



REPUBLIKA E SHQIPËRISË
MINISTRIA E ARSIMIT
QENDRA E SHËRBIMEVE ARSIMORE

OLIMPIADA KOMBËTARE E INFORMATIKËS
NË ARSIMIN E MESËM TË LARTË

Faza e dytë

Klasa 10

17 janar 2026

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/C++.

Për përdorim nga komisioni i vlerësimit

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

Totali i pikëve të fituara

KOMISIONI I VLERËSIMIT

1.....

2.....

1. Jepet një tabelë $n \times m$, ku çdo qelizë përmban vlerat 0 (pa monedhë) ose 1 (me monedhë). Nisemi nga qeliza (1,1) dhe duam të shkojmë te qeliza (n,m), duke lëvizur vetëm djathtas ose poshtë. Shkruani një program që gjen numrin e rrugëve që arrijnë në (n,m), ku $1 \leq n \leq 40$, $1 \leq m \leq 40$ dhe mbledhin saktësisht K monedha, ku $0 \leq K \leq 60$.

12 pikë

Zgjidhje:

```
#include <iostream>

#include <vector>

using namespace std;

// Deklarojmë matricën e DP: dp[rreshti][kolona][monedhat]

// Përdorim long long sepse numri i rrugëve mund të jetë shumë i madh

long long dp[41][41][61];

int tabela[41][41];

int main() {

    int n, m, K;

    cout << "Jepni n, m dhe K: ";

    cin >> n >> m >> K;

    cout << "Jepni vlerat e tabelës (0 ose 1):" << endl;

    for (int i = 1; i <= n; i++) {

        for (int j = 1; j <= m; j++) {

            cin >> tabela[i][j];

        }

    }

    // Inicializojmë DP me 0

    for (int i = 0; i <= n; i++)

        for (int j = 0; j <= m; j++)

            for (int k = 0; k <= K; k++)

                dp[i][j][k] = 0;

    // Rasti bazë: qeliza e parë (1,1)

    int monedhaPare = tabela[1][1];

    if (monedhaPare <= K) {
```

```

dp[1][1][monedhaPare] = 1;

}

// Mbushja e matrices DP
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= m; j++) {
        for (int k = 0; k <= K; k++) {
            if (dp[i][j][k] > 0) {
                // Levizim Djathtas (i, j+1)
                if (j + 1 <= m) {
                    int k_ri = k + tabela[i][j + 1];
                    if (k_ri <= K) {
                        dp[i][j + 1][k_ri] += dp[i][j][k];
                    }
                }
            }
            // Levizim Poshte (i+1, j)
            if (i + 1 <= n) {
                int k_ri = k + tabela[i + 1][j];
                if (k_ri <= K) {
                    dp[i + 1][j][k_ri] += dp[i][j][k];
                }
            }
        }
    }
}

cout << "Numri i rrugëve me saktësisht " << K << " monedha: ";
cout << dp[n][m][K] << endl;

return 0;
}

```

2. Shkruani një program i cili gjeneron një numër a nga intervali 1 në 200. Përdoruesi duhet të gjejë numrin a me maksimumi 10 tentativa. Për çdo tentativë, programi jep mesazhin “Më lartë”, “Më poshtë” ose “U gjet”. Nëse gjendet numri, programi ndalon.

8 pikë**Zgjidhje:**

```
#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int main() {

    srand(time(0));      // Inicializimi i numrave të rastësishëm

    int a = rand() % 200 + 1; // Numër nga 1 në 200

    int guess;

    for (int tentativa = 1; tentativa <= 10; tentativa++) {

        cout << "Tentativa " << tentativa << ": Jepni nje numer: ";

        cin >> guess;

        if (guess < a) {

            cout << "Me larte" << endl;

        } else if (guess > a) {

            cout << "Me poshte" << endl;

        } else {

            cout << "U gjet!" << endl;

            return 0; // Programi ndalon nëse numri gjendet

        }

    }

    cout << "Nuk e gjetet numrin. Numri ishte: " << a << endl;

    return 0;

}
```

3. Shkruani një program që implementon një sistem menaxhimi të bibliotekës. Për çdo libër, sistemi ruan titullin, autorin, çmimin dhe numrin serial. Sistemi duhet të mundësojë:

10 pikë

- a) shtimin e librave;
- b) fshirjen e librave sipas numrit serial;
- c) kërkimin sipas titullit;
- d) afishimin e librave të renditur sipas çmimit më të lirë.

Zgjidhje:

```
#include <iostream>

#include <string>

using namespace std;

struct Libri {
    string titulli;
    string autori;
    float cmimi;
    int serial;
};

int main() {
    Libri libra[100];
    int n = 0;
    int zgjedhja;
    do {
        cout << "\n--- MENU ---\n";
        cout << "1. Shto liber\n";
        cout << "2. Fshi liber sipas numrit serial\n";
        cout << "3. Kerko liber sipas titullit\n";
        cout << "4. Afisho librat sipas cmimit me te lire\n";
        cout << "5. Dil\n";
        cout << "Zgjedhja juaj: ";
        cin >> zgjedhja;
        cin.ignore();
        if (zgjedhja == 1) {
```

```
cout << "Titulli: ";

getline(cin, libra[n].titulli);

cout << "Autori: ";

getline(cin, libra[n].autori);

cout << "Cmimi: ";

cin >> libra[n].cmimi;

cout << "Numri serial: ";

cin >> libra[n].serial;

n++;

cout << "Libri u shtua me sukses!\n";

}

else if (zgjedhja == 2) {

    int s;

    cout << "Jepni numrin serial: ";

    cin >> s;

    bool uGjet = false;

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

        if (libra[i].serial == s) {

            for (int j = i; j < n - 1; j++) {

                libra[j] = libra[j + 1];

            }

            n--;

            uGjet = true;

            cout << "Libri u fshi.\n";

            break;

        }

    }

    if (!uGjet)

        cout << "Libri nuk u gjet.\n";

}
```

```

else if (zgjedhja == 3) {
    string tit;
    cout << "Jepni titullin: ";
    getline(cin, tit);
    bool uGjet = false;
    for (int i = 0; i < n; i++) {
        if (libra[i].titulli == tit) {
            cout << "\nTitulli: " << libra[i].titulli
                << "\nAutori: " << libra[i]. autori
                << "\nCmimi: " << libra[i].cmimi
                << "\nSeriali: " << libra[i].serial << endl;
            uGjet = true;
        }
    }
    if (!uGjet)
        cout << "Libri nuk u gjet.\n";
}

else if (zgjedhja == 4) {
    // Renditje sipas cmimit (nga me i liri)
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (libra[i].cmimi > libra[j].cmimi) {
                swap(libra[i], libra[j]);
            }
        }
    }

    cout << "\nLista e librave:\n";
    for (int i = 0; i < n; i++) {
        cout << libra[i].titulli << " | "
            << libra[i]. autori << " | "

```

```
<< libra[i].cmimi << " | "
```

```
<< libra[i].serial << endl;
```

```
}
```

```
}
```

```
} while (zgjedhja != 5);
```

```
return 0;
```

```
}
```

4. Shkruani një program i cili kontrollon vlefshmërinë e fjalëkalimit të futur nga përdoruesi. Kriteret për kontrollin e fjalëkalimit janë: **10 pikë**

- a) të paktën 1 shkronjë midis [a-z];
- b) të paktën 1 shkronjë midis [A-Z];
- c) të paktën 1 numër midis [0-9];
- d) të paktën 1 karakter nga [\$, #, @, %];
- e) gjatësia e fjalëkalimit të jetë nga 8 deri në 16 karaktere.

Zgjidhje:

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
int main() {
```

```
    string password;
```

```
    bool kaVogel = false, kaMadh = false, kaNumer = false, kaSpecial = false;
```

```
    cout << "Vendosni fjalekalimin: ";
```

```
    cin >> password;
```

```
    // Kontrolli i gjatësisë
```

```
    if (password.length() < 8 || password.length() > 16) {
```

```
        cout << "Fjalekalimi nuk eshte i vlefshem!" << endl;
```

```
        return 0;
```

```
}
```

// Kontrolli i karaktereve

```
for (int i = 0; i < password.length(); i++) {
    if (password[i] >= 'a' && password[i] <= 'z')
        kaVogel = true;
    else if (password[i] >= 'A' && password[i] <= 'Z')
        kaMadh = true;
    else if (password[i] >= '0' && password[i] <= '9')
        kaNumer = true;
    else if (password[i] == '$' || password[i] == '#' ||
        password[i] == '@' || password[i] == '%')
        kaSpecial = true;
}

// Vendimi final

if (kaVogel && kaMadh && kaNumer && kaSpecial)
    cout << "Fjalekalimi eshte i vlefshem!" << endl;
else
    cout << "Fjalekalimi nuk eshte i vlefshem!" << endl;

return 0;
}
```

5. Jepen tre numra natyrorë m , A , F , të tillë që $10 \leq A \leq F \leq m$.

10 pikë

Një numër x quhet i mirë nëse është i plotëpjesëtueshëm me shumën e shifrave të tij.

Gjeni sasinë e numrave të mirë në intervalin $[A, F]$.

Zgjidhje:

```
#include <iostream>

using namespace std;

// Funkcion per te llogaritur shumen e shifrave te nje numri

int llogaritShumenEShifrave(int n) {
    int shuma = 0;
    while (n > 0) {
        shuma += n % 10; // Merr shifren e fundit
```

n /= 10; // Heq shifren e fundit

}

return shuma;

}

int main() {

int m, A, F;

int sasiaNumraveTeMire = 0;

// Leximi i te dhenave m, A, F

cout << "Jepni vlerat per m, A dhe F ($10 \leq A \leq F \leq m$).\" << endl;

cout << "m: "; cin >> m;

cout << "A: "; cin >> A;

cout << "F: "; cin >> F;

// Kontrolli i kushtit: $10 \leq A \leq F \leq m$

if ($A < 10 \parallel A > F \parallel F > m$) {

cout << "Gabim: Vlerat nuk plotesojne kushtin $10 \leq A \leq F \leq m$." << endl;

return 1;

}

// Gjetja e numrave te mire ne intervalin [A, F]

for (int x = A; x <= F; x++) {

int shuma = llogaritShumenEShifrave(x);

// Kushti: x eshte i mire nese plotejpjesetohet me shumen e shifrave

if (x % shuma == 0) {

sasiaNumraveTeMire++;

}

}

// Afishimi i rezultatit

cout << "Sasia e numrave te mire ne intervalin [" << A << ", " << F << "] eshte: "

<< sasiaNumraveTeMire << endl;

return 0;

}