

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

Sending organization: Colegiul National ”Ferdinand I” 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

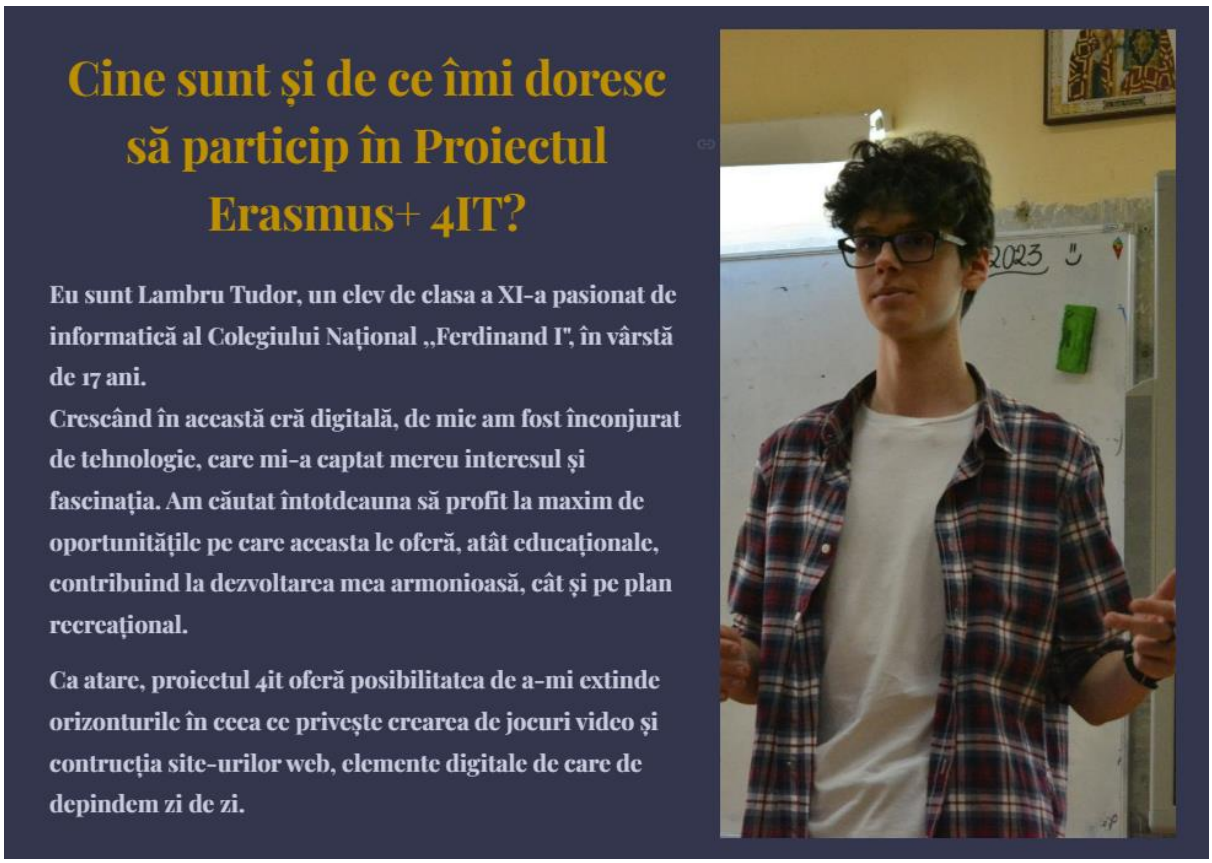
Practice tutor: ***Lazaros Lazarou***

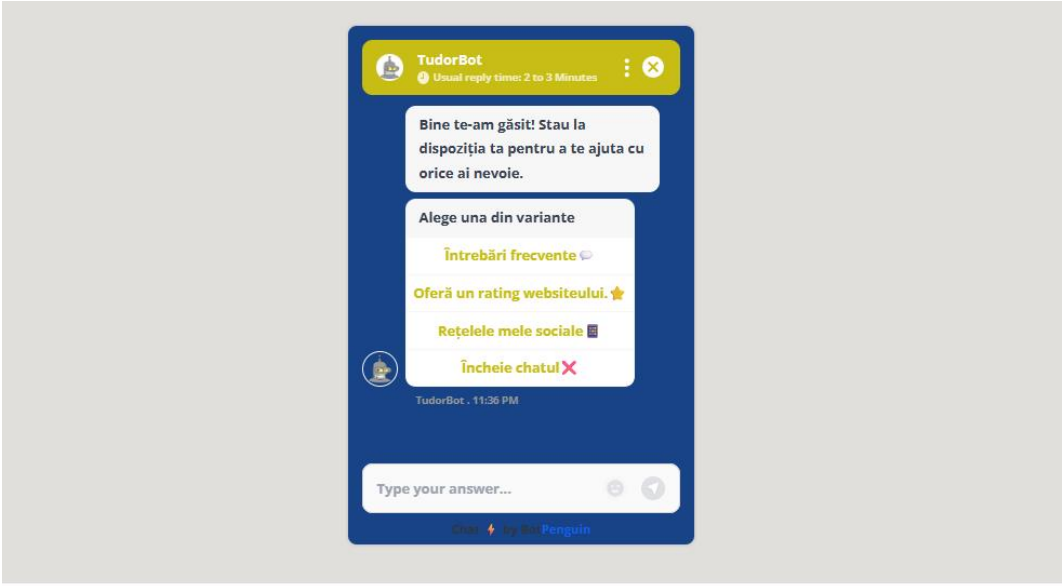
Participant: ***Lambru Tudor***

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
Lambru Tudor







Site realizat de Lambru Tudor, elev al Colegiului „Național Ferdinand I"

National College Ferdinand · Strada George Bacovia 45, Bacău 600237, Romania

★★★★☆ · High school



Ce valori mă definesc?

Pentru a mă descrie folosind doar trei cuvinte, aș alege respect, toleranță și seriozitate. Mă identific drept o fire serioasă, perseverentă, foarte analitică, ce gândește de două ori înainte să ia o decizie și se hrănește cu informație. Prietenii și persoanele din jurul meu mă consideră adesea un model de om dedicat și corect.



Aspirațiile mele

Deși de fel sunt ambițios și provocările noi mă inspiră să iau atitudine, nu țintesc la a-mi clădi un viitor extravagant. Prefer un stil de viață liniștit, dar care să îmi satisfacă nevoile culturale (călătorie, artă).

Pe plan profesional, îmi doresc să reușesc în domeniul IT ca Software Developer într-un oraș mare, meserie care mi s-ar potrivi datorită îndemânării privind soluționarea problemelor și a calităților mele de autodidact, pe lângă pasiunea pe care o afișez față de evoluția constantă a tehnologiei, urmărind cu atenție fiecare nouă inovație.

Despre hobby-uri și pasiuni

Totuși, este necesar pentru fiecare dintre noi să se dezvolte pe cât mai multe planuri. De aceea, am un set bine stabilit de hobby-uri și activități ce-mi aduc plăcere. În cazul meu, deconectarea de la viața cotidiană se face prin intermediul artei, în special muzica, filmele și cărțile. Pentru menținerea unei bune sănătăți fizice practic sporturi precum alergatul, alpinismul și fotbalul.

O componentă esențială a secolului XXI este, de asemenea, călătoritul, și mă pot lăuda cu o sumedenie de destinații de referință, în care mi-am putut forma un portofoliu vast de fotografii.
De aceea, filozofia mea de viață are la bază încercarea a câtor mai multe lucruri, spre a-mi extinde orizonturile și a deveni, pe cât posibil, atotputernic din punct de vedere intelectual.



Site realizat de Lambru Tudor, elev al Colegiului „Național Ferdinand I"

National College Ferdinand · Strada George Bacovia 45, Bacău 600237, Romania

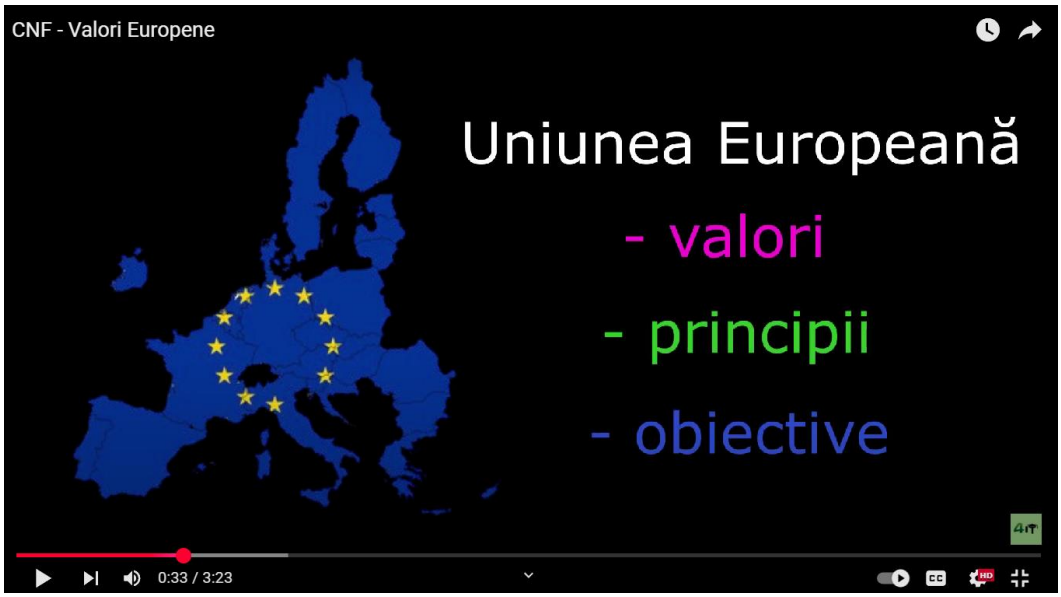
★★★★☆ · High school

Training before mobility

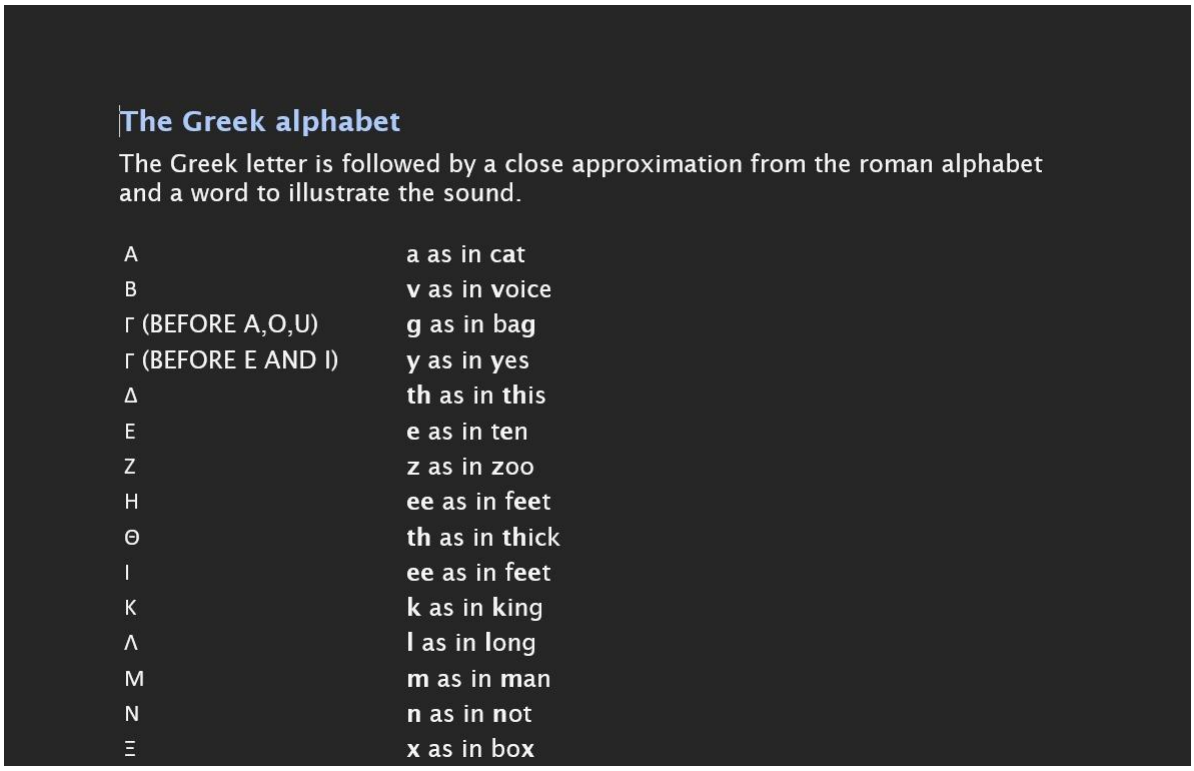
1. Cultural training

- **European values, active citizenship and active involvement**

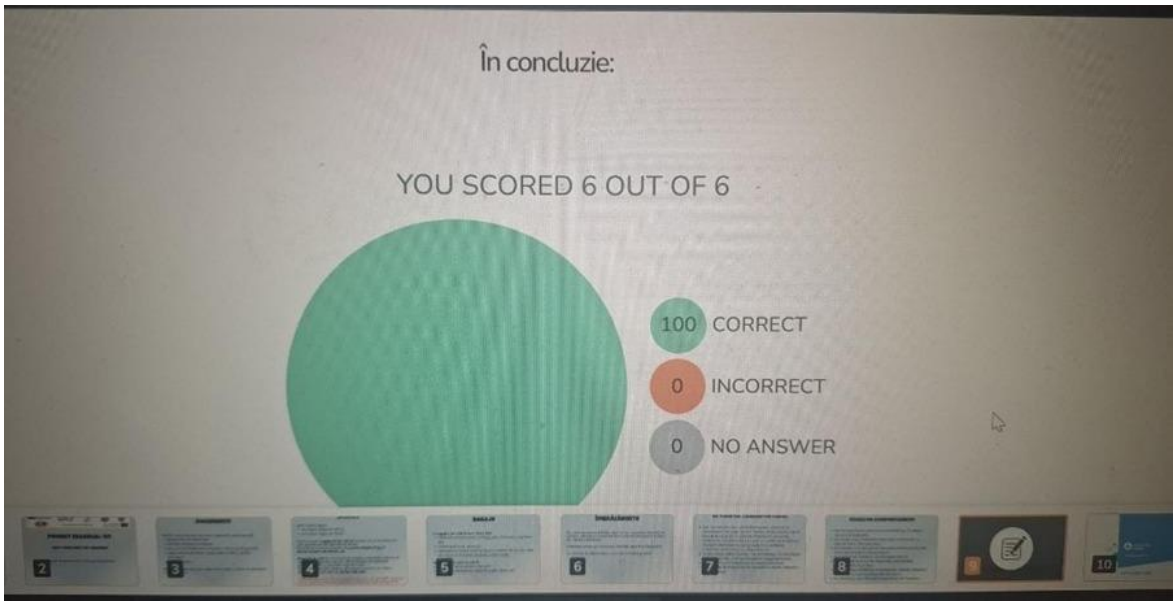




2.Linguistic training (with Stavroula Metaxa)



3. Pedagogical training (with Ema Fâciu)



Mobility projects

Arguments - Introduction

```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;
}

```

Write a program where you will have a dynamic array to save characters. The user will input numbers that range from 65 to 90, which will be converted to letters and saved into a dynamic array. The program will be able to make the array bigger

```
#include <iostream>
using namespace std;
int main()
{
    int ARRAYSIZE = 0;
    int* a;
    int* temp;
    int  n = 0, tmpN=0;
    cout<<"How many numbers would you like to enter?"<<endl;
    cin>>ARRAYSIZE;
    a = new int[ARRAYSIZE];

    do
    {
        cout<<"Enter number "<<n<<":";
        cin >> a[n];
        if(!(a[n]>=65 && a[n]<=90))
        {
            cout<<"Please input numbers between 65 and 90!";
        }
        else
        {
            cout<<"Number saved!"<<endl;
            n++;
        }

        if (n >= ARRAYSIZE)
        {
            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 int[ARRAYSIZE];
                  for (int i=0; i<n; i++)
                  {
                      temp[i] = a[i];
                  }
                  delete [] a;
                  a = temp;
              }
          }

    }
    while (n<ARRAYSIZE);

    cout<<"Now printing the letters that correspond to the ASCII codes you inputed"<<endl;

    for (int i=0; i<ARRAYSIZE; i++)
    {

        cout<<char(a[i])<<endl;
    }
    delete [] a;

    return 0;
}
```

Collect name and telephone numbers. Use two dynamic arrays (parallel arrays). Search for a name and return the phone number.

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    string *names;
    string *phone;
    string* tempnames;
    string* tempphone;
    string name;
    string number;
    int n=0, tempN, i, answer, found, pos, arrSize=0;
    cout<<"How many students do you want to store? ";
    cin>>arrSize;
    names = new string[arrSize];
    phone = new string[arrSize];

    do
    {
        cout<<"Enter student "<<n+1<<": ";
        cin>>names[n];
        cout<<"Enter "<<names[n]<<"'s phone number: ";
        cin>>phone[n];
        n++;
        if(n>=arrSize)
        {
            cout<<"You're out of space! How many more people do you want to enter?(0 to stop) ";
            cin>>tempN;
            if(tempN>0)
            {
                arrSize=arrSize+tempN;
                tempnames= new string[arrSize];
                tempphone= new string[arrSize];
                for(int i=0; i<n; i++)
                {
                    tempnames[i]=names[i];
                    tempphone[i]=phone[i];
                }
                delete[] names;
                delete[] phone;
                names= tempnames;
                phone= tempphone;
            }
        }
    }
    while(n<arrSize);
    cout<<"You saved ----------------------------------------------"<<endl;
    for(i=0; i<arrSize; i++)
    {
        cout<<"Name: "<<names[i]<<" Phone number: "<<phone[i]<<endl;
    }

    do
    {
        cout<<"Are you searching by name(type 1) or phone number(type 2)? 0 to exit ";
        cin>>answer;
        if (answer==1)
        {
            found=0;
            cout<<"Whose phone number do you need? ";
```

```
            cin>>name;
            for(i=0; i<arrSize; i++)
            {
                if(name==names[i])
                {
                    found=1;
                    pos=i;
                    break;
                }
            }
            if(found)
            {
                cout<<name<<"'s phone number is: "<<phone[pos]<<endl;
            }
            else cout<<"This person can't be found."<<endl;


        }
        else if (answer==2)
        {
            found=0;
            cout<<"Whose name are you searching? ";
            cin>>number;
            for(i=0; i<arrSize; i++)
            {
                if(number==phone[i])
                {
                    found=1;
                    pos=i;
                    break;
```

```
                }
            }
            if(found)
            {
                cout<<phone[pos]<<" belongs to: "<<names[pos]<<endl;
            }
            else cout<<"This phone number cannot be identified."<<endl;
        }

    }
    while(answer>0);


    return 0;
}
```

A program that's supposed to work as a database that stores car manufacturer, model, plate number and owner, using classes and dynamic arrays

```
#include <iostream>
using namespace std;
class car
{
private:
    string Brand;
    string Model;
    string NumberPlate;
    string Owner;
public:
    car()
    {
        Brand="EMPTY";
        Model="EMPTY";
        NumberPlate="EMPTY";
        Owner="EMPTY";
    }
    void setBrand(string b)
    {
        Brand=b;
    }
    void setModel(string m)
    {
        Model=m;
    }
    void setPlate(string p)
    {
        NumberPlate=p;
    }
    void setOwner(string o)
    {
        Owner=o;
    }
    string getBrand()
    {
        return Brand;
```

```
39          string getModel()
40          {
41              return Model;
42          }
43          string getPlate()
44          {
45              return NumberPlate;
46          }
47          string getOwner()
48          {
49              return Owner;
50          }
51      };
52      car* enterData(car* myclass, int MAXCARS)
53      {
54          string tbrand, tmodel, tplate, towner;
55
56          for(int i = 0; i < MAXCARS; i++)
57          {
58              cout << "-------------Car " << i + 1 << "-------------" << endl;
59
60              cout << "Brand: ";
61              getline(cin, tbrand);
62              myclass[i].setBrand(tbrand);
63
64              cout << "Model: ";
65              getline(cin, tmodel);
66              myclass[i].setModel(tmodel);
67
68              cout << "Number plate: ";
69              getline(cin, tplate);
70              myclass[i].setPlate(tplate);
71
72              cout << "Owner: ";
73              getline(cin, towner);
74          myclass[i].setOwner(towner);
75
76          cout << endl;
77      }
78
79      return myclass;
80  }
81
82
83  void search(car* myclass, int MAXCARS)
84  {
85      int answer, found, i, pos;
86      string searchedBrand, searchedModel, searchedPlate, searchedOwner;
87      cout<<endl<<"Search by"<<endl<<"1. Brand"<<endl<<"2. Model"<<endl<<"3. Plate number"<<endl<<"4. Owner"<<endl<<"Please insert the number of your preferred option, 0 to exit: ";
88      cin>>answer;
89      if(answer==1)
90      {
91          found=0;
92          cout<<"Enter car brand: ";
93          getline(cin,searchedBrand);
94          for(i=0; i<MAXCARS; i++)
95          {
96              if(searchedBrand==myclass[i].getBrand())
97              {
98                  found=1;
99                  pos=i;
100                 break;
101             }
102         }
103
104
105     }
106     if (answer==2)
107     {
108         found=0;
109         cout<<"Enter car model: ";
```

```
            getline(cin,searchedModel);
            for(i=0; i<MAXCARS; i++)
            {
                if(searchedModel==myclass[i].getModel())
                {
                    found=1;
                    pos=i;
                    break;
                }
            }

        }
        if (answer==3)
        {
            found=0;
            cout<<"Enter plate number: ";
            getline(cin,searchedPlate);
            for(i=0; i<MAXCARS; i++)
            {
                if(searchedPlate==myclass[i].getPlate())
                {
                    found=1;
                    pos=i;
                    break;
                }
            }

        }
        if (answer==4)
        {
            found=0;
            cout<<"Enter car owner: ";
            getline(cin,searchedOwner);
            for(i=0; i<MAXCARS; i++)
            {
                if(searchedOwner==myclass[i].getOwner())
```

```
            if(searchedOwner==myclass[i].getOwner())
            {
                found=1;
                pos=i;
                break;
            }
        }

    }
    if(found)
    {
        cout<<endl<<"Car found! ";
        cout<<myclass[pos].getBrand()<<" "<<myclass[pos].getModel()<<", owned by "<<myclass[pos].getOwner()<<", plate number "<<myclass[pos].getPlate()<<endl;
    }
    else cout<<"Car could not be found."<<endl;

}

int main()
{
    const int MAXCARS=1;
    car* myclass= new car[MAXCARS];
    myclass = enterData(myclass, MAXCARS);
    cout<<"You have saved: "<<endl;
    for(int i=0; i<MAXCARS; i++)
    {
        cout<<i+1<<". "<<myclass[i].getBrand()<<" "<<myclass[i].getModel()<<", owned by "<<myclass[i].getOwner()<<", plate number "<<myclass[i].getPlate()<<endl;
    }
    search(myclass,MAXCARS);
    cout<<endl<<endl;

    return 0;
}
```

Same program, but destined for use in a supermarket

```
#include <iostream>
using namespace std;
int answer, pos, found;
string name;
class product
{
private:
    string brand;
    string type;
    float price;
    int stock;
public:
    product()
    {
        brand="";
        type="";
        price=0.0;
        stock=0;
    }
    void setBrand(string b)
    {
        brand=b;
    }
    void setType(string t)
    {
        type=t;
```

```
27      }
28      void setPrice(float p)
29      {
30          price=p;
31      }
32      void setStock(int s)
33      {
34          stock=s;
35      }
36      string getBrand()
37      {
38          return brand;
39      }
40      string getType()
41      {
42          return type;
43      }
44      float getPrice()
45      {
46          return price;
47      }
48      int getStock()
49      {
50          return stock;
51      }
```

```
52
53      };
54      int main()
55      {
56          const int MAXPRODUCTS=3;
57          product inventory[MAXPRODUCTS];
58          string tbrand="", ttype="";
59          float tprice=0.0;
60          int tstock=0;
61          for(int i=0; i<MAXPRODUCTS; i++)
62          {
63              cout<<"Product "<<i+1<<endl;
64              cout<<"Brand: ";
65              getline(cin,tbrand);
66              inventory[i].setBrand(tbrand);
67              cout<<"Type: ";
68              getline(cin,ttype);
69              inventory[i].setType(ttype);
70              cout<<"Price: ";
71              cin>>tprice;
72              inventory[i].setPrice(tprice);
73              cout<<"Stock: ";
74              cin>>tstock;
75              inventory[i].setStock(tstock);
76
77          }
78          cout<<endl<<endl;
```

```
    cout<<"----------YOU HAVE SAVED----------"<<endl;
    for(int i=0; i<MAXPRODUCTS; i++)
    {
        cout<<"Product "<<i+1<<endl;
        cout<<"Brand: "<<inventory[i].getBrand()<<" || "<<"Type: "<<inventory[i].getType()<<" || Price: "<<inventory[i].getPrice()<<" || Stock remaining: "<<inventory[i].getStock()<<endl;
    }
    do
    {
        cout<<endl<<endl;
        cout<<"Are you searching by brand or by type? (0 to exit) "<<endl;
        cout<<"Enter 1 for Name or 2 for type ";
        cin>>answer;
        if(answer==1)
        {
            found=0;
            cout<<"Enter product brand: ";
            cin>>name;
            for(int i=0; i<MAXPRODUCTS; i++)
            {
                if(name==inventory[i].getBrand())
                {
                    found=1;
                    pos=i;
                    break;
                }
            }
            if(found)
            {
                cout<<endl;
                cout<<"Product found!"<<endl;
                cout<<"Brand: "<<inventory[pos].getBrand()<<" || "<<"Type: "<<inventory[pos].getType()<<" || Price: "<<inventory[pos].getPrice()<<" || Stock remaining: "<<inventory[pos].getStock()<<endl;
            }
            else cout<<"Product cannot be found";
        }
        if(answer==2)
        {
            found=0;
            cout<<"Enter product type: ";
            cin>>name;
            for(int i=0; i<MAXPRODUCTS; i++)
            {
                if(name==inventory[i].getType())
                {
                    found=1;
                    pos=i;
                    break;
                }
            }

            if(found)
            {
                cout<<endl;
                cout<<"Product found!"<<endl;
                cout<<"Brand: "<<inventory[pos].getBrand()<<" || "<<"Type: "<<inventory[pos].getType()<<" || Price: "<<inventory[pos].getPrice()<<" || Stock remaining: "<<inventory[pos].getStock()<<endl;
            }
            else cout<<"Product cannot be found";
        }


    }
    while(answer>0);

    return 0;
}
```

Photos from the mobility



