

2025-1-RO01-KA121-VET-000319082

4IT Erasmus+Project, 2025-1-RO01-KA121-VET-000319082

Beneficiary: "Ferdinand I" National College, Bacau

Sending organization: "Gheorghe Vranceanu" National College, Bacau

MOBILITY PORTFOLIO

Paphos, Cyprus

Mobility period 28.10.2025- 10.11.2025

Practice period: 29.10.2025-08.11.2025

RECEIVING COMPANY: Lazarous ICT Professionals

Practice tutor: Lazaros Lazarou

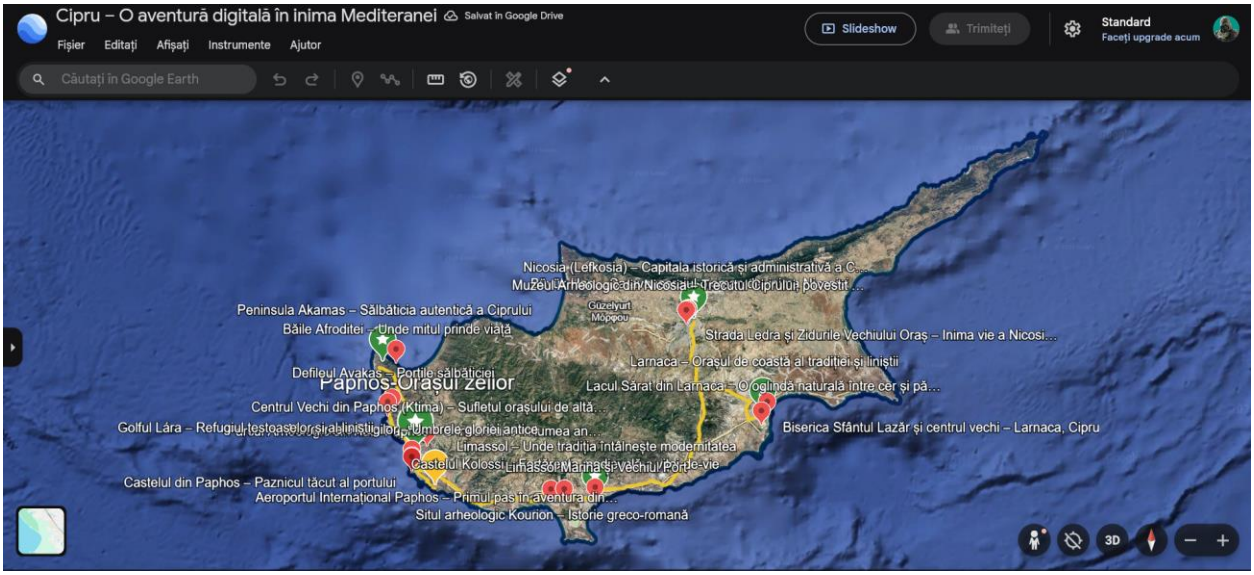
Participant: Varga David-Marian

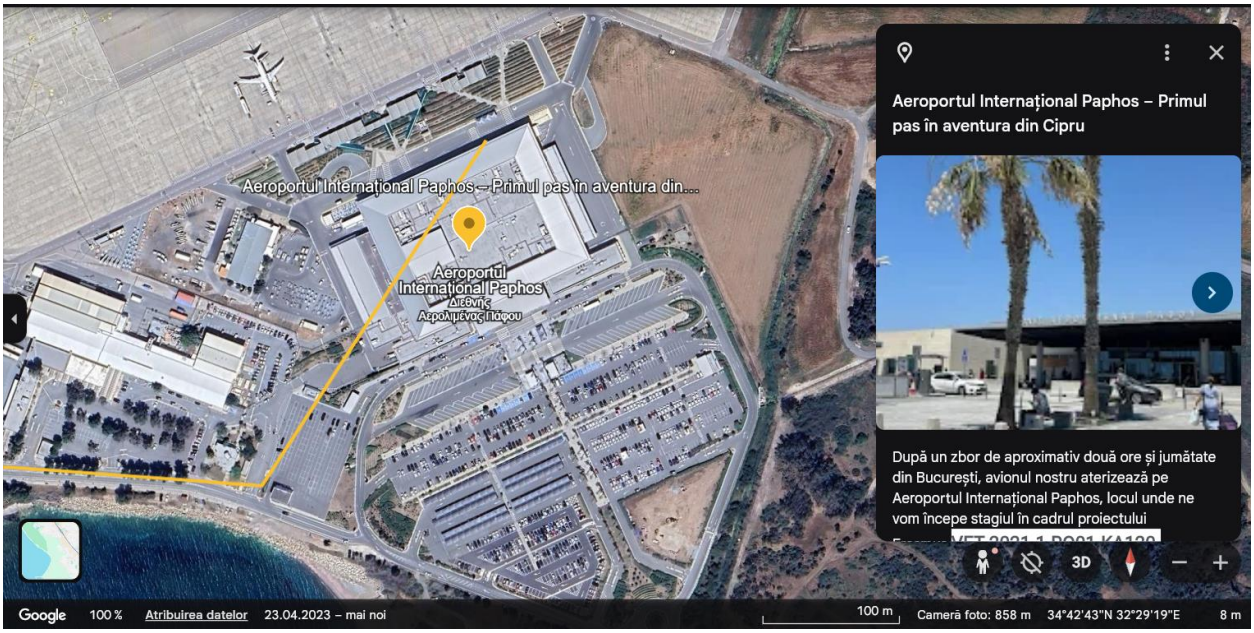
DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

2025-1-RO01-KA121-VET-000319082

Proiectul pentru pregătire culturală:



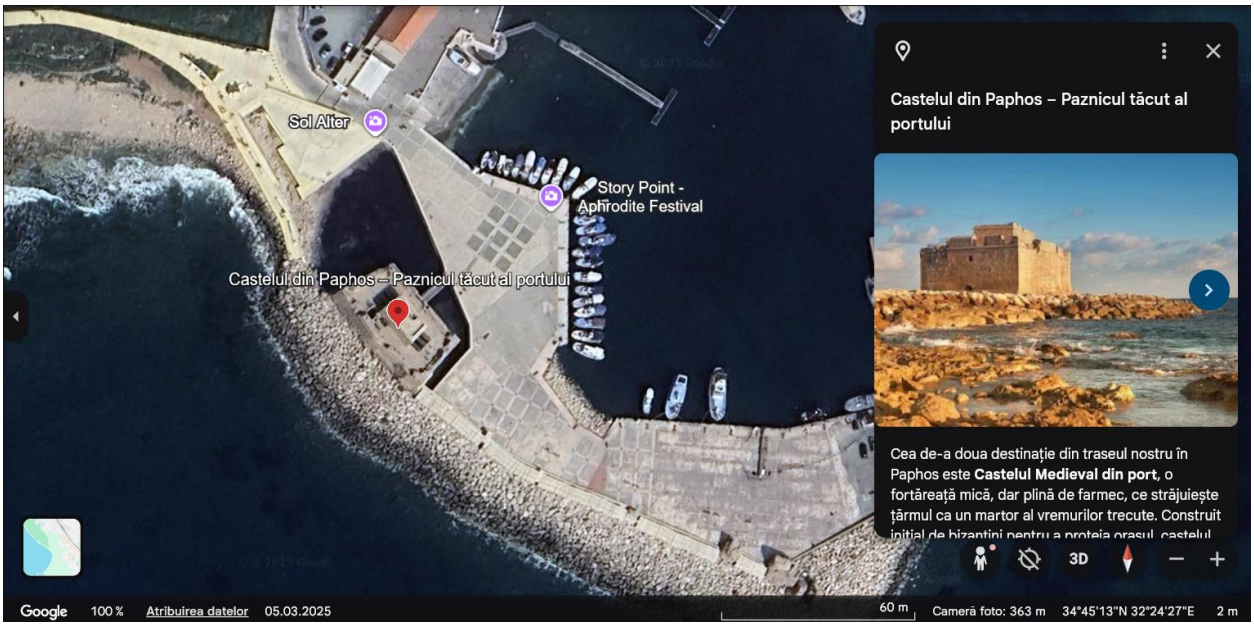


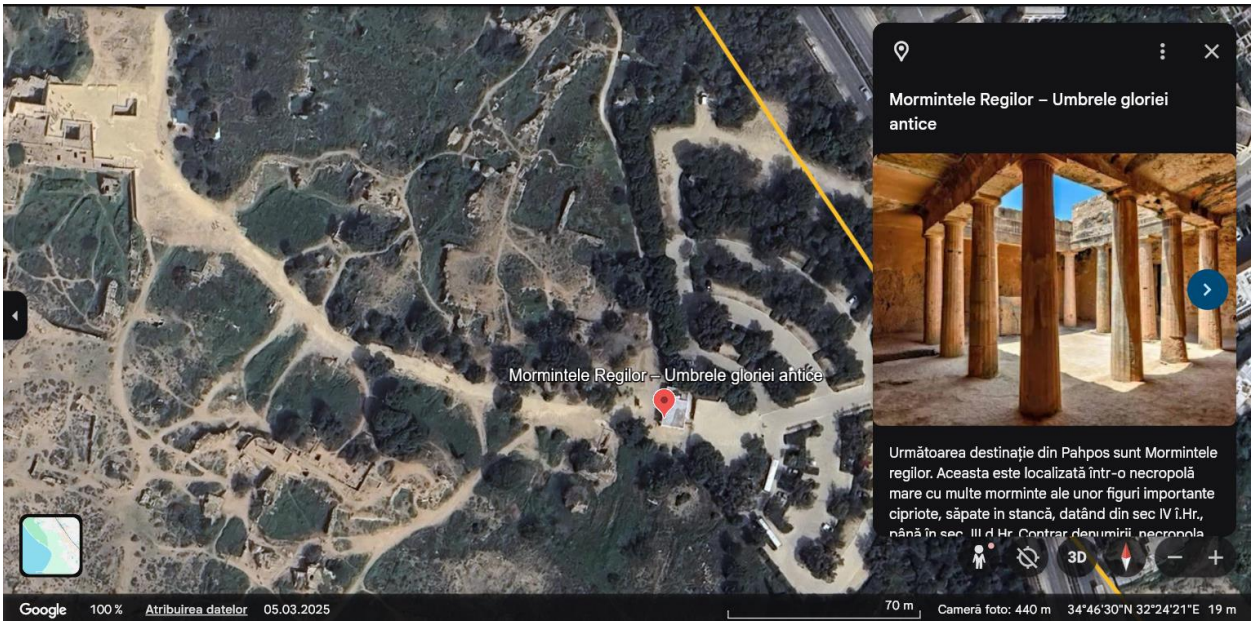
DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082



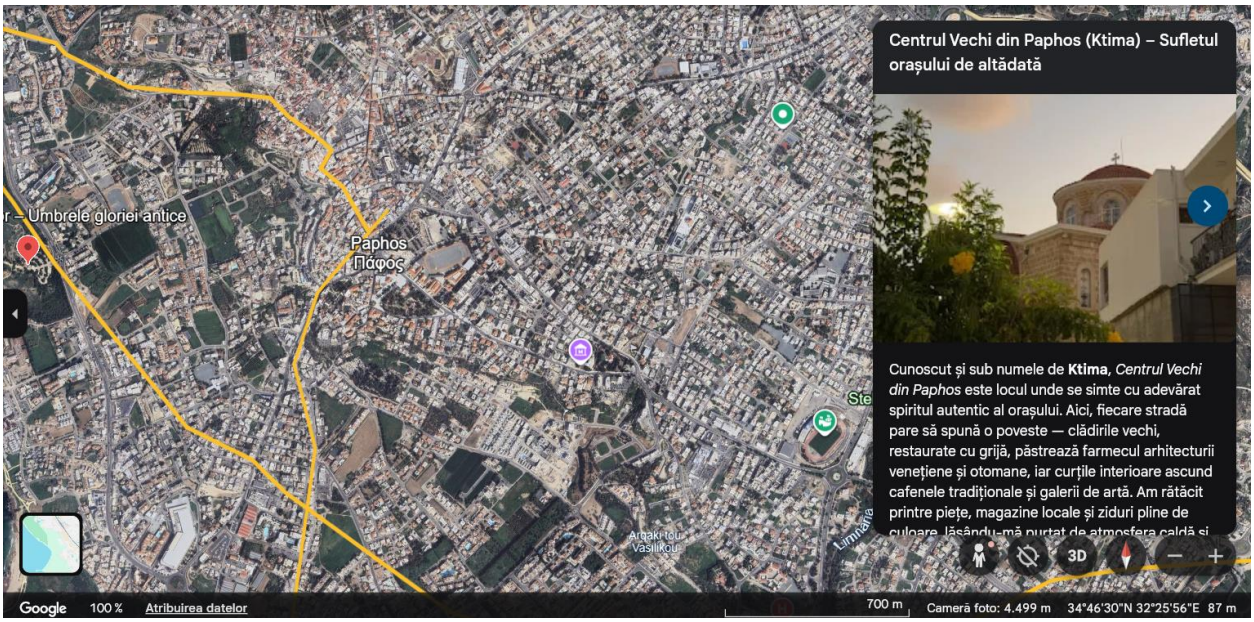


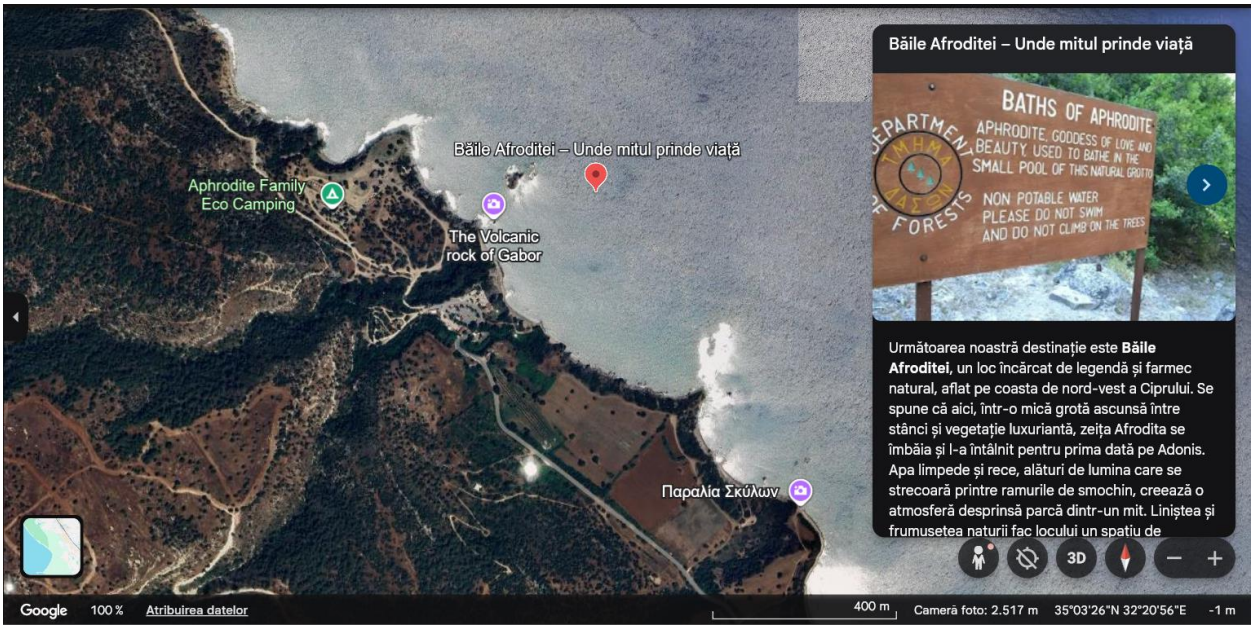
DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082



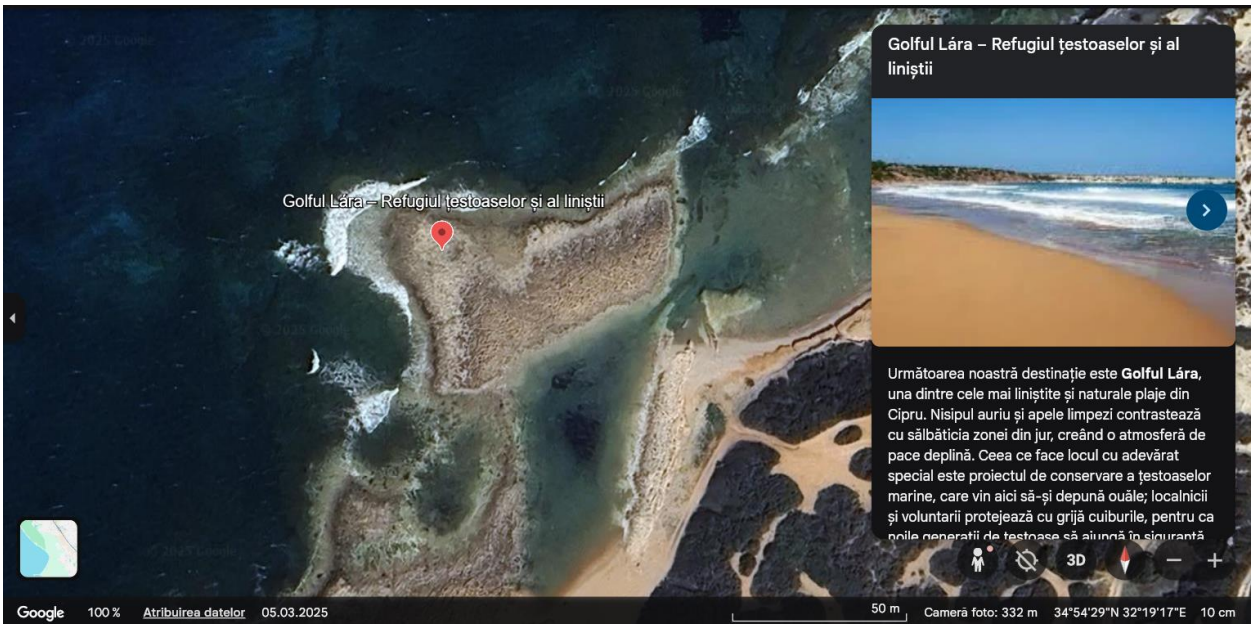


DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

2025-1-RO01-KA121-VET-000319082



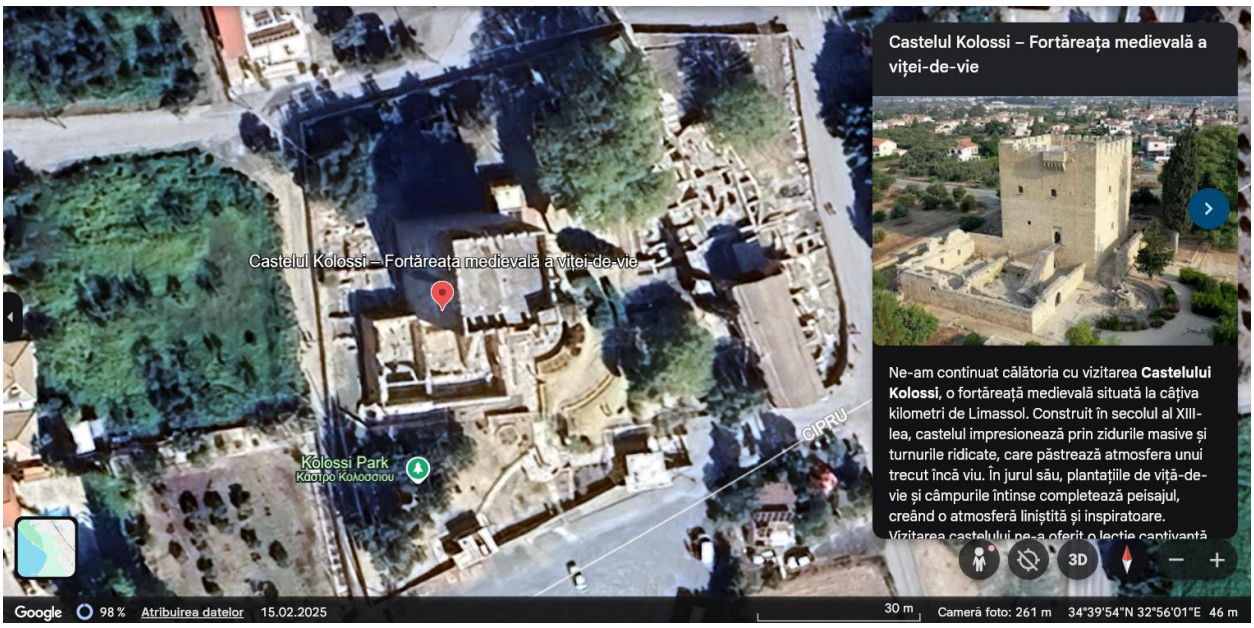


DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

2025-1-RO01-KA121-VET-000319082



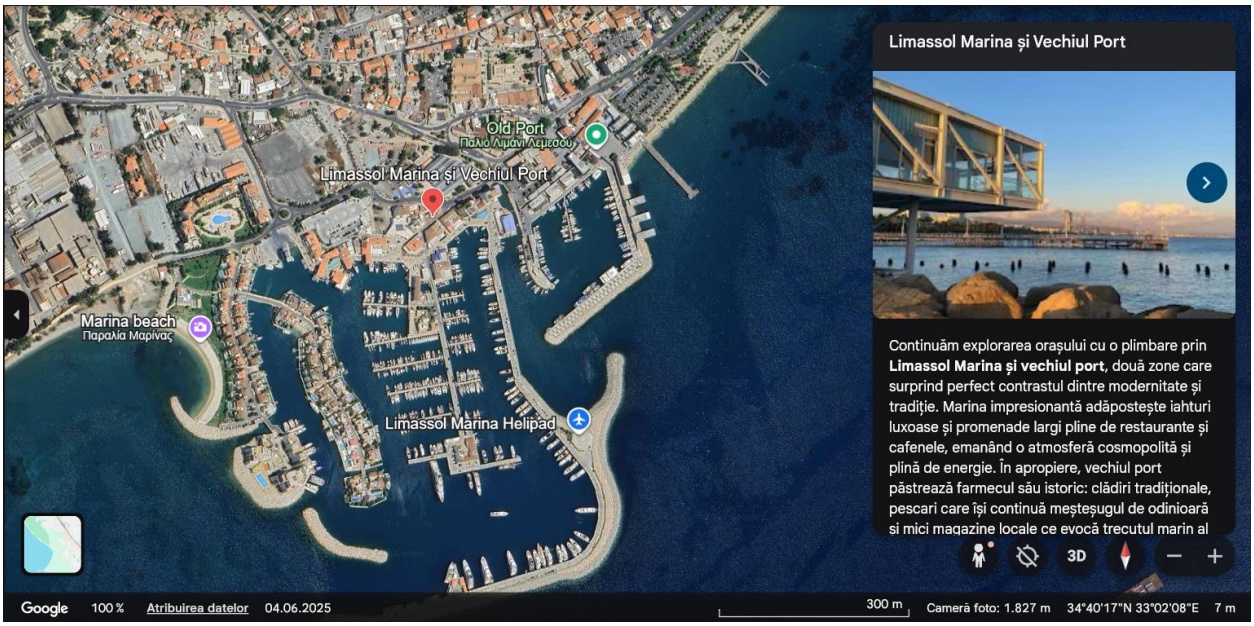


DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082



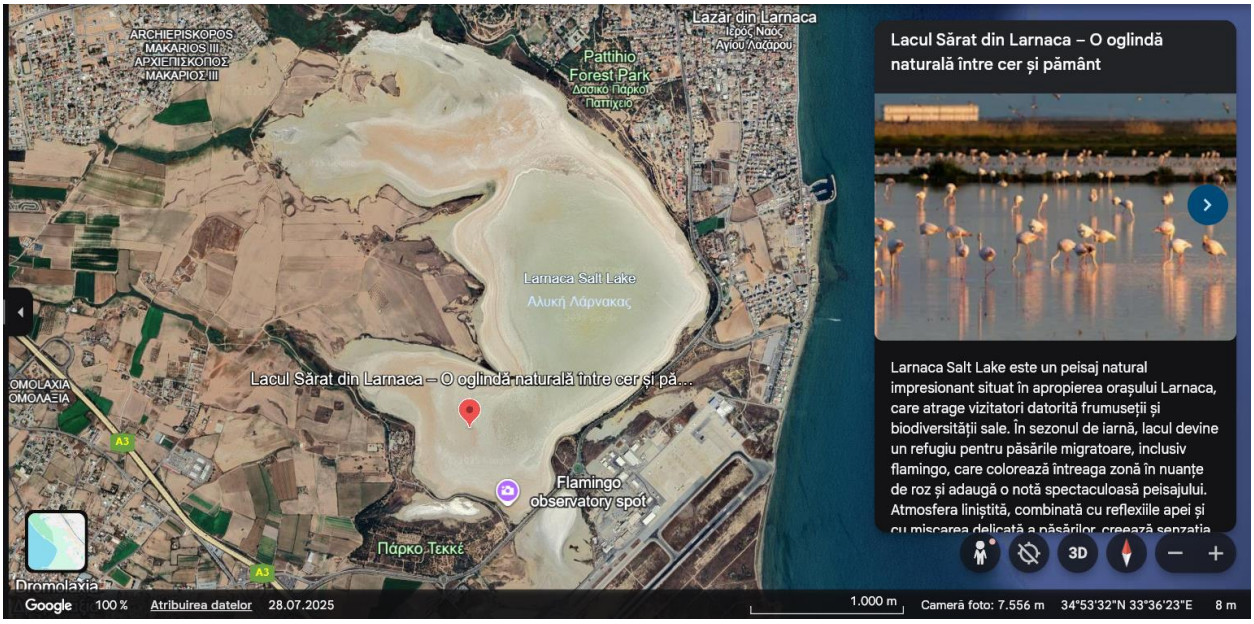


DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

2025-1-RO01-KA121-VET-000319082

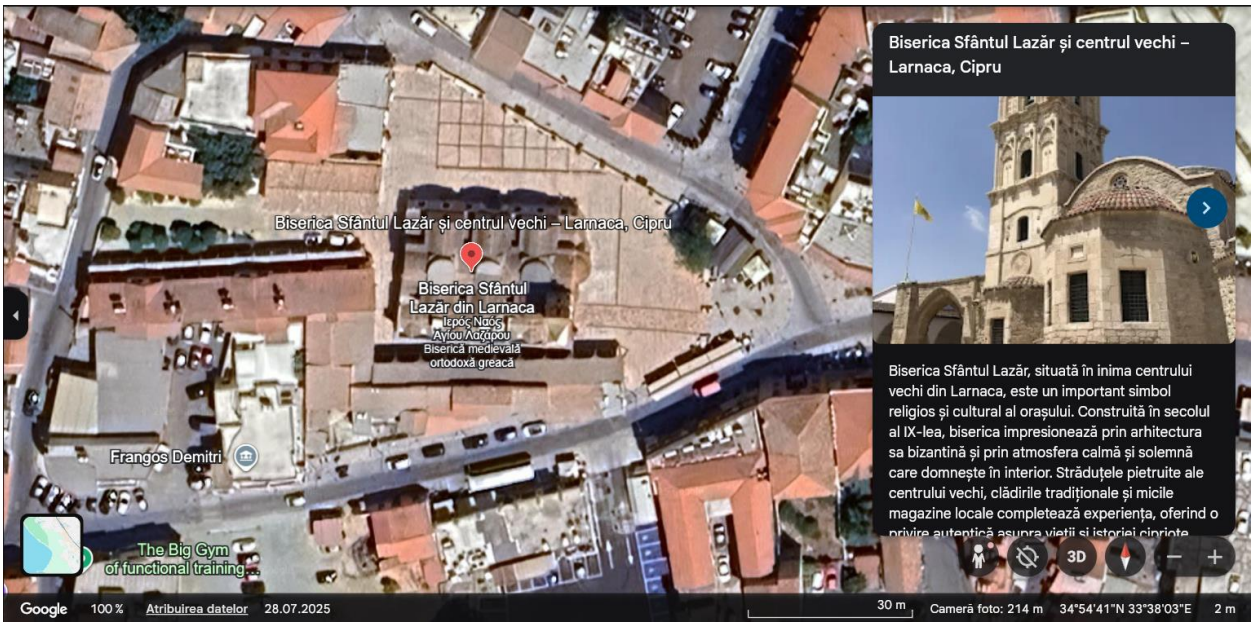




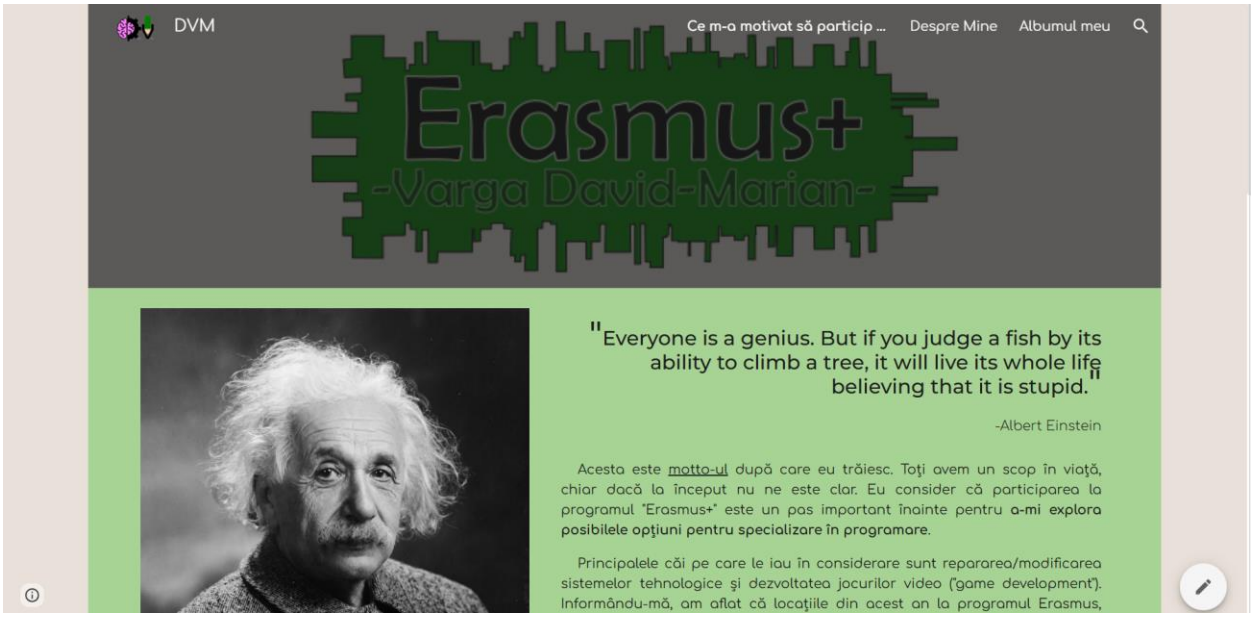
DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

2025-1-RO01-KA121-VET-000319082



Site-ul meu:



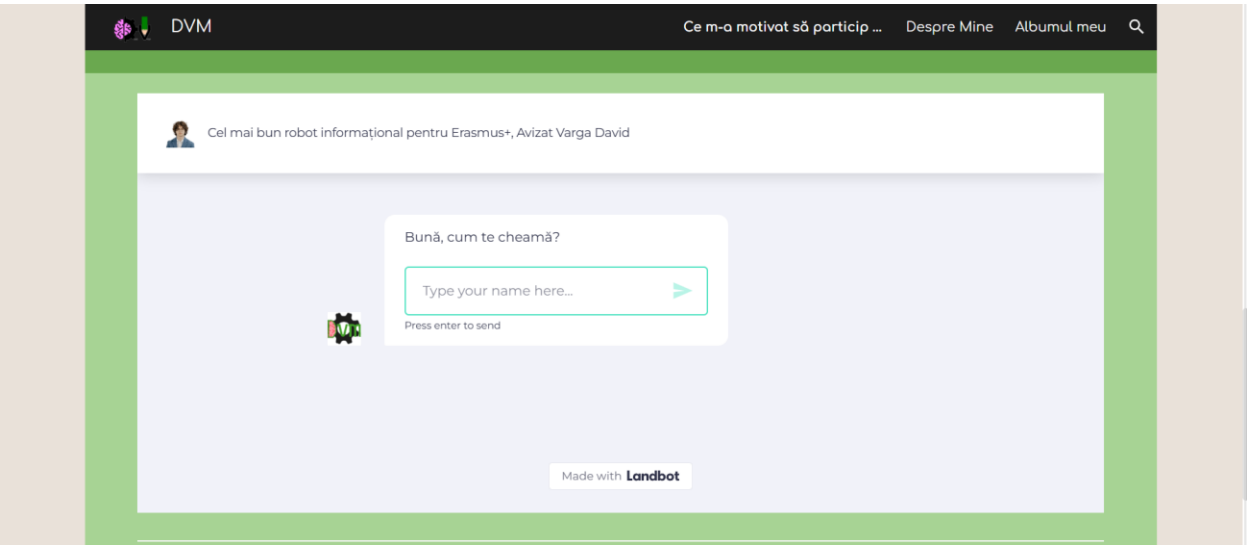
DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082



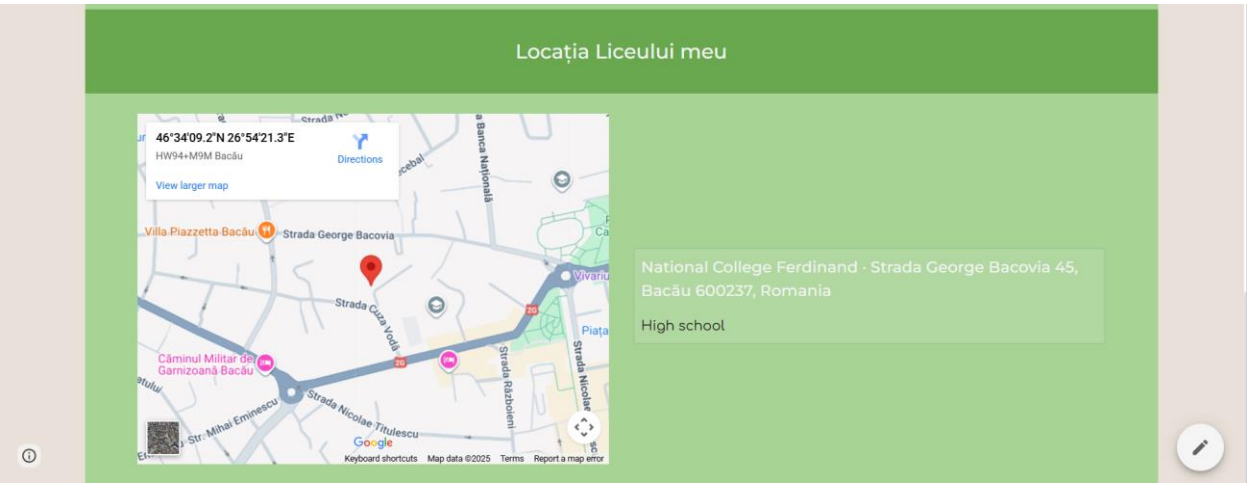


DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082



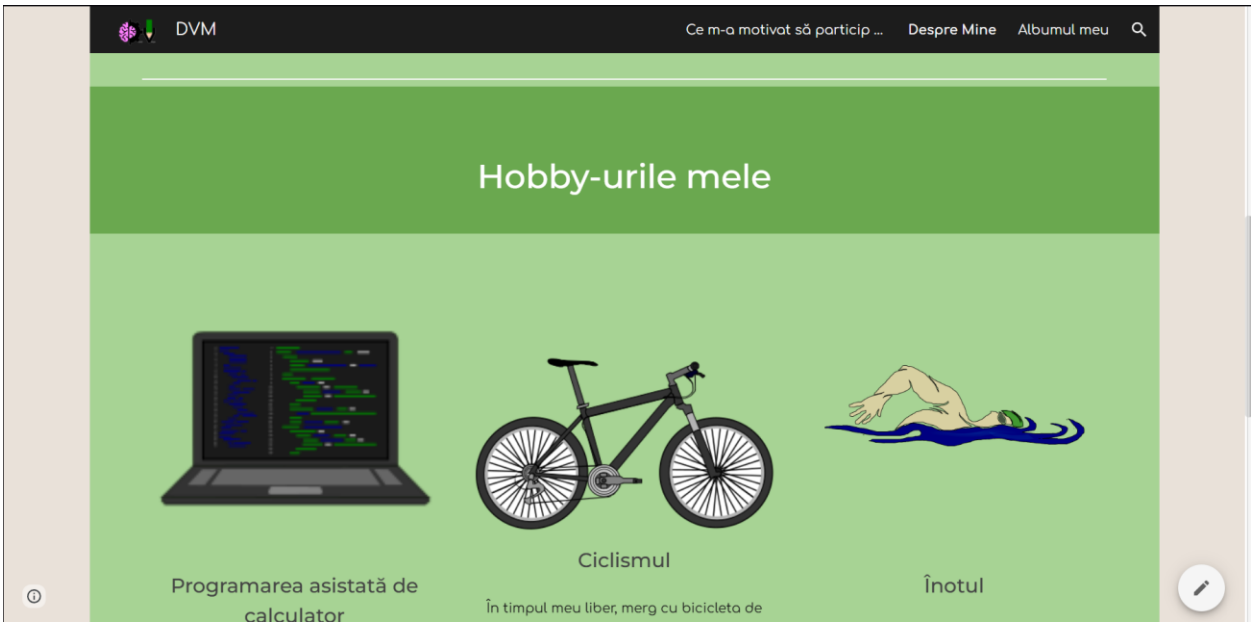


DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082

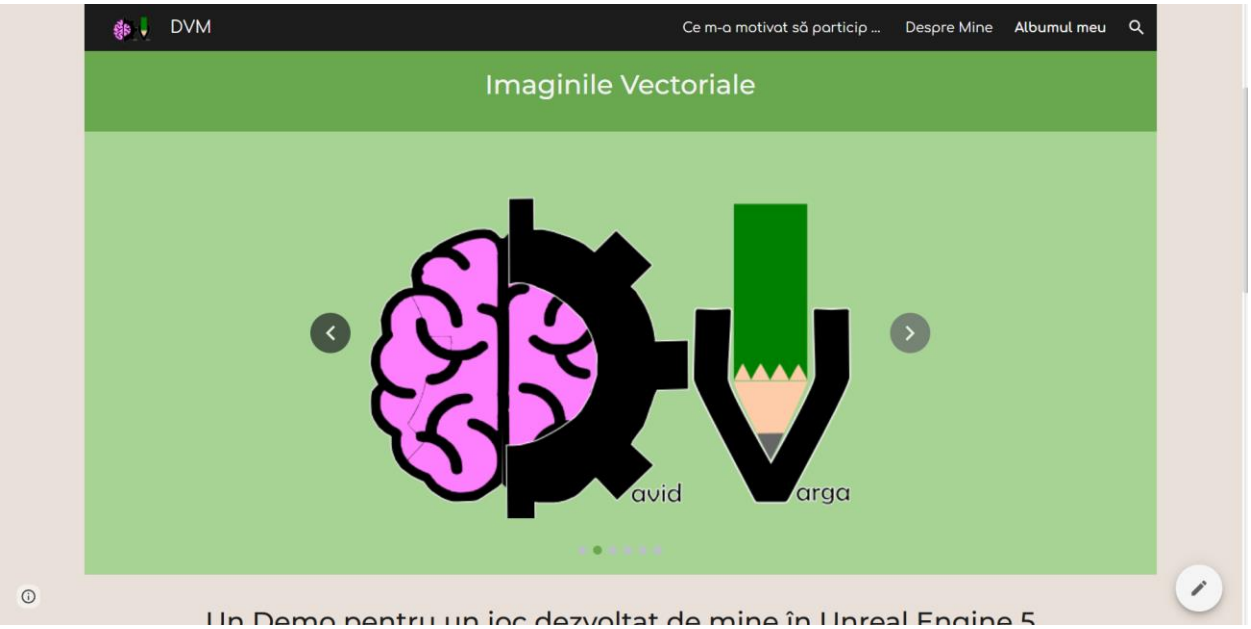




DISCLAIMER: The European Commission’s support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

2025-1-RO01-KA121-VET-000319082





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.

Co-funded by the Erasmus+ Programme of the European Union
Erasmus+
2025-1-RO01-KA121-VET-000319082

Lecțiile și temele pe care le-am făcut în cadrul proiectului, în ordine:

1)C++

În acest program se folosesc pointeri pentru a aloca spatiu unui vector și a-l crește dinamic, utilizând șiun al doilea vector care reține valorile celui principal, întrucât crescându-i marimea maximă, îi ștergem și valorile, iar apoi i le reatribuim vectorului original.

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.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082

C: > Users > Lenovo > AppData > Local > Microsoft > Windows > INetCache > IE > R4RK3DYB > GROWING_ARRAYS[1].cpp

```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* intarr;                                      //a is a pointer to an integer. This is going to be our main array to save integers
    int* temp;                                      //temp is a pointer to an integer and is going to be a temporary array. We need it in order to be able to GROW ou
    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 t

    //--- 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
    intarr = new int[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 >> intarr[n];                               //save the number we just read into our dynamic array
        cout<<"Number saved!"<<endl;
        n++;                                        //after saving the number we have to increase the counter
        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): ";//LET THE USER KNOW THAT THE ARRAY IS OUT OF SPACE AND ASK HOW MANY
            cin>>tmpN;                              //read how many more number the user needs to input
            if (tmpN > 0){                          //ERROR CHECKING: If the number the user gave us is not zero OR below zero we grow the array
                ARRAYSIZE = ARRAYSIZE + tmpN;       // Add to the previous size the added size the user gave us
                temp = new int[ARRAYSIZE];          // create new temporary bigger array. We need it to copy the contents of the previous array so we don't lose them
                for (int i=0; i<n; i++) {           //We use the for loop to copy the contents from the previous array into the temporary array(which has a bigger siz
                    temp[i] = intarr[i];                // copy values to new array.
                }
                delete [] intarr;                       // free old array memory.
                intarr = temp;                          // now a points to new array.
```

C: > Users > Lenovo > AppData > Local > Microsoft > Windows > INetCache > IE > R4RK3DYB > GROWING_ARRAYS[1].cpp

```cpp
int main(){
    do{
        if (n >= ARRAYSIZE){                        //if the new counter is equal to the ARRAYSIZE we need to GROW THE ARRAY
            if (tmpN > 0){                          //ERROR CHECKING: If the number the user gave us is not zero OR below zero we grow the array
                intarr = temp;                          // now a points to new array.
            }
        }

    }while (n<ARRAYSIZE);


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

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

        cout<<intarr[i]<<endl;
    }
    delete [] intarr; //free the memory of intarr

    return 0;
}

/*Here you must not delete the temp because we set intarr to equal temp
        Remember that pointers are variables which actually save memory addresses.
        in this program, at the part where we grow our array we set intarr=temp.
        Therefore intarr and temp point to the same part of memory. Because of that you
        need to delete only pointer intarr, and the memory will be deleted for temp as well.
    */
```

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.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082

C: > Users > Lenovo > Desktop > > erasmus > erasmus1.cpp

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

int main () {
    int *exam1;
    int *exam2;
    int maxSize;
    int *temp1;
    int *temp2;
    int n=0;
    int tempN=0;

    cout << "How many Students would you like to add memory on? \n";
    cin >> maxSize;
    exam1= new int [maxSize];
    exam2 = new int [maxSize];

    do {
        cout << "Student " << n+1 <<endl;
        cin >> exam1[n]; cout << endl;
        cin >> exam2[n]; cout << endl;
        n++;
        if (n>= maxSize) {
            cout <<" Out of space! would you like to add more memory? (0 to exit)";
            cin >> tempN;
            if (tempN>0){
                maxSize = maxSize + tempN;
                temp1 = new int[maxSize];
                temp2 = new int[maxSize];
                for (int i=0; i<maxSize; i++) {
                    temp1[i] = exam1[i];
                    temp2[i] = exam2[i];
                }
                delete [] exam1;
                delete [] exam2;
```

erasmus1.cpp | GROWING_ARRAYS[1].cpp | erasmus2.cpp | objectsIntro[1].cpp | erasmus3.cpp | erasmus4.cpp

C: > Users > Lenovo > Desktop > > erasmus > erasmus1.cpp

```cpp
int main () {
    do {
        if (n>= maxSize) {
            if (tempN>0){
                delete [] exam2;
                exam1=temp1;
                exam2=temp2;
            }
        }


    } while (n < maxSize);


    cout << "Here is all the information stored:" << endl;

    for (int i=0; i<maxSize; i++)
        cout << "Student " << i+1 << endl << exam1[i] << endl << exam2[i] <<endl << "average: " << (exam1[i]+exam2[i])/2 <<endl;


    return 0;
}
```

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.

2025-1-RO01-KA121-VET-000319082

În acest program utilizăm creșterea dinamică a unui vector menționată mai sus și obiecte, care sunt create cu ajutorul funcției “class”. Astfel, creem un vector ce conține obiecte.

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.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082

C: > Users > Lenovo > AppData > Local > Microsoft > Windows > INetCache > IE > FFPE2GHK > objectsIntro[1].cpp

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

class student{
    private:
        string name;
        int age;
        float ex1,ex2;

    public:
        void setName(string n){name=n;}
        void setAge(int a){age=a;}
        void setEx1(float e1){ex1=e1;}
        void setEx2(float e2){ex2=e2;}

        string getName(){return name;}
        int getAge(){return age;}
        float getEx1(){return ex1;}
        float getEx2(){return ex2;}
};

int main(){

    student st;
    student myClass[5];
    string name;
    int age;
    float e1,e2;

    st.setName("Lazaros");
    st.setAge(49);
    st.setEx1(7);
    st.setEx2(8);

    cout<<"Name: "<<st.getName()<<endl;
```

C: > Users > Lenovo > AppData > Local > Microsoft > Windows > INetCache > IE > FFPE2GHK > objectsIntro[1].cpp

```cpp
int main(){
    cout<<"Name: "<<st.getName()<<endl;
    cout<<"Age: "<<st.getAge()<<endl;
    cout<<"Exam 1: "<<st.getEx1()<<endl;
    cout<<"Exam 2: "<<st.getEx2()<<endl;
    cout<<"Avg: "<<(st.getEx1()+st.getEx2())/2<<endl;


    for (int i=0;i<5;i++){
        cout<<"Name: ";
        cin>>name;
        myClass[i].setName(name);
        cout<<"Age: ";
        cin>>age;
        myClass[i].setAge(age);
        cout<<"Exam1: ";
        cin>>e1;
        myClass[i].setEx1(e1);
        cout<<"Exam 2: ";
        cin>>e2;
        myClass[i].setEx2(e2);
    }
    for (int i=0;i<5;i++){
        cout<<myClass[i].getName()<<endl;
        cout<<myClass[i].getAge()<<endl;
        cout<<myClass[i].getEx1()<<endl;
        cout<<myClass[i].getEx2()<<endl;
    }
    return 0;
}
```

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.

Co-funded by the Erasmus+ Programme of the European Union

2025-1-RO01-KA121-VET-000319082

C: > Users > Lenovo > Desktop > > erasmus > erasmus2.cpp

```cpp
#include <iostream>
#include <math.h>
using namespace std;

/*
EXERCISE
write a program which takes input from the user
the information for cars.
you need to save the following information:
BRAND
MODEL
YEAR
HP
When you are done reading print your information on the screen


*/

class autovehicle {
    private:
        string brand;
        string model;
        int year;
        int hp;

    public:
        void setBrand (string b) {brand=b;}
        void setModel (string m) {model=m;}
        void setYear (int y) {year=y;}
        void setHP (int h) {hp=h;}

        string getBrand () {return brand;}
        string getModel () {return model;}
        int getYear () {return year;}
        int getHp () {return hp;}
```

C: > Users > Lenovo > Desktop > > erasmus > erasmus2.cpp

```cpp
class autovehicle {
    public:
        int getHp () {return hp;}

};

int main () {
    autovehicle car;
    autovehicle *myCars;
    autovehicle *temp;
    string brand, model;
    int year, hp, maxSize;
    int n = 0, tempN = 0;


    cout << "How many cars would you like to add to the list?";
    cin >> maxSize;
    myCars = new autovehicle [maxSize];

    do {
        cout << "Brand: ";
        cin >> brand;
        myCars[n].setBrand(brand);
        cout << "Model: ";
        cin >> model;
        myCars[n].setModel(model);
        cout << "Year: ";
        cin >> year;
        myCars[n].setYear(year);
        cout << "HorseP: ";
        cin >> hp;
        myCars[n].setHP(hp);
        cout << endl <<endl;
        n++;
        if (n>=maxSize) {
```

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.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082

C: > Users > Lenovo > Desktop > > erasmus > erasmus2.cpp

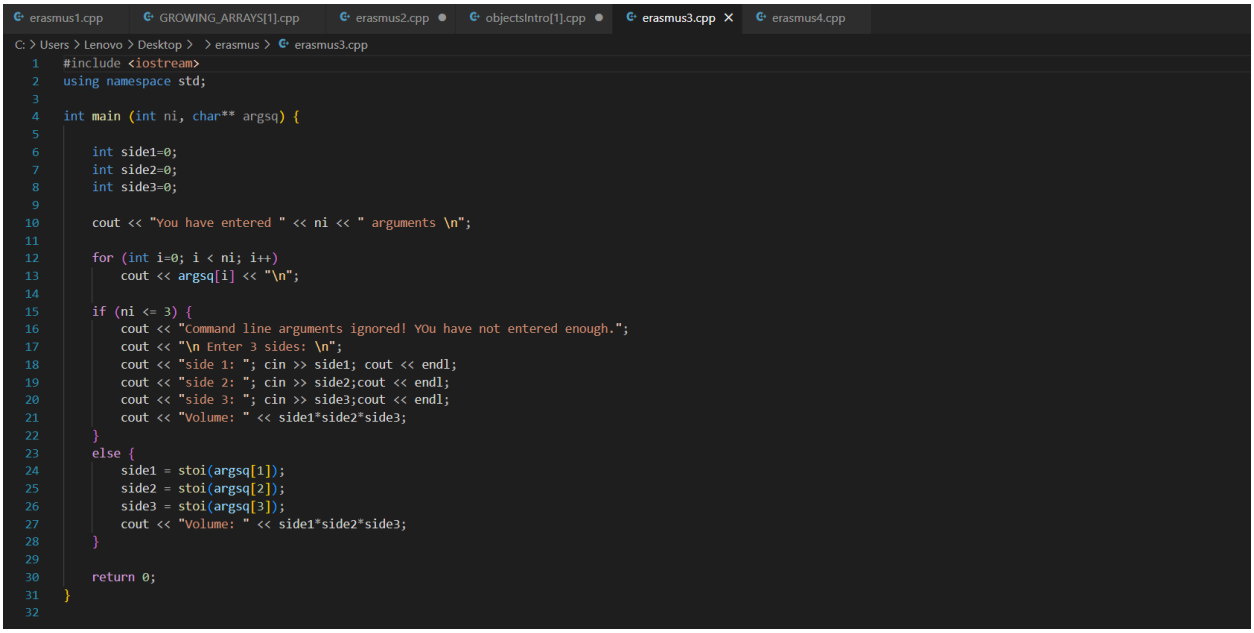
```
39  int main () {
52      do {
67          if (n>=maxSize) {
68              cout << " Out of space! would you like to add more memory? (0 to exit) \n";
69              cin >> tempN;
70              if (tempN > 0) {
71                  maxSize = maxSize + tempN;
72                  temp = new autovehicle [maxSize];
73                  for (int i = 0; i < n; i++) {
74                      temp[i] = myCars[i];
75                  }
76                  delete [] myCars;
77                  myCars = temp;
78              }
79          }
80      } while (n<maxSize);
81
82      cout << "Here is all the memory storedon the cars: \n";
83      for (int i=0; i<maxSize; i++) {
84          cout << myCars[i].getBrand() <<endl;
85          cout << myCars[i].getModel() <<endl;
86          cout << myCars[i].getYear() <<endl;
87          cout << myCars[i].getHp() <<endl <<endl;
88      }
89
90      return 0;
91  }
92
```

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.

Co-funded by the Erasmus+ Programme of the European Union

2025-1-RO01-KA121-VET-000319082

Aici am folosit “command line arguements” pentru a rula programul direct din CMD, oferindu-i tot acolo cu ce valori să opereze.



```
erasmus1.cpp | GROWING_ARRAYS[1].cpp | erasmus2.cpp | objectsIntro[1].cpp | erasmus3.cpp | erasmus4.cpp
C: > Users > Lenovo > Desktop > > erasmus > erasmus3.cpp
#include <iostream>
using namespace std;

int main (int ni, char** argsq) {

    int side1=0;
    int side2=0;
    int side3=0;

    cout << "You have entered " << ni << " arguments \n";

    for (int i=0; i < ni; i++)
        cout << argsq[i] << "\n";

    if (ni <= 3) {
        cout << "Command line arguments ignored! YOu have not entered enough.";
        cout << "\n Enter 3 sides: \n";
        cout << "side 1: "; cin >> side1; cout << endl;
        cout << "side 2: "; cin >> side2;cout << endl;
        cout << "side 3: "; cin >> side3;cout << endl;
        cout << "Volume: " << side1*side2*side3;
    }
    else {
        side1 = stoi(argsq[1]);
        side2 = stoi(argsq[2]);
        side3 = stoi(argsq[3]);
        cout << "Volume: " << side1*side2*side3;
    }

    return 0;
}

```

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.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082

C: > Users > Lenovo > Desktop > > erasmus > erasmus4.cpp

```cpp
#include <iostream>
using namespace std;
/*
    Part 1
    -------
    create a program that reads the
    following information from the user:
    Patient name   string
    Patient Age    int
    Patient Phone  string
    Patient Medicine string
    you cill create an array of objects since a hospital
    has so many patients.
    Part 2
    ------
    Create a menu for your program in order to give the user options.
    Give the following options:
    1. Add a new user
    2. Edit a User
    3. Delete a User
    4. Search by Name
    5. Search by Medicine
    6. Print All data (you can use it for testing)
    0. Exit
    With the creation of the menu, you are now
    required to change your program in such a way
    that whatever choice the user makes you will
    call a function to accomplish the selected task.
    PART 3
    -------
    Write The Search Function for your program
*/

```

C: > Users > Lenovo > Desktop > > erasmus > erasmus4.cpp

```cpp
//We will use the patient class to create the data structure
//to save our patients information. The data structure is going to
//be a dynamic array of patients
class patient{
    private:
        string name;
        int age;
        string phone;
        string medicine;

    public:
        //All the following functions are used to
        //save information for the objects that we create in
        //the object array. All SetFunctions are used to assign
        //values into the objects.
        //All GetFunctions are used to retrieve the values
        //from the objects
        void setName(string n) {name=n;}
        void setAge(int a)     {age=a;}
        void setPhone(string p)  {phone=p;}
        void setMedicine(string m)  {medicine=m;}

        string getName()    {return name;}
        int getAge()        {return age;}
        string getPhone()   {return phone;}
        string getMedicine(){return medicine;}
};
//the addPatient function is used to
//add one patient into our array of patient objects
void addPatient(patient *cl,int index);
//growArray is a function to make the array bigger. it takes
//a patient pointer and the arraySize
patient *growArray(patient *cl,int arraySize,int index);
//printArray is a function to print all the data
//that are saved into the array.
```

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.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082

```
70  //that are saved into the array.
71  //it takes as parameters the clinic array and the index
72  void printArray(patient *clinic, int index);
73
74  int main(){
75      int ArraySize=1;//set the array size to 1 and then we can grow the array if needed
76      int index=0; //index is the last index that we are using in the array(the tail of the array)
77      patient *clinic, *tmp; //a pointer to a patient to create a dynamic array
78      clinic=new patient[ArraySize]; //allocate space for the array
79      int userChoice=1; //needed to read the choice the user is giving us.
80
81      while (userChoice >0){
82          //print the menu for the user
83          cout<<"1. Add a new user"<<endl;
84          cout<<"2. Edit a User"<<endl;
85          cout<<"3. Delete a User"<<endl;
86          cout<<"4. Search by Name"<<endl;
87          cout<<"5. Search by Medicine"<<endl;
88          cout<<"6. Print The Array"<<endl;
89          cout<<"0. Exit"<<endl;
90          cout<<"What would you like to do? ";
91          cin>>userChoice;
92          cout<<"------------------------------"<<endl;
93
94          if (userChoice == 1){
95              cout<<"Add a new Patient"<<endl<<"----------------"<<endl;
96              if (index>=ArraySize){
97                  ArraySize=ArraySize+1;//grow the array by 1 position
98                  clinic=growArray(clinic,ArraySize,index); //call function to grow array
99              }
100             addPatient(clinic,index);//call the function to add a patient
101             index++;//after we added a patient we must increment the index
102         }else if(userChoice==2){
103             cout<<"Edit a Patient"<<endl<<"----------------"<<endl;
104         }else if(userChoice==3){
```

C: > Users > Lenovo > Desktop > > erasmus > erasmus4.cpp

```
74  int main(){
81      while (userChoice >0){
104         }else if(userChoice==3){
105             cout<<"Delete a Patient"<<endl<<"----------------"<<endl;
106         }else if (userChoice==4){
107             cout<<"Search By Name"<<endl<<"----------------"<<endl;
108             string nameSearch {};
109             int isNSTrue =1;
110             while (isNSTrue >= 1){
111                 cin >>  nameSearch;
112                 cout << "\n Searching for results...\n";
113                 int searchNameCount=0;
114                 for (int i=0; i < index; i++) {
115                     if (nameSearch == clinic[i].getName()) {
116                         cout << clinic[i].getName() <<endl;
117                         cout << clinic[i].getAge()<<endl;
118                         cout << clinic[i].getPhone()<<endl;
119                         cout << clinic[i].getMedicine()<<endl;
120                         cout << endl << "---------------" <<endl;
121                         searchNameCount++;
122                     }
123                 }
124                 cout << "found " << searchNameCount << " results" << endl ;
125                 cout << "Would you like to perform another search? Type 0 to exit, any number to continue.";
126                 cin >> isNSTrue; cout << endl;
127             }
128         }else if(userChoice==5){
129             cout<<"Search By Medicine"<<endl<<"----------------"<<endl;
130             string MedSearch {};
131             int isMSTrue =1;
132             while (isNSTrue >= 1){
133                 cin >>  MedSearch;
134                 cout << "\n Searching for results...\n";
135                 int searchMedCount=0;
136                 for (int i=0; i < index; i++) {
```

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.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082

C: > Users > Lenovo > Desktop > > erasmus > erasmus4.cpp

```
74  int main(){
81      while (userChoice >0){
128         }else if(userChoice==5){
132             while (isNSTrue >= 1){
136                 for (int i=0; i < index; i++) {
137                     if (MedSearch == clinic[i].getMedicine()) {
138                         cout << clinic[i].getName() <<endl;
139                         cout << clinic[i].getAge()<<endl;
140                         cout << clinic[i].getPhone()<<endl;
141                         cout << clinic[i].getMedicine()<<endl;
142                         cout << endl << "---------------" <<endl;
143                         searchMedCount++;
144                     }
145                 }
146                 cout << "found " << searchMedCount << " results" << endl ;
147                 cout << "Would you like to perform another search? Type 0 to exit, any number to continue.";
148                 cin >> isMSTrue; cout << endl;
149             }
150         }else if(userChoice==6){
151             printArray(clinic,index);
152
153         }else if (userChoice==0){
154             cout<<"Thank you For using the program!"<<endl<<"-----------------"<<endl;
155         }else{
156             cout<<"Wrong Choice! Try again"<<endl;
157             userChoice=10;
158         }
159     }
160     return 0;
161 }
162 /*
163 function growArray
164 The function is used to grow the clinic array by 1.
165 it takes as parameters, the clinic array (it is a patient pointer),
166 the arraySize (which we incremented in the main function,
167 and the index which we will increment in the main function,
168 after we make the array bigger.
169 The function returns a patient pointer which is the new bigger array.
170 */
171 patient *growArray(patient *clinic,int ArraySize, int index){
172     patient *tmp;
173     tmp=new patient[ArraySize];
174     cout<<"Memory for tmp allocated.\nArraySize= "<<ArraySize<<endl;
175     cout<<"Index= "<<index<<endl;
176     //copy the data from the original to temp array
177     for (int i=0;i<index;i++){
178         tmp[i].setAge(clinic[i].getAge());
179         tmp[i].setMedicine(clinic[i].getMedicine());
180         tmp[i].setName(clinic[i].getName());
181         tmp[i].setPhone(clinic[i].getPhone());
182     }
183     delete [] clinic;//delete the array with the old size
184     return tmp; //return temp back to the main function
185 }
```

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.

Co-funded by the Erasmus+ Programme of the European Union
Erasmus+
2025-1-RO01-KA121-VET-000319082

C: > Users > Lenovo > Desktop > > erasmus > erasmus4.cpp

```cpp
/*
Function addPatient
-takes as parameters a patient pointer (the dynamic array)
 the index, which is the next available position to save the data
 and the arraySize because we need to check if the array is full
-the function asks the user about the patient information and saves the data
 from the keyboard into the dynamic array, at position index.
-The function will check if the array is full and grow the array dynamically.
*/
void addPatient(patient *cl,int index){

    string name,phone,med;
    int age;

    cout<<"Name: ";
    cin>>name;
    cl[index].setName(name);
    cout<<"Age: ";
    cin>>age;
    cl[index].setAge(age);
    cout<<"phone: ";
    cin>>phone;
    cl[index].setPhone(phone);
    cout<<"Medicine Given: ";
    cin>>med;
    cl[index].setMedicine(med);
    cout<<"Patient Succesfully Added!"<<endl;

}
}
/*
function printArray(patient*,int)
takes parameters a patient pointer(the clinic array)
and an integer (the index)
once it is called it prints from zero to index the data that are
saved on the array.
*/
void printArray(patient *clinic, int index){
    cout<<"PRINTING ALL DATA"<<endl<<"-----------------"<<endl;
    for (int i=0;i<index;i++){
        cout<<"----- Patient "<<i+1<<"-----"<<endl;
        cout<<clinic[i].getName()<<endl;
        cout<<clinic[i].getAge()<<endl;
        cout<<clinic[i].getPhone()<<endl;
        cout<<clinic[i].getMedicine()<<endl;
    }

}

```

2)Arduino:

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.

2025-1-RO01-KA121-VET-000319082

a)Programare

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.

Co-funded by the Erasmus+ Programme of the European Union

2025-1-RO01-KA121-VET-000319082

sketch_nov4a | Arduino IDE 2.3.6

File Edit Sketch Tools Help

Select Board

sketch_nov4a.ino

```ino
void setup() {
  // put your setup code here, to run once:
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(pin, HIGH);
  delay(1000);
  digitalWrite(pin, LOW);
  delay(1000);
  digitalWrite(pin, HIGH);
  delay(1000);
  digitalWrite(pin, LOW);
  delay(1000);
  digitalWrite(pin, HIGH);
  delay(1000);
  digitalWrite(pin, LOW);
  delay(1000);

  digitalWrite(pin, HIGH);
  delay(2000);
  digitalWrite(pin, LOW);
  delay(2000);
  digitalWrite(pin, HIGH);
  delay(2000);
  digitalWrite(pin, LOW);
  delay(2000);
  digitalWrite(pin, HIGH);
  delay(2000);
  digitalWrite(pin, LOW);
  delay(2000);

  digitalWrite(pin, HIGH);
  delay(1000);
  digitalWrite(pin, LOW);
```

Distance_sensor.ino

```ino
#define PIN1 4
#define Pin2 7
#define echo 10
#define trg 9
#define buzzer 3
float distance, duration,time;
void setup() {
  Serial.begin(9600);
  pinMode(LED_BUILTIN,OUTPUT);
  pinMode(PIN1, OUTPUT);
  pinMode(Pin2, OUTPUT);
  pinMode(echo,INPUT);
  pinMode(trg,OUTPUT);
  pinMode(buzzer, OUTPUT);
}

void loop() {
   digitalWrite(trg, LOW);
  delayMicroseconds(2);
  digitalWrite(trg, HIGH);
  delayMicroseconds(10);
  digitalWrite(trg, LOW);

  duration = pulseIn(echo, HIGH);
  distance = (duration*.0343)/2;
  if(distance<90)
  time = distance*10;
  else time = distance;
  digitalWrite(PIN1,HIGH);
  delay(time);
  digitalWrite(PIN1,LOW);
  delay(time);
  if(distance<20)
  digitalWrite(Pin, HIGH);
  else digitalWrite(pin, LOW)

}
```

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.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082

```
const int trigPin = 9;
const int echoPin = 10;

float duration, distance;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

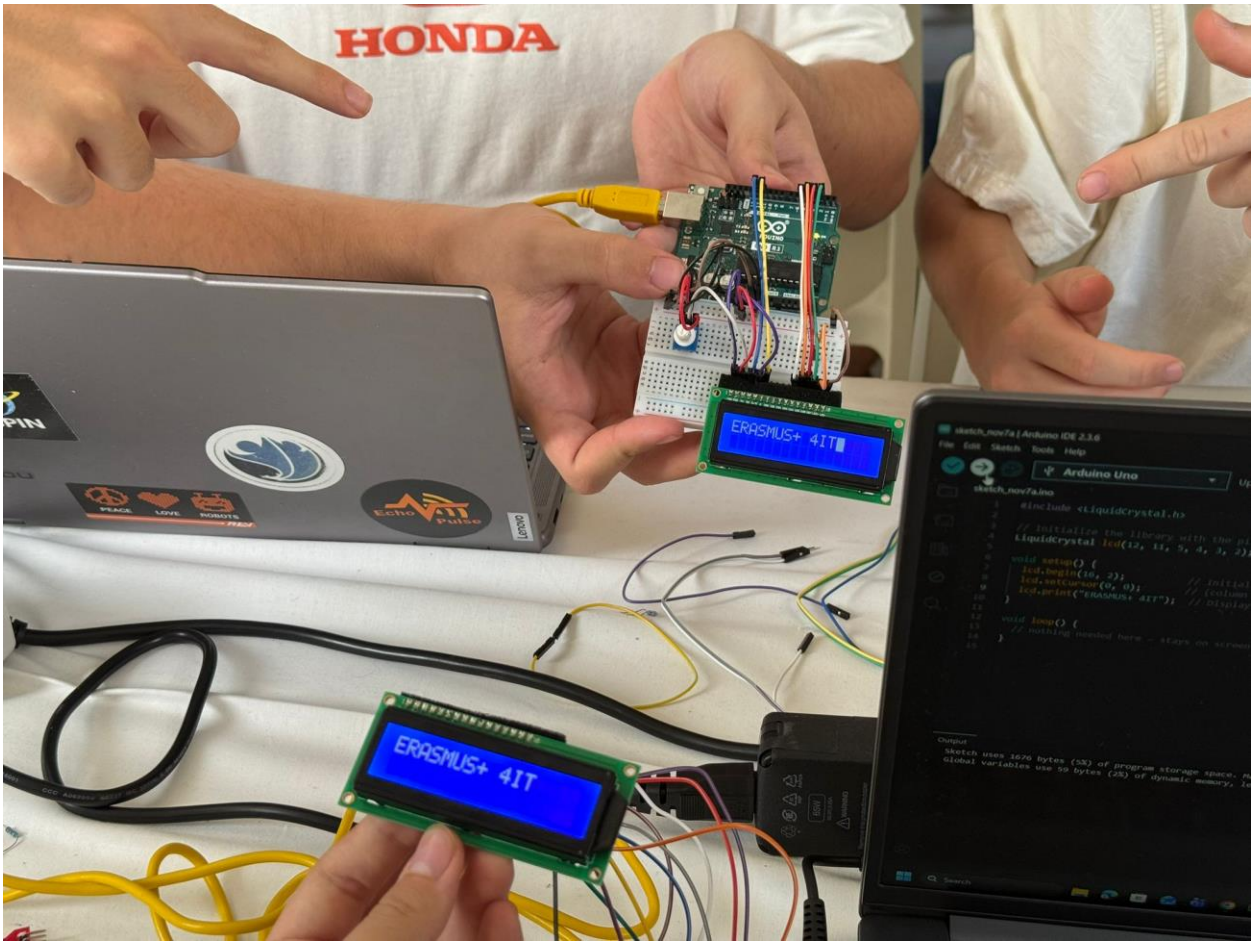
  duration = pulseIn(echoPin, HIGH);
  distance = (duration*.0343)/2;
  Serial.print("Distance: ");
  Serial.println(distance);
  delay(100);
}

```

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.

Co-funded by the Erasmus+ Programme of the European Union
Erasmus+
2025-1-RO01-KA121-VET-000319082

b)Circuite

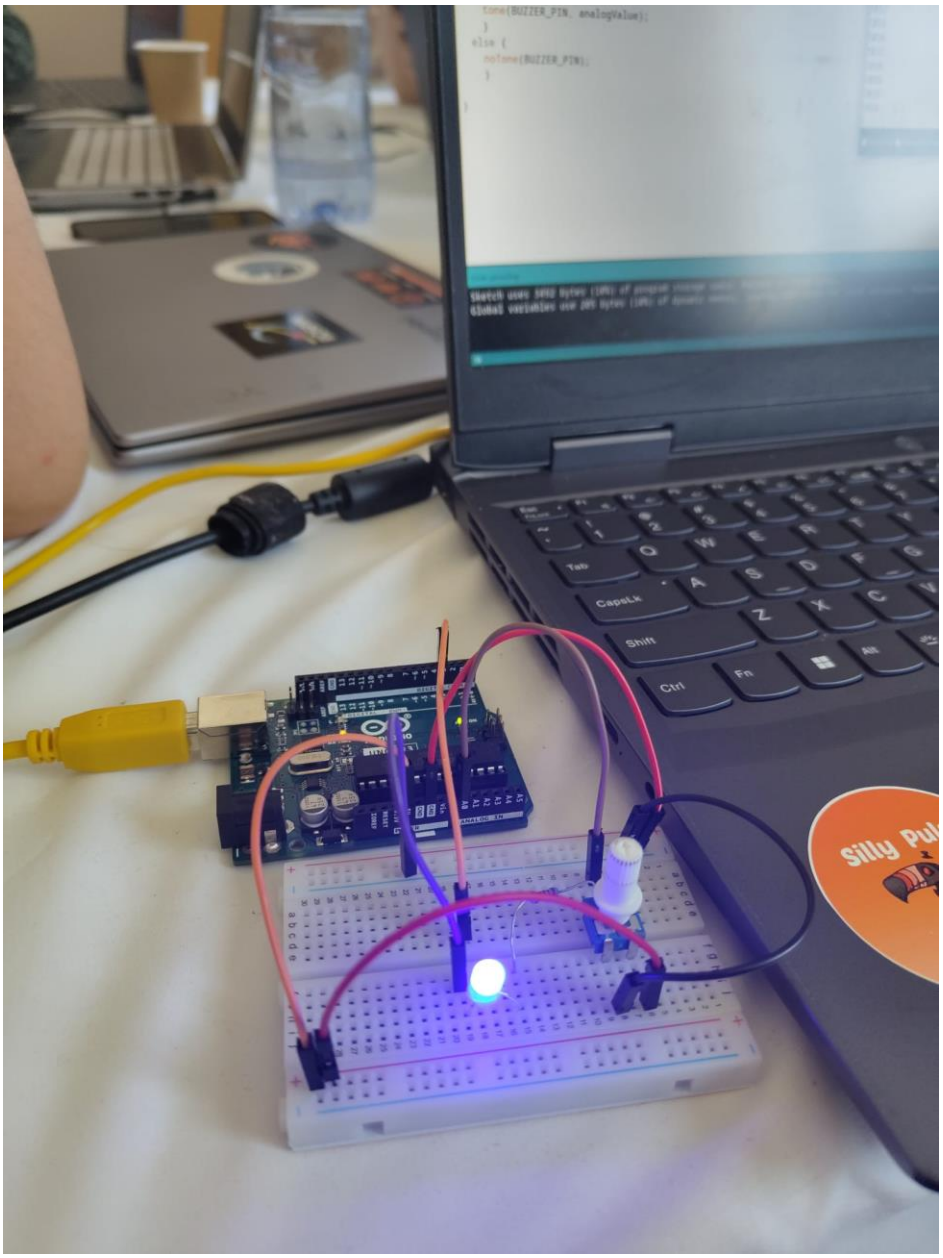


DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082

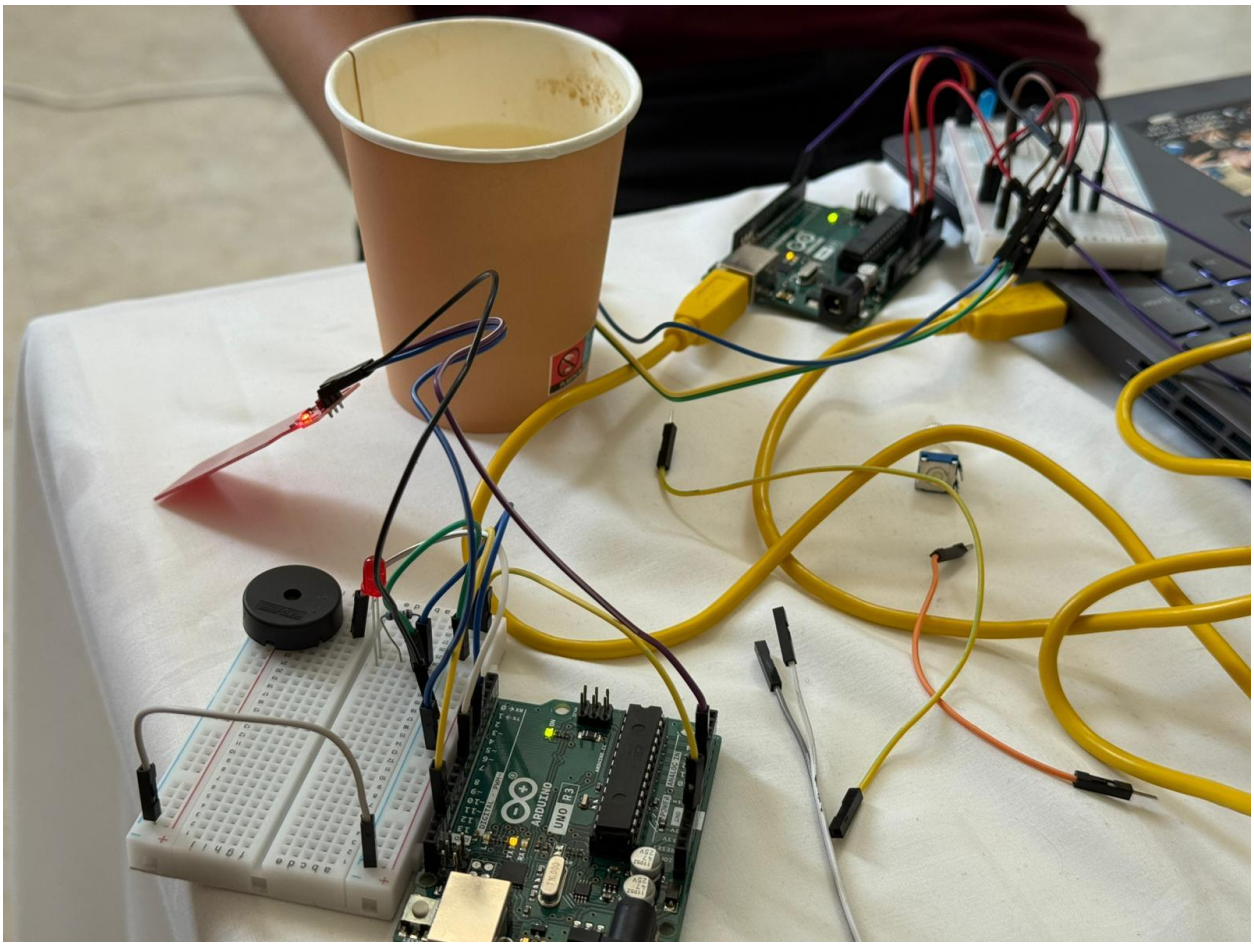


DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082



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.