



REPUBLIKA E SHQIPËRISË
MINISTRIA E ARSIMIT
DHE SPORTIT
QENDRA E SHËRBIMEVE ARSIMORE

OLIMPIADA KOMBËTARE E INFORMATIKËS
NË ARSIMIN E MESËM TË LARTË

Faza e dytë

Klasa 10

18 janar 2025

Udhëzime për nxënësin:

- Olimpiada fillon në orën 10:00 dhe mbaron në orën 13:00.
- Testi përmban 5 pyetje.
- Për të zgjidhur secilin ushtrim nxënësi mund të përdorë gjuhën e programimit C++.

Për përdorim nga komisioni i vlerësimit

Pyetja	1	2	3	4	5
	7 pikë	8 pikë	12 pikë	13 pikë	10 pikë
Pikët e fituara					

Totali i pikëve të fituara

KOMISIONI I VLERËSIMIT

1.....

2.....

1. Shkruani një program që konverton një numër të dhënë nga sistemi dhjetor në sistemin binar dhe e shfaq atë si një varg karakteresh (string). **7 pikë**

Zgjidhje:

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
string decimalToBinary(int n) {  
    string binary = "";  
    while (n > 0) {  
        binary = to_string(n % 2) + binary;  
        n /= 2;  
    }  
    return binary;  
}
```

```
int main() {  
    int num;  
    cout << "Shkruani një numër dhjetor: ";  
    cin >> num;  
    cout << "Numri në sistemin binar është: " << decimalToBinary(num) << endl;  
    return 0;  
}
```

2. Shkruani një program që implementon një sistem të thjeshtë të menaxhimit të shkollës, ku mund të regjistrohen të dhëna për 10 nxënës, të shfaqen informacionet për ta dhe të gjenden nxënësit me notat më të ulëta në dy lëndë të caktuara.

8 pikë

Zgjidhje:

```
#include <iostream>

#include <string>

using namespace std;

struct Student {
    string name;
    int mathGrade;
    int physicsGrade;
};

void findLowestGrades(Student students[], int size) {
    Student minMath = students[0];
    Student minPhysics = students[0];

    for (int i = 1; i < size; i++) {
        if (students[i].mathGrade < minMath.mathGrade)
            minMath = students[i];
        if (students[i].physicsGrade < minPhysics.physicsGrade)
            minPhysics = students[i];
    }

    cout << "Nxënësi me notën më të ulët në matematikë: " << minMath.name << endl;
    cout << "Nxënësi me notën më të ulët në fizikë: " << minPhysics.name << endl;
}

int main() {
    Student students[5] = {
```

```
{ "Student1", 80, 90 },  
{ "Student2", 70, 85 },  
{ "Student3", 75, 60 },  
{ "Student4", 65, 70 },  
{ "Student5", 90, 95 }  
};
```

```
findLowestGrades(students, 5);
```

```
return 0;
```

```
}
```

3. Shkruani një program që implementon një algoritëm të të gjitha permutimeve (kombinimeve) të mundshme për një string të dhënë dhe kontrollon nëse përmban një palindromë. Programi duhet të shfaqë palindromën nëse ka. **12 pikë**

Zgjidhje:

```
#include <iostream>
```

```
#include <algorithm>
```

```
using namespace std;
```

```
// Kontrolllo nëse një varg është palindromë
```

```
bool isPalindrome(string s) {
```

```
    int len = s.length();
```

```
    for (int i = 0; i < len / 2; i++) {
```

```
        if (s[i] != s[len - i - 1])
```

```
            return false;
```

```
    }
```

```
    return true;
```

```
}
```

```
int main() {  
    string input;  
    cout << "Shkruani një string: ";  
    cin >> input;  
  
    sort(input.begin(), input.end());  
  
    do {  
        if (isPalindrome(input)) {  
            cout << "Palindroma e gjetur: " << input << endl;  
            return 0;  
        }  
    } while (next_permutation(input.begin(), input.end()));  
  
    cout << "Nuk u gjet asnjë palindromë." << endl;  
    return 0;  
}
```

4. Një përdorues merr një hua me një normë interesi të caktuar dhe periudhë shlyerjeje. Shkruani një program i cili llogarit dhe shfaq pagesën mujore për huanë e marrë, duke përdorur formulën e pagesës mujore të huasë. Formula e njohur për këtë është:

13 pikë

$$M = \frac{P \cdot r \cdot (1 + r)^n}{(1 + r)^n - 1}$$

Ku:

- M është pagesa mujore;
- P është shuma e huasë;
- r është norma e interesit mujor (norma vjetore e interesit e ndarë me 12);
- n është numri i total i pagesave mujore (periudha në muaj).

Zgjidhje:

```

#include <iostream>

#include <cmath>

using namespace std;

int main() {
    double P, annualInterestRate, n;

    cout << "Shkruani shumën e huasë (P): ";
    cin >> P;

    cout << "Shkruani normën vjetore të interesit (%): ";
    cin >> annualInterestRate;

    cout << "Shkruani periudhën e shlyerjes (në muaj): ";
    cin >> n;

    double r = annualInterestRate / 12 / 100; // Norma mujore e interesit
    double M = P * r * pow(1 + r, n) / (pow(1 + r, n) - 1);

    cout << "Pagesa mujore është: " << M << endl;

    return 0;
}

```

5. Jepet një çantë me kapacitet K . Jepen n objekte me masa të ndryshme $m_1, m_2, m_3, \dots, m_n$. Shkruani një program që llogarit kombinimin e objekteve, i cili mundëson shfrytëzimin maksimal të kapacitetit të çantës.

10 pikë

Zgjidhje:

```

#include <iostream>

#include <vector>

using namespace std;

void knapsack(int capacity, vector<int>& weights) {
    int n = weights.size();

```

```
int bestWeight = 0;

vector<int> bestCombination;

// Gjenero të gjitha kombinimet e mundshme
for (int i = 0; i < (1 << n); i++) { // 2^n kombinime
    vector<int> currentCombination;
    int currentWeight = 0;

    for (int j = 0; j < n; j++) {
        if (i & (1 << j)) { // Kontrolllo nëse objekti i j-të është përfshirë
            currentCombination.push_back(weights[j]);
            currentWeight += weights[j];
        }
    }

    // Kontrolllo nëse pesha aktuale është më e mira brenda kufizimit
    if (currentWeight <= capacity && currentWeight > bestWeight) {
        bestWeight = currentWeight;
        bestCombination = currentCombination;
    }
}

// Shfaq rezultatin
cout << "Pesha më e mirë: " << bestWeight << endl;
cout << "Objektet e zgjedhura: ";
for (int w : bestCombination) {
    cout << w << " ";
}
cout << endl;
}
```

```
int main() {  
    int capacity = 15; // Kapaciteti i çantës  
    vector<int> weights = {2, 3, 5, 8, 9}; // Pesha e objekteve  
  
    knapsack(capacity, weights);  
  
    return 0;  
}
```