >> arrSize; //read the number of students
   name = new string[arrSize]; //dynamically create an array to store the names
   number = new string[arrSize]; //dynamically create an array to store the numbers
   for(int i = 0; i < arrSize; i++)
   {
```

```
        cout << "Enter the name of student " << i << " -> ";
        cin >> name[i]; //name of student number n
        cout << "Enter " << name[i] << "'s phone number -> ";
        cin >> number[i]; //number of student number n
        cout << "Information saved!\n";
    }
    cout << "Now printing the information you saved:\n";
    for(int i = 0; i < arrSize; i++) //print the information saved
        cout << name[i] << " " << number[i] << "\n";
    cout << "Whose number are you looking for? Enter 0 to exit. ";
    cin >> n; //user inputs the name we need to search
    while(n != "0") //while the user wants to keep searching
    {
        search_name(name, number, arrSize, n);
        cout << "Whose number are you looking for? Enter 0 to exit. ";
        cin >> n; //read another name to continue searching
    }
    cout << "Whose name are you looking for? Enter 0 to exit. ";
    cin >> n; //user inputs the number we need to search
    while(n != "0") //while the user wants to keep searching
    {
        search_number(name, number, arrSize, n);
        cout << "Whose name are you looking for? Enter 0 to exit. ";
        cin >> n; //read another number to continue searching
    }
    delete [] name;
    delete [] number;
    return 0;
}
```

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

//classes
class student
{
   string name;
   int age;
   public:
      void setAge(int a)
      {
         age = a;
      }

};

int main()
{
   student oneStudent; //an object student which is named oneStudent; (oneStudent.age = 10 DOESN'T work)
   oneStudent.setAge(10);
   return 0;
}
```

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

class car
{
   //private:
      string brand;
      string model;
      string plateNumber;
      string name;
   public:
      car()
      {
         brand = "EMPTY";
         model = "EMPTY";
         plateNumber = "EMPTY";
         name = "EMPTY";
      }
      void setBrand(string n) {brand = n;}
      void setModel(string n) {model = n;}
      void setPlateNumber(string n) {plateNumber = n;}
      void setName(string n) {name = n;}
      string getBrand() {return brand;}
      string getModel() {return model;}
      string getPlateNumber() {return plateNumber;}
      string getName() {return name;}
};

int main()
{
   const int MAXCARS = 3;
   car myCars[MAXCARS];
   string tBrand = "", tModel = "", tPlateNumber = "", tName = "", temp = "";
   for(int i = 0; i < MAXCARS; i++)
   {
      cout << "Brand " << i << ": ";
      cin >> tBrand;
      myCars[i].setBrand(tBrand);
      cout << "Model " << i << ": ";
      cin >> tModel;
      myCars[i].setModel(tModel);
      cout << "Plate number " << i << ": ";
      cin >> tPlateNumber >> temp;
      tPlateNumber = tPlateNumber + " " + temp;
      myCars[i].setPlateNumber(tPlateNumber);
      cout << "Name " << i << ": ";
      cin >> tName;
```

```
        myCars[i].setName(tName);
    }
    cout << "\n\n\nYOU HAVE SAVED THE FOLLOWING INFORMATION\n";
    for(int i = 0; i < MAXCARS; i++)
        cout << myCars[i].getBrand() << " " << myCars[i].getModel() << " " <<
myCars[i].getPlateNumber() << " " << myCars[i].getName() << "\n";
    return 0;
}
```

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

class product
{
   private:
      string brand;
      string type;
      float price;
      int stock;
      string barcode;
   public:
      product()
      {
         brand = "EMPTY";
         type = "EMPTY";
         price = -1.1;
         stock = -1;
         barcode = "EMPTY";
      }
      void setBrand(string n) {brand = n;}
      void setType(string n) {type = n;}
      void setPrice(float n) {price = n;}
      void setStock(int n) {stock = n;}
      void setBarcode(string n) {barcode = n;}
      string getBrand() {return brand;}
      string getType() {return type;}
      float getPrice() {return price;}
      int getStock() {return stock;}
      string getBarcode() {return barcode;}
};

void searchName(product supermarket[], const int MAXPROD, string n)
{
   bool ok = false;
   for(int i = 0; i < MAXPROD && !ok; i++)
   {
      if(supermarket[i].getBrand() == n)
      {
         ok = true;
         cout << "The product you were looking for:\n";
         cout << "Brand: " << supermarket[i].getBrand() << "\n";
         cout << "Type: " << supermarket[i].getType() << "\n";
         cout << "Price: " << supermarket[i].getPrice() << "\n";
         cout << "Stock: " << supermarket[i].getStock() << "\n";
         cout << "Barcode: " << supermarket[i].getBarcode() << "\n";
      }
```

```
    }
    if(!ok) cout << "Couldn't find any products with that brand.\n";
}

void searchType(product supermarket[], const int MAXPROD, string n)
{
    bool ok = false;
    for(int i = 0; i < MAXPROD && !ok; i++)
    {
        if(supermarket[i].getType() == n)
        {
            ok = true;
            cout << "The product you were looking for:\n";
            cout << "Brand: " << supermarket[i].getBrand() << "\n";
            cout << "Type: " << supermarket[i].getType() << "\n";
            cout << "Price: " << supermarket[i].getPrice() << "\n";
            cout << "Stock: " << supermarket[i].getStock() << "\n";
            cout << "Barcode: " << supermarket[i].getBarcode() << "\n";
        }
    }
    if(!ok) cout << "Couldn't find any products of that type.\n";
}

int main()
{
    const int MAXPROD = 2;
    product supermarket[MAXPROD];
    string tBrand = "", tType = "", tBarcode = "";
    float tPrice = 0.0;
    int tStock = 0;
    for(int i = 0; i < MAXPROD; i++)
    {
        cout << "Brand " << i << ": ";
        cin >> tBrand;
        supermarket[i].setBrand(tBrand);
        cout << "Type " << i << ": ";
        cin >> tType;
        supermarket[i].setType(tType);
        cout << "Price " << i << ": ";
        cin >> tPrice;
        supermarket[i].setPrice(tPrice);
        cout << "Stock " << i << ": ";
        cin >> tStock;
        supermarket[i].setStock(tStock);
        cout << "Barcode " << i << ": ";
        cin >> tBarcode;
        supermarket[i].setBarcode(tBarcode);
```

```
    }
    cout << "\n\n\nNow printing the products you added:\n";
    for(int i = 0; i < MAXPROD; i++)
    {
        cout << "Brand: " << supermarket[i].getBrand() << "\n";
        cout << "Type: " << supermarket[i].getType() << "\n";
        cout << "Price: " << supermarket[i].getPrice() << "\n";
        cout << "Stock: " << supermarket[i].getStock() << "\n";
        cout << "Barcode: " << supermarket[i].getBarcode() << "\n";
        cout << "---------------------------------------\n";
    }
    while(tType != "0")
    {
        cout << "Please enter the type: (0 to exit) ";
        cin >> tType;
        searchType(supermarket, MAXPROD, tType);
    }
    while(tBrand != "0")
    {
        cout << "Please enter the brand: (0 to exit) ";
        cin >> tBrand;
        searchName(supermarket, MAXPROD, tBrand);
    }
    return 0;
}
```

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

class car
{
   //private:
      string brand;
      string model;
      string plateNumber;
      string name;
   public:
      car()
      {
         brand = "EMPTY";
         model = "EMPTY";
         plateNumber = "EMPTY";
         name = "EMPTY";
      }
      void setBrand(string n) {brand = n;}
      void setModel(string n) {model = n;}
      void setPlateNumber(string n) {plateNumber = n;}
      void setName(string n) {name = n;}
      string getBrand() {return brand;}
      string getModel() {return model;}
      string getPlateNumber() {return plateNumber;}
      string getName() {return name;}
};

int main()
{
   const int MAXCARS = 3;
   car myCars[MAXCARS];
   string tBrand = "", tModel = "", tPlateNumber = "", tName = "", temp = "";
   for(int i = 0; i < MAXCARS; i++)
   {
      cout << "Brand " << i << ": ";
      cin >> tBrand;
      myCars[i].setBrand(tBrand);
      cout << "Model " << i << ": ";
      cin >> tModel;
      myCars[i].setModel(tModel);
      cout << "Plate number " << i << ": ";
      cin >> tPlateNumber >> temp;
      tPlateNumber = tPlateNumber + " " + temp;
      myCars[i].setPlateNumber(tPlateNumber);
      cout << "Name " << i << ": ";
      cin >> tName;
```

```
        myCars[i].setName(tName);
    }
    cout << "\n\n\nYOU HAVE SAVED THE FOLLOWING INFORMATION\n";
    for(int i = 0; i < MAXCARS; i++)
        cout << myCars[i].getBrand() << " " << myCars[i].getModel() << " " <<
myCars[i].getPlateNumber() << " " << myCars[i].getName() << "\n";
    return 0;
}
```

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

class car
{
   //private:
      string brand;
      string model;
      string plateNumber;
      string name;
   public:
      car()
      {
         brand = "EMPTY";
         model = "EMPTY";
         plateNumber = "EMPTY";
         name = "EMPTY";
      }
      void setBrand(string n) {brand = n;}
      void setModel(string n) {model = n;}
      void setPlateNumber(string n) {plateNumber = n;}
      void setName(string n) {name = n;}
      string getBrand() {return brand;}
      string getModel() {return model;}
      string getPlateNumber() {return plateNumber;}
      string getName() {return name;}
};

void readCar(car *&myCars, int &MAXCARS)
{
   car *tempa;
   int n = 0, nTemp = 0;
   cout << "How many cars would you like to add? ";
   cin >> MAXCARS;
   myCars = new car[MAXCARS];
   string tBrand = "", tModel = "", tPlateNumber = "", tName = "", temp = "";
   do
   {
      cout << "Brand " << n + 1 << ": ";
      cin >> tBrand;
      myCars[n].setBrand(tBrand);
      cout << "Model " << n + 1 << ": ";
      cin >> tModel;
      myCars[n].setModel(tModel);
      cout << "Plate number " << n + 1 << ": ";
      cin >> tPlateNumber >> temp;
      tPlateNumber = tPlateNumber + " " + temp;
```

```
        myCars[n].setPlateNumber(tPlateNumber);
        cout << "Name " << n + 1 << ": ";
        cin >> tName;
        myCars[n].setName(tName);
        n++;
        if(n >= MAXCARS)
        {
           cout << "Out of space! How many more cars would you like to enter? (0 to exit) ";
           cin >> nTemp;
           if(nTemp > 0)
           {
             MAXCARS += nTemp;
             tempa = new car[MAXCARS];
             for(int i = 0; i < n; i++)
              tempa[i] = myCars[i];
             delete [] myCars;
             myCars = tempa;
           }
        }

    }while(n < MAXCARS);
    cout << "\n\n\n";
}

void printCar(car *myCars, int MAXCARS)
{
    cout << "\n\n\nYOU HAVE SAVED THE FOLLOWING INFORMATION\n";
    for(int i = 0; i < MAXCARS; i++)
    {
        cout << "---------- Car " << i + 1 << "----------\n";
        cout << "Brand: " << myCars[i].getBrand() << "\n";
        cout << "Model: " << myCars[i].getModel() << "\n";
        cout << "Plate number: " << myCars[i].getPlateNumber() << "\n";
        cout << "Name: " << myCars[i].getName() << "\n";
    }
    cout << "\n\n\n";
}

void searchBrand(car *myCars, int MAXCARS, string n)
{
    bool ok = false;
    for(int i = 0; i < MAXCARS && !ok; i++)
    {
        if(myCars[i].getBrand() == n)
        {
           ok = true;
           cout << "The car you are searching for is:\n";
```

```
            cout << "Brand: " << myCars[i].getBrand() << "\n";
            cout << "Model: " << myCars[i].getModel() << "\n";
            cout << "Plate number: " << myCars[i].getPlateNumber() << "\n";
            cout << "Name: " << myCars[i].getName() << "\n";
        }
    }
    if(!ok) cout << "\n\n\nCouldn't find the car you are looking for.\n";
    cout << "\n\n\n";
}

void searchModel(car *myCars, int MAXCARS, string n)
{
    bool ok = false;
    for(int i = 0; i < MAXCARS && !ok; i++)
    {
        if(myCars[i].getModel() == n)
        {
            ok = true;
            cout << "\n\n\nThe car you are searching for is:\n";
            cout << "Brand: " << myCars[i].getBrand() << "\n";
            cout << "Model: " << myCars[i].getModel() << "\n";
            cout << "Plate number: " << myCars[i].getPlateNumber() << "\n";
            cout << "Name: " << myCars[i].getName() << "\n";
        }
    }
    if(!ok) cout << "\n\n\nCouldn't find the car you are looking for.\n";
    cout << "\n\n\n";
}

void searchName(car *myCars, int MAXCARS, string n)
{
    bool ok = false;
    for(int i = 0; i < MAXCARS && !ok; i++)
    {
        if(myCars[i].getName() == n)
        {
            ok = true;
            cout << "\n\n\nThe car you are searching for is:\n";
            cout << "Brand: " << myCars[i].getBrand() << "\n";
            cout << "Model: " << myCars[i].getModel() << "\n";
            cout << "Plate number: " << myCars[i].getPlateNumber() << "\n";
            cout << "Name: " << myCars[i].getName() << "\n";
        }
    }
    if(!ok) cout << "\n\n\nCouldn't find the car you are looking for.\n";
    cout << "\n\n\n";
}
```

```
void searchPlateNumber(car *myCars, int MAXCARS, string n)
{
   bool ok = false;
   for(int i = 0; i < MAXCARS && !ok; i++)
   {
      if(myCars[i].getPlateNumber() == n)
      {
         ok = true;
         cout << "\n\n\nThe car you are searching for is:\n";
         cout << "Brand: " << myCars[i].getBrand() << "\n";
         cout << "Model: " << myCars[i].getModel() << "\n";
         cout << "Plate number: " << myCars[i].getPlateNumber() << "\n";
         cout << "Name: " << myCars[i].getName() << "\n";
      }
   }
   if(!ok) cout << "\n\n\nCouldn't find the car you are looking for.\n";
   cout << "\n\n\n";
}

int main()
{
   int MAXCARS = 0, option = 1000;
   string temp = "", temp1 = "", temp2 = "";
   car *myCars;
   while(option > 0)
   {
      cout << "What would you like to do?\n1. Enter data\n2. Display data\n3. Search by brand\n4.
Search by model\n5. Search by owner\n6. Search by plate number\n0. Exit\n";
      cin >> option;
      cout << "\n\n\n";
      switch(option)
      {
         case 1:
            readCar(myCars, MAXCARS);
            break;
         case 2:
            printCar(myCars, MAXCARS);
            break;
         case 3:
            cout << "Please enter the brand: ";
            cin >> temp;
            searchBrand(myCars, MAXCARS, temp);
            break;
         case 4:
            cout << "Please enter the model: ";
            cin >> temp;
```

```
            searchModel(myCars, MAXCARS, temp);
            break;
        case 5:
            cout << "Please enter the owner: ";
            cin >> temp;
            searchName(myCars, MAXCARS, temp);
            break;
        case 6:
            cout << "Please enter the plate number: ";
            cin >> temp1 >> temp2;
            temp = temp1 + " " + temp2;
            searchBrand(myCars, MAXCARS, temp);
            break;
      }
   }
   return 0;
}
```

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

class product
{
   private:
      string brand;
      string type;
      float price;
      int stock;
      string barcode;
   public:
      product()
      {
         brand = "EMPTY";
         type = "EMPTY";
         price = -1.1;
         stock = -1;
         barcode = "EMPTY";
      }
      void setBrand(string n) {brand = n;}
      void setType(string n) {type = n;}
      void setPrice(float n) {price = n;}
      void setStock(int n) {stock = n;}
      void setBarcode(string n) {barcode = n;}
      string getBrand() {return brand;}
      string getType() {return type;}
      float getPrice() {return price;}
      int getStock() {return stock;}
      string getBarcode() {return barcode;}
};

void searchName(product supermarket[], const int MAXPROD, string n)
{
   bool ok = false;
   for(int i = 0; i < MAXPROD && !ok; i++)
   {
      if(supermarket[i].getBrand() == n)
      {
         ok = true;
         cout << "The product you were looking for:\n";
         cout << "Brand: " << supermarket[i].getBrand() << "\n";
         cout << "Type: " << supermarket[i].getType() << "\n";
         cout << "Price: " << supermarket[i].getPrice() << "\n";
         cout << "Stock: " << supermarket[i].getStock() << "\n";
         cout << "Barcode: " << supermarket[i].getBarcode() << "\n";
      }
```

```
    }
    if(!ok) cout << "Couldn't find any products with that brand.\n";
}

void searchType(product supermarket[], const int MAXPROD, string n)
{
    bool ok = false;
    for(int i = 0; i < MAXPROD && !ok; i++)
    {
        if(supermarket[i].getType() == n)
        {
            ok = true;
            cout << "The product you were looking for:\n";
            cout << "Brand: " << supermarket[i].getBrand() << "\n";
            cout << "Type: " << supermarket[i].getType() << "\n";
            cout << "Price: " << supermarket[i].getPrice() << "\n";
            cout << "Stock: " << supermarket[i].getStock() << "\n";
            cout << "Barcode: " << supermarket[i].getBarcode() << "\n";
        }
    }
    if(!ok) cout << "Couldn't find any products of that type.\n";
}

void scanBarcode(product supermarket[], const int MAXPROD, string n)
{
    bool ok = false;
    for(int i = 0; i < MAXPROD && !ok; i++)
    {
        if(supermarket[i].getBarcode() == n)
        {
            ok = true;
            cout << "Brand: " << supermarket[i].getBrand() << "\n";
            cout << "Type: " << supermarket[i].getType() << "\n";
            cout << "Price: " << supermarket[i].getPrice() << "\n";
            supermarket[i].setStock(supermarket[i].getStock() - 1);
            cout << "Stock: " << supermarket[i].getStock() + 1 << " -> " << supermarket[i].getStock() <<
"\n";
        }
    }
    if(!ok) cout << "Couldn't find the product you scanned.\n";
}

int main()
{
    const int MAXPROD = 2;
    product supermarket[MAXPROD];
    string tBrand = "", tType = "", tBarcode = "";
```

```
    float tPrice = 0.0;
    int tStock = 0;
    for(int i = 0; i < MAXPROD; i++)
    {
        cout << "Brand " << i << ": ";
        cin >> tBrand;
        supermarket[i].setBrand(tBrand);
        cout << "Type " << i << ": ";
        cin >> tType;
        supermarket[i].setType(tType);
        cout << "Price " << i << ": ";
        cin >> tPrice;
        supermarket[i].setPrice(tPrice);
        cout << "Stock " << i << ": ";
        cin >> tStock;
        supermarket[i].setStock(tStock);
        cout << "Barcode " << i << ": ";
        cin >> tBarcode;
        supermarket[i].setBarcode(tBarcode);
    }
    cout << "\n\n\nNow printing the products you added:\n";
    for(int i = 0; i < MAXPROD; i++)
    {
        cout << "Brand: " << supermarket[i].getBrand() << "\n";
        cout << "Type: " << supermarket[i].getType() << "\n";
        cout << "Price: " << supermarket[i].getPrice() << "\n";
        cout << "Stock: " << supermarket[i].getStock() << "\n";
        cout << "Barcode: " << supermarket[i].getBarcode() << "\n";
        cout << "---------------------------------------\n";
    }
    int choice = 1;
    while(choice > 0)
    {
        cout << "\n\n\nWould you like to search by brand or by type? (1 for brand, 2 for type, 0 to
exit) ";
        cin >> choice;
        switch(choice)
        {
            case 1:
                cout << "Please enter the brand: ";
                cin >> tBrand;
                searchName(supermarket, MAXPROD, tBrand);
                break;
            case 2:
                cout << "Please enter the type: ";
                cin >> tType;
                searchType(supermarket, MAXPROD, tType);
```

```
            break;
        }
    }
    string bar;
    while(bar != "0")
    {
        cout << "\n\n\nPlease scan a product: (0 to exit) ";
        cin >> bar;
        if(bar != "0") scanBarcode(supermarket, MAXPROD, bar);
    }
    return 0;
}
```

```cpp
#include <iostream>

using namespace std;

//write a program where you will pass a pointer to a function; a pointer pointing to an array; inside the function you will do a search

void searchPtr(int *arr, int arrSize, int x)
{
   bool ok = false;
   for(int i = 0; i < arrSize && !ok; i++)
      if(arr[i] == x)
      {
         ok = true;
         cout << "Found " << x << " inside of the array.\n";
      }
   if(!ok) cout << "Couldn't find " << x << " inside of the array.\n";
}

int main()
{
   int *arr, arrSize = 0, x = 0;
   cout << "How many numbers would you like to enter? ";
   cin >> arrSize;
   arr = new int[arrSize];
   for(int i = 0; i < arrSize; i++)
      cin >> arr[i];
   cout << "What number do you want to search for? ";
   cin >> x;
   searchPtr(arr, arrSize, x);
   delete [] arr;
   return 0;
}
```

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

class car
{
   //private:
      string brand;
      string model;
      string plateNumber;
      string name;
   public:
      car()
      {
         brand = "EMPTY";
         model = "EMPTY";
         plateNumber = "EMPTY";
         name = "EMPTY";
      }
      void setBrand(string n) {brand = n;}
      void setModel(string n) {model = n;}
      void setPlateNumber(string n) {plateNumber = n;}
      void setName(string n) {name = n;}
      string getBrand() {return brand;}
      string getModel() {return model;}
      string getPlateNumber() {return plateNumber;}
      string getName() {return name;}
};

car* readCar(car *myCars, int &MAXCARS)
{
   car *tempa;
   int n = MAXCARS, nTemp = 0, x;
   cout << "How many cars would you like to add? ";
   cin >> x;
   MAXCARS += x;
   myCars = new car[MAXCARS];
   string tBrand = "", tModel = "", tPlateNumber = "", tName = "", temp = "";
   do
   {
      cout << "Brand " << n + 1 << ": ";
      cin >> tBrand;
      myCars[n].setBrand(tBrand);
      cout << "Model " << n + 1 << ": ";
      cin >> tModel;
      myCars[n].setModel(tModel);
      cout << "Plate number " << n + 1 << ": ";
      cin >> tPlateNumber >> temp;
```

```
        tPlateNumber = tPlateNumber + " " + temp;
        myCars[n].setPlateNumber(tPlateNumber);
        cout << "Name " << n + 1 << ": ";
        cin >> tName;
        myCars[n].setName(tName);
        n++;
        if(n >= MAXCARS)
        {
           cout << "Out of space! How many more cars would you like to enter? (0 to exit) ";
           cin >> nTemp;
           if(nTemp > 0)
           {
              MAXCARS += nTemp;
              tempa = new car[MAXCARS];
              for(int i = 0; i < n; i++)
               tempa[i] = myCars[i];
              delete [] myCars;
              myCars = tempa;
           }
        }

    }while(n < MAXCARS);
    return myCars;
}

void printCar(car *myCars, int MAXCARS)
{
    cout << "\n\n\nYOU HAVE SAVED THE FOLLOWING INFORMATION\n";
    for(int i = 0; i < MAXCARS; i++)
    {
       cout << "---------- Car " << i + 1 << "----------\n";
       cout << "Brand: " << myCars[i].getBrand() << "\n";
       cout << "Model: " << myCars[i].getModel() << "\n";
       cout << "Plate number: " << myCars[i].getPlateNumber() << "\n";
       cout << "Name: " << myCars[i].getName() << "\n";
    }
    cout << "\n\n\n";
}

void searchBrand(car *myCars, int MAXCARS, string n)
{
    bool ok = false;
    for(int i = 0; i < MAXCARS && !ok; i++)
    {
       if(myCars[i].getBrand() == n)
       {
          ok = true;
```

```
            cout << "The car you are searching for is:\n";
            cout << "Brand: " << myCars[i].getBrand() << "\n";
            cout << "Model: " << myCars[i].getModel() << "\n";
            cout << "Plate number: " << myCars[i].getPlateNumber() << "\n";
            cout << "Name: " << myCars[i].getName() << "\n";
        }
    }
    if(!ok) cout << "\n\n\nCouldn't find the car you are looking for.\n";
    cout << "\n\n\n";
}

void searchModel(car *myCars, int MAXCARS, string n)
{
    bool ok = false;
    for(int i = 0; i < MAXCARS && !ok; i++)
    {
        if(myCars[i].getModel() == n)
        {
            ok = true;
            cout << "\n\n\nThe car you are searching for is:\n";
            cout << "Brand: " << myCars[i].getBrand() << "\n";
            cout << "Model: " << myCars[i].getModel() << "\n";
            cout << "Plate number: " << myCars[i].getPlateNumber() << "\n";
            cout << "Name: " << myCars[i].getName() << "\n";
        }
    }
    if(!ok) cout << "\n\n\nCouldn't find the car you are looking for.\n";
    cout << "\n\n\n";
}

void searchName(car *myCars, int MAXCARS, string n)
{
    bool ok = false;
    for(int i = 0; i < MAXCARS && !ok; i++)
    {
        if(myCars[i].getName() == n)
        {
            ok = true;
            cout << "\n\n\nThe car you are searching for is:\n";
            cout << "Brand: " << myCars[i].getBrand() << "\n";
            cout << "Model: " << myCars[i].getModel() << "\n";
            cout << "Plate number: " << myCars[i].getPlateNumber() << "\n";
            cout << "Name: " << myCars[i].getName() << "\n";
        }
    }
    if(!ok) cout << "\n\n\nCouldn't find the car you are looking for.\n";
    cout << "\n\n\n";
```

```
}

void searchPlateNumber(car *myCars, int MAXCARS, string n)
{
   bool ok = false;
   for(int i = 0; i < MAXCARS && !ok; i++)
   {
      if(myCars[i].getPlateNumber() == n)
      {
         ok = true;
         cout << "\n\n\nThe car you are searching for is:\n";
         cout << "Brand: " << myCars[i].getBrand() << "\n";
         cout << "Model: " << myCars[i].getModel() << "\n";
         cout << "Plate number: " << myCars[i].getPlateNumber() << "\n";
         cout << "Name: " << myCars[i].getName() << "\n";
      }
   }
   if(!ok) cout << "\n\n\nCouldn't find the car you are looking for.\n";
   cout << "\n\n\n";
}

int main()
{
   int MAXCARS = 0, option = 1000;
   string temp = "", temp1 = "", temp2 = "";
   car *myCars;
   while(option > 0)
   {
      cout << "What would you like to do?\n1. Enter data\n2. Display data\n3. Search by brand\n4.
Search by model\n5. Search by owner\n6. Search by plate number\n0. Exit\n";
      cin >> option;
      cout << "\n\n\n";
      switch(option)
      {
         case 1:
            myCars = readCar(myCars, MAXCARS);
            break;
         case 2:
            printCar(myCars, MAXCARS);
            break;
         case 3:
            cout << "Please enter the brand: ";
            cin >> temp;
            searchBrand(myCars, MAXCARS, temp);
            break;
         case 4:
            cout << "Please enter the model: ";
```

```
            cin >> temp;
            searchModel(myCars, MAXCARS, temp);
            break;
        case 5:
            cout << "Please enter the owner: ";
            cin >> temp;
            searchName(myCars, MAXCARS, temp);
            break;
        case 6:
            cout << "Please enter the plate number: ";
            cin >> temp1 >> temp2;
            temp = temp1 + " " + temp2;
            searchBrand(myCars, MAXCARS, temp);
            break;
      }
   }
   return 0;
}
```

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

class product
{
   private:
      string brand;
      string type;
      float price;
      int stock;
      string barcode;
   public:
      product()
      {
         brand = "EMPTY";
         type = "EMPTY";
         price = -1.1;
         stock = -1;
         barcode = "EMPTY";
      }
      void setBrand(string n) {brand = n;}
      void setType(string n) {type = n;}
      void setPrice(float n) {price = n;}
      void setStock(int n) {stock = n;}
      void setBarcode(string n) {barcode = n;}
      string getBrand() {return brand;}
      string getType() {return type;}
      float getPrice() {return price;}
      int getStock() {return stock;}
      string getBarcode() {return barcode;}
};

product* readMarket(product *supermarket, int &MAXPROD)
{
   int x, n = MAXPROD, nTemp = 0;
   product *temp;
   string tBrand = "", tType = "", tBarcode = "";
   float tPrice = 0.0;
   int tStock = 0;
   cout << "How many products would you like to enter? ";
   cin >> x;
   MAXPROD += x;
   supermarket = new product[MAXPROD];
   do
   {
      cout << "Brand " << n + 1 << ": ";
      cin >> tBrand;
```

```
        supermarket[n].setBrand(tBrand);
        cout << "Type " << n + 1 << ": ";
        cin >> tType;
        supermarket[n].setType(tType);
        cout << "Price " << n + 1 << ": ";
        cin >> tPrice;
        supermarket[n].setPrice(tPrice);
        cout << "Stock " << n + 1 << ": ";
        cin >> tStock;
        supermarket[n].setStock(tStock);
        cout << "Barcode " << n + 1 << ": ";
        cin >> tBarcode;
        supermarket[n].setBarcode(tBarcode);
        n++;
        if(n >= MAXPROD)
        {
           cout << "Out of space! How many more products would you like to enter? (0 to exit) ";
           cin >> nTemp;
           if(nTemp > 0)
           {
              MAXPROD += nTemp;
              temp = new product[MAXPROD];
              for(int i = 0; i < n; i++)
                 temp[i] = supermarket[i];
              delete [] supermarket;
              supermarket = temp;
           }

        }
    } while(n < MAXPROD);
    return supermarket;
}

void printMarket(product *supermarket, int MAXPROD)
{
    cout << "\n\n\nNow printing the products you added:\n";
    for(int i = 0; i < MAXPROD; i++)
    {
        cout << "Brand: " << supermarket[i].getBrand() << "\n";
        cout << "Type: " << supermarket[i].getType() << "\n";
        cout << "Price: " << supermarket[i].getPrice() << "\n";
        cout << "Stock: " << supermarket[i].getStock() << "\n";
        cout << "Barcode: " << supermarket[i].getBarcode() << "\n";
        cout << "---------------------------------------\n";
    }
}
```

```
void searchBrand(product *supermarket, int MAXPROD, string n)
{
   bool ok = false;
   for(int i = 0; i < MAXPROD && !ok; i++)
   {
      if(supermarket[i].getBrand() == n)
      {
         ok = true;
         cout << "The product you were looking for:\n";
         cout << "Brand: " << supermarket[i].getBrand() << "\n";
         cout << "Type: " << supermarket[i].getType() << "\n";
         cout << "Price: " << supermarket[i].getPrice() << "\n";
         cout << "Stock: " << supermarket[i].getStock() << "\n";
         cout << "Barcode: " << supermarket[i].getBarcode() << "\n";
      }
   }
   if(!ok) cout << "Couldn't find any products with that brand.\n";
}

void searchType(product *supermarket, int MAXPROD, string n)
{
   bool ok = false;
   for(int i = 0; i < MAXPROD && !ok; i++)
   {
      if(supermarket[i].getType() == n)
      {
         ok = true;
         cout << "The product you were looking for:\n";
         cout << "Brand: " << supermarket[i].getBrand() << "\n";
         cout << "Type: " << supermarket[i].getType() << "\n";
         cout << "Price: " << supermarket[i].getPrice() << "\n";
         cout << "Stock: " << supermarket[i].getStock() << "\n";
         cout << "Barcode: " << supermarket[i].getBarcode() << "\n";
      }
   }
   if(!ok) cout << "Couldn't find any products of that type.\n";
}

void searchPrice(product *supermarket, int MAXPROD, float n)
{
   bool ok = false;
   for(int i = 0; i < MAXPROD && !ok; i++)
   {
      if(supermarket[i].getPrice() == n)
      {
         ok = true;
         cout << "The product you were looking for:\n";
```

```
            cout << "Brand: " << supermarket[i].getBrand() << "\n";
            cout << "Type: " << supermarket[i].getType() << "\n";
            cout << "Price: " << supermarket[i].getPrice() << "\n";
            cout << "Stock: " << supermarket[i].getStock() << "\n";
            cout << "Barcode: " << supermarket[i].getBarcode() << "\n";
        }
    }
    if(!ok) cout << "Couldn't find any products with that price.\n";
}

void searchStock(product *supermarket, int MAXPROD, int n)
{
    bool ok = false;
    for(int i = 0; i < MAXPROD && !ok; i++)
    {
        if(supermarket[i].getStock() == n)
        {
            ok = true;
            cout << "The product you were looking for:\n";
            cout << "Brand: " << supermarket[i].getBrand() << "\n";
            cout << "Type: " << supermarket[i].getType() << "\n";
            cout << "Price: " << supermarket[i].getPrice() << "\n";
            cout << "Stock: " << supermarket[i].getStock() << "\n";
            cout << "Barcode: " << supermarket[i].getBarcode() << "\n";
        }
    }
    if(!ok) cout << "Couldn't find any products with that stock.\n";
}

void scanBarcode(product *supermarket, int MAXPROD, string n)
{
    bool ok = false;
    for(int i = 0; i < MAXPROD && !ok; i++)
    {
        if(supermarket[i].getBarcode() == n)
        {
            ok = true;
            cout << "Brand: " << supermarket[i].getBrand() << "\n";
            cout << "Type: " << supermarket[i].getType() << "\n";
            cout << "Price: " << supermarket[i].getPrice() << "\n";
            supermarket[i].setStock(supermarket[i].getStock() - 1);
            cout << "Stock: " << supermarket[i].getStock() + 1 << " -> " << supermarket[i].getStock() <<
"\n";
        }
    }
    if(!ok) cout << "Couldn't find the product you scanned.\n";
}
```

```
int main()
{
   int MAXPROD = 0;
   product *supermarket;
   string temp = "";
   float tempfloat = 0.0;
   int tempint = 0;
   int choice = 1;
   while(choice > 0)
   {
      cout << "What would you like to do?\n1. Enter data\n2. Print data\n3. Search by brand\n4.
Search by type\n5. Search by price\n6. Search by stock available\n0. Exit\n";
      cin >> choice;
      switch(choice)
      {
         case 1:
            supermarket = readMarket(supermarket, MAXPROD);
            break;
         case 2:
            printMarket(supermarket, MAXPROD);
            break;
         case 3:
            cout << "Please enter the brand: ";
            cin >> temp;
            searchBrand(supermarket, MAXPROD, temp);
            break;
         case 4:
            cout << "Please enter the type: ";
            cin >> temp;
            searchType(supermarket, MAXPROD, temp);
            break;
         case 5:
            cout << "Please enter the price: ";
            cin >> tempfloat;
            searchPrice(supermarket, MAXPROD, tempfloat);
            break;
         case 6:
            cout << "Please enter the stock: ";
            cin >> tempint;
            searchStock(supermarket, MAXPROD, tempint);
            break;
      }
   }
   /*string bar;
   while(bar != "0")
   {
```

```
        cout << "\n\n\nPlease scan a product: (0 to exit) ";
        cin >> bar;
        if(bar != "0") scanBarcode(supermarket, MAXPROD, bar);
    }*/
    return 0;
}
```

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

class car
{
   //private:
      string brand;
      string model;
   public:
      car()
      {
         brand = "EMPTY";
         model = "EMPTY";
      }
      void setBrand(string n) {brand = n;}
      void setModel(string n) {model = n;}
      string getBrand() {return brand;}
      string getModel() {return model;}
};


car* readData(car *, int &, int &);
void displayData(car *, int);
void searchBrand(car *, int);

int main()
{
   int ARRSIZE = 1;
   int nextAvailPos = 0;
   car *carsArray;
   carsArray = new car[ARRSIZE];
   int choice = -1;
   do
   {
      cout << "Please choose from the menu below:\n";
      cout << "1. Enter Data\n";
      cout << "2. Display Data\n";
      cout << "3. Search by Brand\n";
      cout << "0. Exit\n";
      cin >> choice;
      switch(choice)
      {
         case 1:
            carsArray = readData(carsArray, ARRSIZE, nextAvailPos);
            break;
         case 2:
            displayData(carsArray, ARRSIZE);
```

```
            break;
         case 3:
            searchBrand(carsArray, ARRSIZE);
            break;
         case 0:
            break;
      }
   }while(choice != 0);
   return 0;
}

//function readData(car *, int &, int &)
//the function populates the array with information from the user
//the function accepts a pointer to a car object (the array) and the size of the array
//the function returns back a pointer to a car object which is the array and is now populated with
data
car* readData(car *arr, int &arrSize, int &nextPos)
{
   int numCars = 0, newSize = 0; //numCars = how many cars the user is going to input; newSize =
new size of the array which is calculated by adding arrSize + numCars
   string tString; //temporary string to read keyboard input
   car *tmp; //temporary car object to copy everything from the car array and grow the array
   cout << "How many cars are you entering? ";
   cin >> numCars;
   for(int i = 0; i < numCars; i++) //repeat the reading process 'numCars' times
   {
      if(nextPos >= arrSize) //first check if the array is full
      {
         newSize = arrSize + numCars; //if you get here, the array is full - calculate new size
         tmp = new car[newSize]; //allocate temporary space for the new bigger array
         for(int j = 0; j < arrSize; j++) //copy everything from the small array to the new bigger one
            tmp[j] = arr[j];
         arrSize = newSize; //save the newSize into arrSize (update the arrSize)
         delete [] arr; //release the memory from the old array back into the OS
         arr = tmp; //set the pointer of the array to point to the new allocated space
      }
      cout << "Enter brand for car " << nextPos << ": ";
      cin >> tString;
      arr[nextPos].setBrand(tString);
      cout << "Enter model for car " << nextPos << ": ";
      cin >> tString;
      arr[nextPos].setModel(tString);
      nextPos++;
   }
   return arr;
}
```

```
void displayData(car *arr, int arrSize)
{
   cout << "\n\n\nYOU HAVE SAVED THE FOLLOWING INFORMATION\n";
   for(int i = 0; i < arrSize - 1; i++)
   {
      cout << "---------- Car " << i << "----------\n";
      cout << "Brand: " << arr[i].getBrand() << "\n";
      cout << "Model: " << arr[i].getModel() << "\n";
   }
}

void searchBrand(car *arr, int arrSize)
{
   string tString;
   cout << "Please enter the brand: ";
   cin >> tString;
   bool ok = false;
   for(int i = 0; i < arrSize && !ok; i++)
   {
      if(arr[i].getBrand() == tString)
      {
         ok = true;
         cout << "The car you are searching for is:\n";
         cout << "Brand: " << arr[i].getBrand() << "\n";
         cout << "Model: " << arr[i].getModel() << "\n";
      }
   }
   if(!ok) cout << "Couldn't find the car you are looking for.\n";
}
```