

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

Sending organization: Colegiul National ”Ferdinand” Bacau, Romania

MOBILITY PORTFOLIO
Paphos, Cyprus

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

Receiving

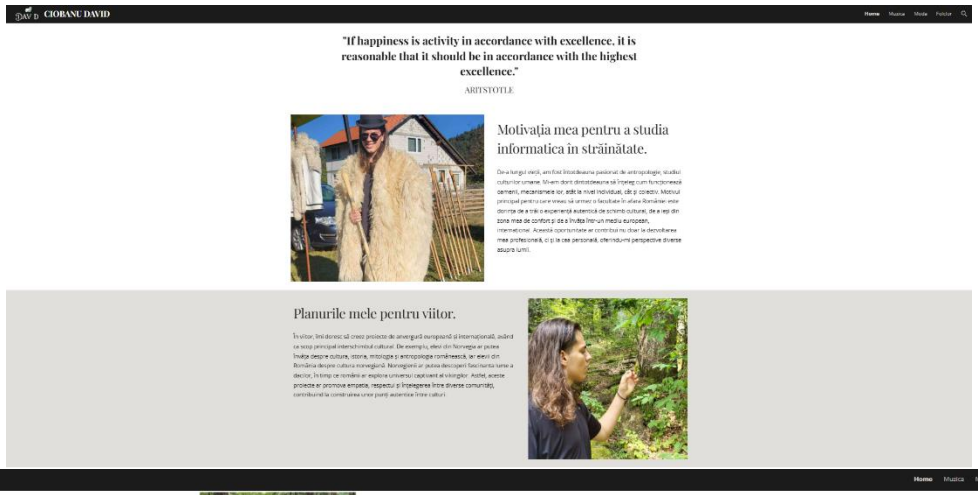
Practice tutor: ***Lazaros Lazarou***

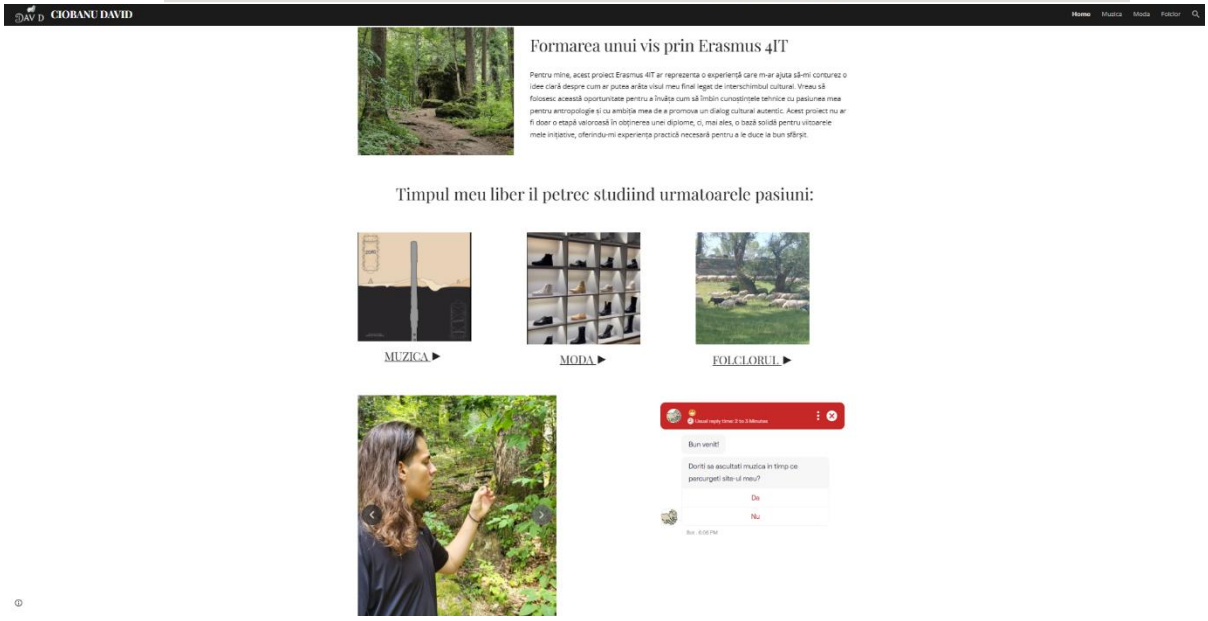
Participant: ***Ciobanu David-Alin***

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.

My personal website – About me
Ciobanu David Alin









MUZICA

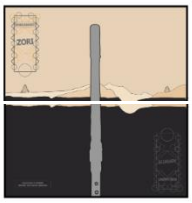
Principala mea pasiune este muzica.

Consider că muzica este cea mai eficientă formă de artă, întrucât reușește să transmită cel mai ușor emoțiile artistului. Acesta, pe lângă linia melodică, reușește să comunice sentimente, idei, gânduri și trăiri prin intermediul versurilor (literatură), al personalităților pe care le adoptă (arta vieții, actorie și modă), al copertei albumului (arta plastică), al videoclipurilor unei piese sau al filmului unui album (cinematografie, teatru), dar și prin direcția culturală din care face parte respectivul artist sau album.

Voi împărtăși câteva dintre albumele mele preferate din acest moment, cu intenția de a-i oferi cititorului șansa să pătrundă în universul meu, să simtă și să înțeleagă modul în care percep eu muzica.

Zori Si Asfintit

Subcarpati



Subcarpați este o trupă înființată în 2010 de Bean MC (Alexe Marius Andrei), artist și producător muzical. De-a lungul vieții, acesta a colaborat cu artiști precum Speci și Kazi Ploae, fiind un membru influent al mișcării hip-hop underground din perioada 2005-2010.

Bean mărturisește că nu a fost întotdeauna pasionat de muzica folclorică, povestind că bunica sa obișnuia să asculte muzică românească în casă. Totuși, mai târziu în viață, a fost cuprins de un dor pentru acest gen muzical, în special pentru muzica ușoară, relaxantă, interpretată la cavalul sau fluier. Declară că „stătea sub un tezaur muzical", care, începând din 2010, avea să devină principala sa sursă de inspirație.

Consider că albumul Zori și Asfințit, lansat în 2018, este „Opus Magnum"-ul trupei Subcarpați. În acea perioadă, trupa era formată din Aforic, Valentin Mușat, IOANA, Alin Stoianovici, Călin Hain și mulți alți instrumentiști și producători muzicali. Acest album reprezintă apogeul calității muzicale și lirice al formației, integrând influențe de folclor, rock progresiv, hip-hop, reggae și muzică ușoară.

Paradoxal, în ciuda acestor influențe din afara granițelor, albumul își păstrează puternic rădăcinile folclorice românești, datorită liniilor melodice ghidate și dominate de cavalul românesc.

De pe acest album, recomand piesele *„Dă-i Foaie"* ►, *„Animalul Din Mine"* ► și *„Marcel Temelie"* ►.

RADIOFONIC

Killa Fonic

Cu fiecare album, Killa Fonic ne arată că a lăsat în urmă personalitatea sa din perioada OSOD, concentrându-se acum pe alte priorități. Povestea sa este una de succes, ilustrând parcursul unui tânăr care își lasă în urmă vicile (inclusiv o mentalitate rigidă) și își descoperă noi valori și obiective în viață.

Ultimele 3-4 albume evidențiază o tendință de aprofundare a spiritualității. Este clar că Killa Fonic se culturalizează din punct de vedere religios și filosofic, inspirându-se masiv din cultura asiatică.

Comparativ cu debutul său, „Ramses 1989", un album dominat de piese trap, noile sale materiale includ doar câteva creații din acest gen. Restul reprezintă o expresie unică, cu influențe din muzica asiatică, jazz, electronică, muzica clasică și multe altele. Sincer, îmi este foarte greu să-i înțeleg pe Killa Fonic ca artist, iar acest lucru îl face preferatul meu. Muzica sa este o provocare intelectuală, un fel de codex ce așteaptă să fie descifrat.

În piesele sale, Killa menționează frecvent teatrul universal (muzicalul), cinematografia asiatică și spiritualitatea universală, teme pe care personal nu le-am explorat încă în profunzime. Îi respect ca artist pentru că este suficient de complex încât să nu îi pot înțelege pe deplin arta – un lucru pe care nu-l pot spune despre mulți alții.

De pe acest album recomand piesele

„Păsări migratoare" ►, *„Dor de sine"* ► și *„În vânt"* ►.

ALL SOULS

Siouxsie and the Banshees



Siouxsie and the Banshees reprezintă o emblemă a mișcării post-punk din Londra, activând în perioada 1970-1990, iar compilația „All Souls" este un exemplu perfect al tendinței de experimentare muzicală caracteristice acelei epoci. Sunetul clasic post-punk se îmbină cu o atmosferă electrică, brută și zgârietoare, emanând o energie de nedescris.

Estetica acestei compilații de piese reprezentative trupei este deschisă, experimentală, parțial întunecată, dar totodată vibrantă și electrică. Versurile sunt abstracte, profunde și atent compuse, fiecare cuvânt contribuind la starea generală transmisă. Din punct de vedere liric, „All Souls" explorează teme precum misticismul, moartea și viața de apoi, izolarea emoțională și trecerea ireversibilă a timpului. Paleta sonoră a compilației se aliniază perfect acestor teme reci și intense, amplificând impactul emoțional.

Piesa „We Hunger", care apare și pe albumul "Hyaena", este preferata mea din toate timpurile. Încărcătura simbolică și atmosfera electrizantă creează o experiență copleșitoare, aproape transcendentă. Ritmul hipnotizant, versurile enigmatice și atmosfera gotică mă cuceresc complet, transformând audiția într-o experiență incomparabilă.

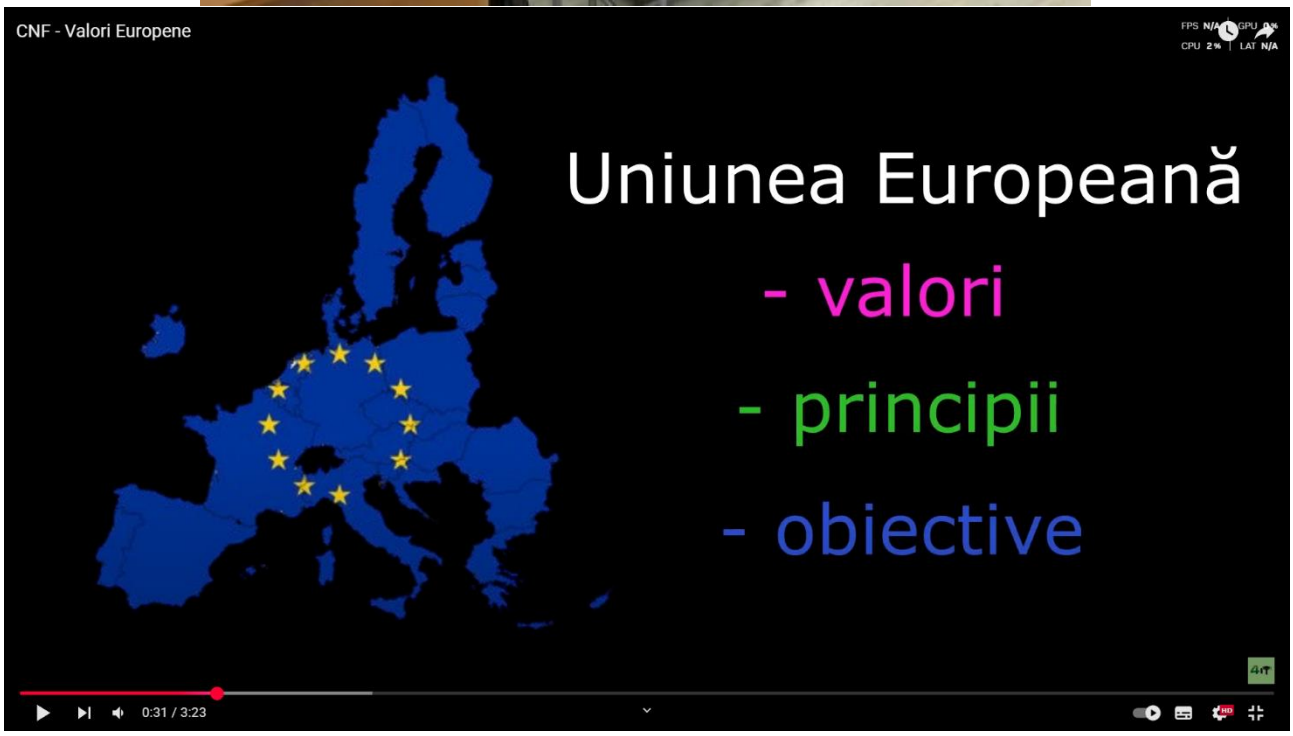
De pe această compilație, recomand piesele *„Halloween"* ►, *„We Hunger"* ► și *„Something Wicked"* ►.

◄ INAPOI INAINTE ►

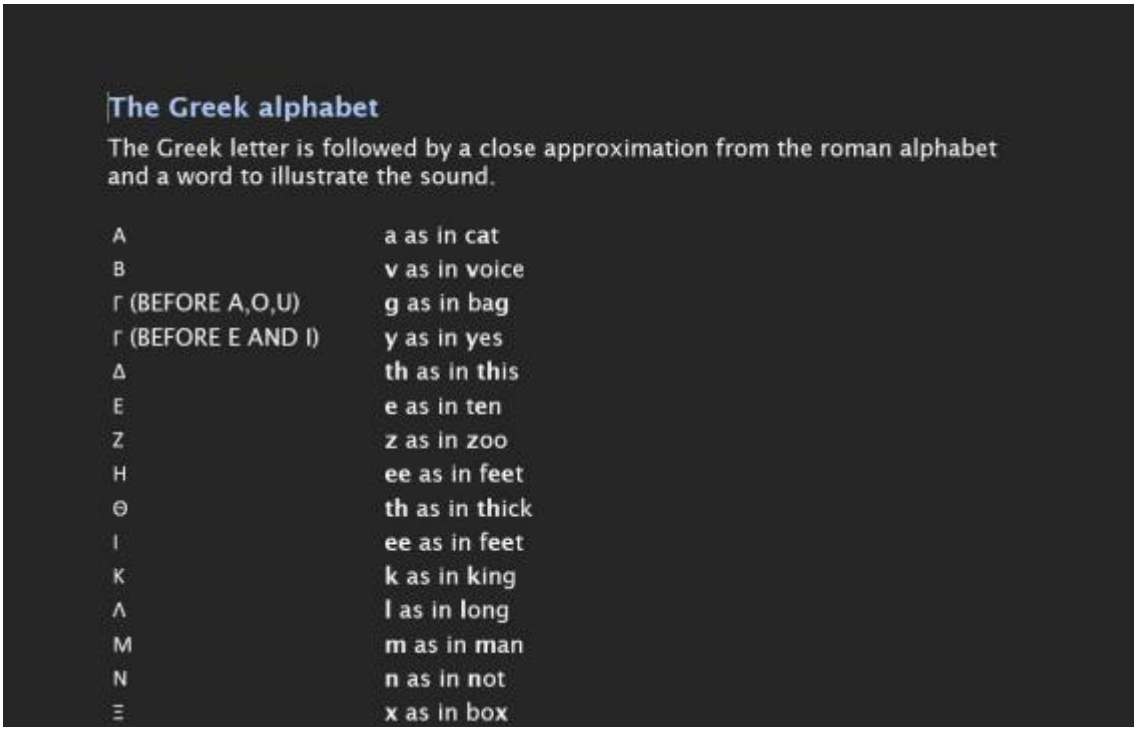
Training before mobility

1. Cultural training
-european values, active citizenship and active involvement (With Demetra)

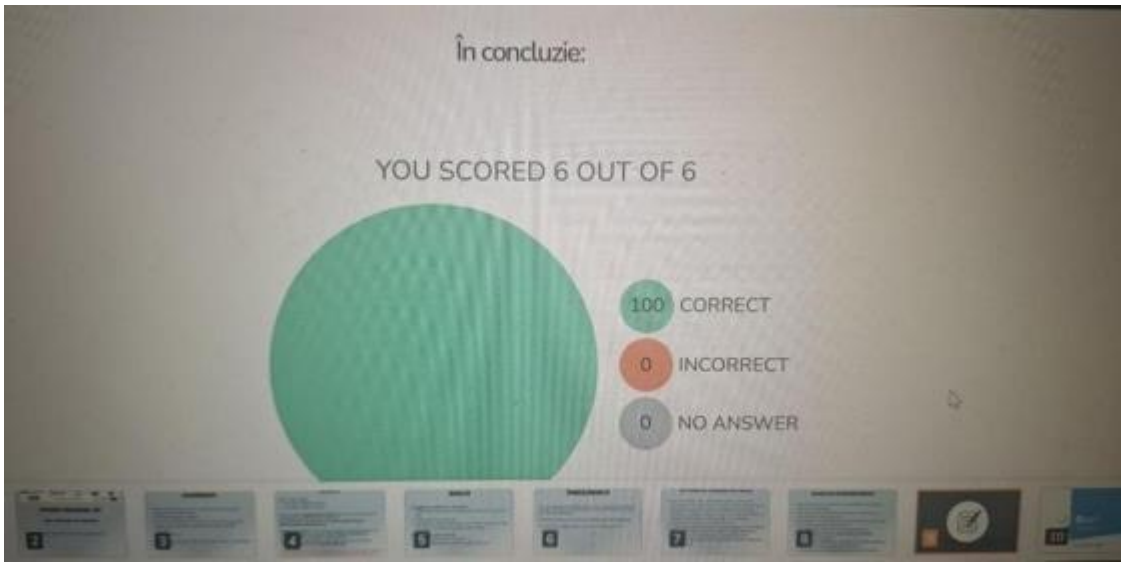




2. Linguistic training (With Stavroula Metaxa)



3. Pedagogical training (with Ema Fâciu)



Mobility projects

Understanding dynamic arrays

```cpp
#include <iostream>
using namespace std;
int main()
{
    int arraySize;
    cout<<"Enter array size:";
    cin>>arraySize;
    int *dynamicArray;
    dynamicArray= new int[arraySize];
    for(int i=0; i<arraySize; i++)
    {
        dynamicArray[i]=i+1;
    }
    cout<<endl;
    for(int i=0; i<arraySize; i++)
    {
        if(i<arraySize-1)
        {
            cout<<dynamicArray[i]<<" - ";
        }
        else cout<<dynamicArray[i];

    }
    cout<<endl;
    delete dynamicArray;
    return 0;
}
```

```
Enter array size:5

1 - 2 - 3 - 4 - 5

Process returned 0 (0x0)   execution time : 0.653 s
Press any key to continue.
```

Make a dynamic array and show the ASCII value as well.

```cpp
#include <iostream>
using namespace std;
/*
This example is showing you how to dynamically grow an array during run time.
It uses pointers to create the array.
*/

int main(){
    int ARRAYSIZE = 0;                          //the ARRAYSIZE is how many numbers the user wants to input. This might change if the user wishes to change it!
    int x;
    char* a;                                     //a is a pointer to a character. This is going to be our main array to save characters
    char* temp;                                  //temp is a pointer to a character and is going to be a temporary array. We need it in order to be able to GROW our array.
    int  n = 0, tmpN=0;                         //n is the counter to count the iterations. tmpN is a number we read from the user in order to see if they need to input more numbers than the initial decision.

    //--- Read into the array
    cout<<"How many numbers would you like to enter?: "; //ask the user and create an array according to the user's demand.
    cin>>ARRAYSIZE;                             //read the size of the array from the user
    a = new char[ARRAYSIZE];                    //dynamically create an array with the number the user gave to us.

    do{
        cout<<"Enter number "<<n+1<<" -> ";     //prompt the user to enter the numbers
        cin>>x;
        if(x >= 65 && x <= 90){
            a[n] = (char)x;                     // save the number as a character (e.g., ASCII A-Z)
            cout<<"Number saved!"<<endl;
            n++;                                //after saving the number we have to increase the counter
        }
        else {
            cout<<"Please enter a number between 65 and 90"<<endl;
        }

        if (n >= ARRAYSIZE){                    //if the new counter is equal to the ARRAYSIZE we need to GROW THE ARRAY
            cout<<"Out of Space!!!! Would you like to enter more? Please enter how many (0 to Exit): ";
            cin>>tmpN;
            if (tmpN > 0){
                ARRAYSIZE = ARRAYSIZE + tmpN;
                temp = new char[ARRAYSIZE];
                for (int i = 0; i < n; i++){
                    temp[i] = a[i];
                }
                delete [] a;
                a = temp;
            }
        }

    } while (n < ARRAYSIZE);

    cout<<"Now printing the numbers you inputted"<<endl;

    for (int i = 0; i < n; i++){                // only print up to n, the number of valid inputs
        cout<<a[i]<<" ("<<(int)a[i]<<")"<<endl;  // optional: print ASCII value too
    }

    delete [] a; //free the memory of a

    return 0;
}
```

```
How many numbers would you like to enter?: 3
Enter number 1 -> 67
Number saved!
Enter number 2 -> 70
Number saved!
Enter number 3 -> 73
Number saved!
Out of Space!!!! Would you like to enter more? Please enter how many (0 to Exit): 1
Enter number 4 -> 24
Please enter a number between 65 and 90
Enter number 4 -> 81
Number saved!
Out of Space!!!! Would you like to enter more? Please enter how many (0 to Exit): 0
Now printing the numbers you inputted
C (67)
F (70)
I (73)
Q (81)

Process returned 0 (0x0)   execution time : 12.661 s
Press any key to continue.
```

Make a dynamic array saving people's telephone number and name. At the end be able to search with the telephone number or name.

```
using namespace std;

int main()
{
    string* name;
    string* phone;
    string ask;
    int arrsize=0;
    int i=0,ok=0;

    cout<<"Enter the number of students that you need to save: ";
    cin>>arrsize;

    name= new string[arrsize];
    phone= new string[arrsize];

    for(i=0; i<arrsize; i++){
        cout<<"Enter Student name:";
        cin>>name[i];
        cout<<"Enter phone number for "<<name[i]<<":";
        cin>>phone[i];
    }

    cout<<"You saved: "<<endl<<"----------------------------"<<endl;
    for(int i=0;i<arrsize;i++){
        cout<<name[i]<<" - "<<phone[i]<<endl;
    }


    do{
        cout<<"----------------------------"<<endl;
        cout<<"What do you want to search for:"<<endl;
        cout<<"1 for telephone number"<<endl;
        cout<<"2 for name"<<endl;
        cout<<"0 to exit"<<endl;
        cin>>ok;
        if(ok==1){
            cout<<"----------------------------"<<endl;
            cout<<"Who's telephone number do you want: "<<endl;
            cin>>ask;
            for(i=0;i<arrsize;i++){
                if(ask==name[i]){
                    cout<<"----------------------------"<<endl;
                    cout<<name[i]<<" - "<<phone[i]<<endl;
                }
            }
        }
        else if(ok==2){
            cout<<"----------------------------"<<endl;
            cout<<"What's the telephone number of the person: "<<endl;
            cin>>ask;
            for(i=0;i<arrsize;i++){
                if(ask==phone[i]){
                    cout<<"----------------------------"<<endl;
                    cout<<name[i]<<" - "<<phone[i]<<endl;
                }
        }
        }
    }while(ok>0);
```

```
Enter the number of students that you need to save: 2
Enter Student name:David
Enter phone number for David:0772858219
Enter Student name:Alex
Enter phone number for Alex:0788291532
You saved:
----------------------------
David - 0772858219
Alex - 0788291532
----------------------------
What do you want to search for:
1 for telephone number
2 for name
0 to exit
1
----------------------------
Who's telephone number do you want:
David
----------------------------
David - 0772858219
----------------------------
What do you want to search for:
1 for telephone number
2 for name
0 to exit
2
----------------------------
What's the telephone number of the person:
0788291532
----------------------------
Alex - 0788291532
----------------------------
What do you want to search for:
1 for telephone number
2 for name
0 to exit
0

Process returned 0 (0x0)   execution time : 24.619 s
Press any key to continue.
```

Write a program for a supermarket which is going to:

Take the information for the different products a supermarket sells and it will save the following information: -Brand of product -Type of product -Price -Stock Remaining -Your program must be able to search for products by name and by Type

```cpp
#include <iostream>

using namespace std;

class products{
private:
    string Brand;
    string Type;
    string Price;
    string Stock;
public:
    products(){
        Brand="Empty";
        Type="Empty";
        Price="0.00";
        Stock="0.00";
    }
    void setBrand(string b){Brand=b;}
    void setType(string t){Type=t;}
    void setPrice(string p){Price=p;}
    void setStock(string s){Stock=s;}
    string getBrand(){return Brand;}
    string getType(){return Type;}
    string getPrice(){return Price;}
    string getStock(){return Stock;}

};

int main()
{
    int ok=0;
    const int MAXPRODUCTS=3;
    products inventory[MAXPRODUCTS];
    string tBrand="", tType="", tPrice="", tStock="";
    string ask;
    int exist=0;
    for(int i=0;i<MAXPRODUCTS;i++){
        cout<<"Brand "<<i+1<<": ";
        cin>>tBrand;
        inventory[i].setBrand(tBrand);

        cout<<"Type "<<i+1<<": ";
        cin>>tType;
        inventory[i].setType(tType);

        cout<<"Price "<<i+1<<": ";
        cin>>tPrice;
        inventory[i].setPrice(tPrice);

        cout<<"Stock "<<i+1<<": ";
        cin>>tStock;
        inventory[i].setStock(tStock);
    }

    cout<<endl<<endl<<endl;
    cout<<"---------YOU HAVE SAVED THE FOLLOWING INFO ------------"<<endl;

    for(int i=0;i<MAXPRODUCTS;i++){
        cout<<inventory[i].getBrand()<<" - "<<inventory[i].getType()<<endl;
        cout<<inventory[i].getPrice()<<" - "<<inventory[i].getStock()<<endl<<endl;
    }

    cout<<"-------------------------------------------------------"<<endl;


    do{
        cout<<"What do you want to search for:"<<endl;
        cout<<"1 for Brand"<<endl;
        cout<<"2 for Type"<<endl;
        cout<<"0 to exit"<<endl;
        cin>>ok;
        exist=0;
        if(ok==1){
            cout<<"Please enter the type, the price or the stock: ";
            cin>>ask;
            cout<<"-------------------------------------------------------"<<endl;
            for(int i=0;i<MAXPRODUCTS;i++){
                if(ask==inventory[i].getType() || ask==inventory[i].getPrice() || ask==inventory[i].getStock()){
                    cout<<inventory[i].getBrand()<<" - "<<inventory[i].getType()<<endl;
                    cout<<inventory[i].getPrice()<<" - "<<inventory[i].getStock()<<endl;
                    cout<<"-------------------------------------------------------"<<endl;
                    exist=1;
                }
            }
            if(exist==0){
                cout<<"This item is not stored"<<endl;
                cout<<"-------------------------------------------------------"<<endl;
            }

        }
        else if(ok==2){
            cout<<"Please enter the brand, the price or the stock: ";
            cin>>ask;
            cout<<"-------------------------------------------------------"<<endl;
            for(int i=0;i<MAXPRODUCTS;i++){
                if(ask==inventory[i].getBrand() || ask==inventory[i].getPrice() || ask==inventory[i].getStock()){
                    cout<<inventory[i].getBrand()<<" - "<<inventory[i].getType()<<endl;
                    cout<<inventory[i].getPrice()<<" - "<<inventory[i].getStock()<<endl;
                    cout<<"-------------------------------------------------------"<<endl;
                    exist=1;
                }
            }
            if(exist==0){
                cout<<"This item is not stored"<<endl;
                cout<<"-------------------------------------------------------"<<endl;
            }
        }
    }while(ok>0);



    return 0;
}
```

```
Brand 1: Pilot
Type 1: GelInk
Price 1: 2.49$
Stock 1: 120
Brand 2: Lindt
Type 2: Chocolate
Price 2: 3.99$
Stock 2: 85
Brand 3: Anker
Type 3: Charger
Price 3: 19$
Stock 3: 47



---------YOU HAVE SAVED THE FOLLOWING INFO ------------
Pilot - GelInk
2.49$ - 120

Lindt - Chocolate
3.99$ - 85

Anker - Charger
19$ - 47

-------------------------------------------------------
What do you want to search for:
1 for Brand
2 for Type
0 to exit
2
Please enter the brand, the price or the stock: 19$
-------------------------------------------------------
Anker - Charger
19$ - 47
-------------------------------------------------------
What do you want to search for:
1 for Brand
2 for Type
0 to exit
1
Please enter the type, the price or the stock: Chocolate
-------------------------------------------------------
Lindt - Chocolate
3.99$ - 85
-------------------------------------------------------
What do you want to search for:
1 for Brand
2 for Type
0 to exit
0

Process returned 0 (0x0)   execution time : 57.698 s
Press any key to continue.
```

Write a program for a car dealership which is going to:

Take the information for the different cars and it will save the following information: -Plate number -Owner -Brand -Model -Your program must be able to search for cars by everything

```
#include <iostream>

using namespace std;

class car
{
private:
    string brand;
    string model;
    string plateNum;
    string owner;
public:
    void setBrand(string B)
    {
        brand=B;
    }
    void setModel(string M)
    {
        model=M;
    }
    void setPlateNum(string P)
    {
        plateNum=P;
    }
    void setOwner(string O)
    {
        owner=O;
    }

    string getBrand()
    {
        return brand;
    }
    string getModel()
    {
        return model;
    }
    string getPlateNum()
    {
        return plateNum;
    }
    string getOwner()
    {
        return owner;
    }

    car()
    {
        brand="Empty";
        model="Empty";
        plateNum="Empty";
        owner="Empty";
    }
};

void enterdata(car*& specs,int& MAXCARS, int& n){
    string temp, temp2;
    int tempn;
    int* tempspecs;

//read the info from the user
    do{
            if(n>=MAXCARS){
            cout<<"Out of space!! How many more do you want to enter? (0 to exit)";
            cin>>tempn;
            if(tempn>0){
                car* newSpecs = new car[MAXCARS+tempn];

                for(int i=0;i<MAXCARS;i++){
                    newSpecs[i]=specs[i];
                }
                delete[] specs;
                specs=newSpecs;
                MAXCARS+=tempn;
            }
        }
        cout<<"---------- CAR "<<n+1<<" ----------"<<endl;

        cout<<"Enter brand: ";
        cin>>temp;
        specs[n].setBrand(temp);

        cout<<"Enter model: ";
        cin>>temp;
        specs[n].setModel(temp);

        cout<<"Enter plate number: "; // input is" LSP 203
        cin>>temp>>temp2;             //the ">>" operator reads up to the white space in a sentence
        temp=temp+" "+temp2;
        specs[n].setPlateNum(temp);

        cout<<"Enter the name of the owner: ";
        cin>>temp;
        specs[n].setOwner(temp);

        n++;

    }while(n<MAXCARS);
}

void displaydata(car specs[], int MAXCARS){
cout<<endl<<endl;
    cout<<"---------- YOU HAVE SAVED ----------"<<endl<<endl;

    for(int i=0; i<MAXCARS; i++)
    {
        cout<<"Brand: "<<specs[i].getBrand()<<endl;
        cout<<"Model: "<<specs[i].getModel()<<endl;
        cout<<"Plate number: "<<specs[i].getPlateNum()<<endl;
        cout<<"Owner: "<<specs[i].getOwner()<<endl<<endl;
    }
}

void Brand(car specs[], int MAXCARS){
    string temp;
    cout<<"Which brand are you looking for: ";
    cin>>temp;
    cout<<"-----------------------------------"<<endl;
    for(int i=0;i<MAXCARS;i++){
        if(specs[i].getBrand()==temp){
            cout<<"Brand: "<<specs[i].getBrand()<<endl;
            cout<<"Model: "<<specs[i].getModel()<<endl;
            cout<<"Plate number: "<<specs[i].getPlateNum()<<endl;
            cout<<"Owner: "<<specs[i].getOwner()<<endl<<endl;
        }
    }
}

void Model(car specs[], int MAXCARS){
    string temp;
    cout<<"Which model are you looking for: ";
    cin>>temp;
    cout<<"-----------------------------------"<<endl;
    for(int i=0;i<MAXCARS;i++){
        if(specs[i].getModel()==temp){
            cout<<"Brand: "<<specs[i].getBrand()<<endl;
            cout<<"Model: "<<specs[i].getModel()<<endl;
            cout<<"Plate number: "<<specs[i].getPlateNum()<<endl;
            cout<<"Owner: "<<specs[i].getOwner()<<endl<<endl;
        }
    }
}

void Owner(car specs[], int MAXCARS){
    string temp;
    cout<<"Which owner are you looking for: ";
    cin>>temp;
    cout<<"-----------------------------------"<<endl;
    for(int i=0;i<MAXCARS;i++){
        if(specs[i].getOwner()==temp){
            cout<<"Brand: "<<specs[i].getBrand()<<endl;
            cout<<"Model: "<<specs[i].getModel()<<endl;
            cout<<"Plate number: "<<specs[i].getPlateNum()<<endl;
            cout<<"Owner: "<<specs[i].getOwner()<<endl<<endl;
        }
    }
}

void PlateNumber(car specs[], int MAXCARS){
    string temp,temp2;
    cout<<"Which plate number are you looking for: ";
    cin>>temp>>temp2;
    temp=temp+" "+temp2;
    cout<<"-----------------------------------"<<endl;
    for(int i=0;i<MAXCARS;i++){
        if(specs[i].getPlateNum()==temp){
            cout<<"Brand: "<<specs[i].getBrand()<<endl;
            cout<<"Model: "<<specs[i].getModel()<<endl;
            cout<<"Plate number: "<<specs[i].getPlateNum()<<endl;
            cout<<"Owner: "<<specs[i].getOwner()<<endl<<endl;
        }
    }
}

int main()
{
    int MAXCARS=2; // variable for MAXCARS
    car* specs = new car[MAXCARS];
    int ok=0,n=0;

    do{
        cout<<"---------------------------"<<endl;
        cout<<"What do you want to do:"<<endl;
        cout<<"1 to enter data"<<endl;
        cout<<"2 to display data"<<endl;
        cout<<"3 to search by brand"<<endl;
        cout<<"4 to search by model"<<endl;
        cout<<"5 to search by owner"<<endl;
        cout<<"6 to search by plate number"<<endl;
        cout<<"0 to exit"<<endl;
        cin>>ok;

        if(ok==1){
            enterdata(specs, MAXCARS, n);
        }
        else if(ok==2){
            displaydata(specs, MAXCARS);
        }
        else if(ok==3){
            Brand(specs, MAXCARS);
        }
        else if(ok==4){
            Model(specs, MAXCARS);
        }
        else if(ok==5){
            Owner(specs, MAXCARS);
        }
        else if(ok==6){
            PlateNumber(specs, MAXCARS);
        }


    }while(ok>0);


    return 0;
}
```

```
---------------------------
What do you want to do:
1 to enter data
2 to display data
3 to search by brand
4 to search by model
5 to search by owner
6 to search by plate number
0 to exit
1
---------- CAR 1 ----------
Enter brand: Mercedes-Benz
Enter model: C-Class
Enter plate number: B 245
Enter the name of the owner: Elena
---------- CAR 2 ----------
Enter brand: Ford
Enter model: Kuga
Enter plate number: CJ 87
Enter the name of the owner: Radu
---------------------------
What do you want to do:
1 to enter data
2 to display data
3 to search by brand
4 to search by model
5 to search by owner
6 to search by plate number
0 to exit
5
Which owner are you looking for: Radu
-----------------------------------
Brand: Ford
Model: Kuga
Plate number: CJ 87
Owner: Radu

---------------------------
What do you want to do:
1 to enter data
2 to display data
3 to search by brand
4 to search by model
5 to search by owner
6 to search by plate number
0 to exit
3
Which brand are you looking for: Mercedes-Benz
-----------------------------------
Brand: Mercedes-Benz
Model: C-Class
Plate number: B 245
Owner: Elena

---------------------------
What do you want to do:
1 to enter data
2 to display data
3 to search by brand
4 to search by model
5 to search by owner
6 to search by plate number
0 to exit
0

Process returned 0 (0x0)   execution time : 51.279 s
Press any key to continue.
```

Understanding arguments

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

int main(int argc, char* argv[]) //argc = argument count
{
    cout<<"You have entered "<<argc<<" arguments:"<<endl;

    //Using a while loop to iterate through arguments
    int i=1;
    while(i<argc)
    {
        cout<<"Argument "<<i+1<<": "<<argv[i]<<endl;
        if (strcmp(argv[i],"/h")==0)
        {
            //write code to show help
        }
        if (strcmp(argv[i],"/s")==0)
        {
            //write code to implement /s option
        }
        if (strcmp(argv[i],"/t")==0)
        {
            //write code to implement /t option
        }
    }
    return 0;
}
```

```
You have entered 1 arguments:

Process returned 0 (0x0)   execution time : 0.040 s
Press any key to continue.
```