



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

OLIMPIADA KOMBËTARE E INFORMATIKËS PËR ARSIMIN E MESËM TË LARTË

Viti shkollor 2022-2023

Faza e tretë

ZGJIDHJET

Ushtrimi 1.**8 pikë**

```
#include <stdio.h>

int find_pmp(int num1, int num2) {
    while(num1 != num2) {
        if(num1 > num2)
            num1 -= num2;
        else
            num2 -= num1;
    }
    return num1;
}

int main() {
    int n, vector[100], result;

    printf("Vendosni numrin e elementëve të vektorit: ");
    scanf("%d", &n);

    printf("Vendosni elementët e vektorit: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &vector[i]);
    }

    result = vector[0];
    for (int i = 1; i < n; i++) {
        result = find_pmp(result, vector[i]);
    }

    printf("PMP i elementëve të këtij vektori është: %d\n", result);

    return 0;
}
```

Ushtrimi 2.**10 pikë**

```
#include <stdio.h>
#define MAX_SIZE 10
int main()
{
    int n, i, j, shuma_1, rreshti_shume, kolona_shume, diag_shume, rev_diag_shume;
    int matrix[MAX_SIZE][MAX_SIZE];
```

```

printf("Vendosni permasat e matricës: ");
scanf("%d", &n);
printf("Vendosni elementet e matricës:\n");
for(i = 0; i < n; i++) {
    for(j = 0; j < n; j++) {
        scanf("%d", &matrix[i][j]);
    }
}
// llogarisim shumën e rreshtit të parë
shuma_1 = 0;
for(i = 0; i < n; i++) {
    shuma_1 += matrix[0][i];
}
// kontrollojmë nëse të gjitha rreshtat, kolonat dhe diagonalet kanë të njëjtën shumë
for(i = 0; i < n; i++) {
    rreshti_shume = 0;
    kolona_shume = 0;
    for(j = 0; j < n; j++) {
        rreshti_shume += matrix[i][j];
        kolona_shume += matrix[j][i];
    }
    if(rreshti_shume != shuma_1 || kolona_shume != shuma_1) {
        printf("Matrica nuk është katror magjik.\n");
        return 0;
    }
}
// Kontrollojmë shumën e diagonaleve
diag_shume = 0;
rev_diag_shume = 0;
for(i = 0; i < n; i++) {
    diag_shume += matrix[i][i];
    rev_diag_shume += matrix[i][n - i - 1];
}
if(diag_shume != shuma_1 || rev_diag_shume != shuma_1) {

```

```
    printf("Matrica nuk është katror magjik.\n");
    return 0;
}
printf("Matrica është katror magjik.\n");
return 0;
}
```

Ushtrimi 3.

4 pikë

```
#include <stdio.h>
#include <string.h>

int main()
{
    char fjale[100];
    int i, j, gjatesi, flag = 1;

    printf("Vendosi nje fjalë: ");
    scanf("%s", fjale);

    gjatesi = strlen(fjale);

    for (i = 0, j = gjatesi - 1; i < gjatesi / 2; i++, j--)
    {
        if (fjale[i] != fjale[j])
        {
            flag = 0;
            break;
        }
    }

    if (flag == 1)
        printf("%s është një palindromë.\n", fjale);
    else
        printf("%s nuk është një palindromë.\n", fjale);

    return 0;
}
```

Ushtrimi 4.**3 pikë**

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main() {
    char str[100][100];
    int i, n;

    printf("Vendosni numrin e fjalëve: ");
    scanf("%d", &n);

    printf("Vendosni %d fjalët: \n", n);
    for(i = 0; i < n; i++) {
        scanf("%s", str[i]);

        // Kthen me gërmë të madhe inicialin e parë
        str[i][0] = toupper(str[i][0]);
    }

    printf("Fjalët e rregulluara janë :\n");
    for(i = 0; i < n; i++) {
        printf("%s\n", str[i]);
    }

    return 0;
}
```

Ushtrimi 5.

10 pikë

```

#include <stdio.h>
#define MAX_SIZE 100
int main() {
    int n, i, j;
    int matrix[MAX_SIZE][MAX_SIZE];
    printf("Vendosni dimensionin e matricës: ");
    scanf("%d", &n);
    printf("Vendosni elementët e matricës:\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }
    // Kthen matricën në kahun orar me 90 gradë
    for (i = 0; i < n/2; i++) {
        for (j = i; j < n-i-1; j++) {
            int temp = matrix[i][j];
            matrix[i][j] = matrix[n-j-1][i];
            matrix[n-j-1][i] = matrix[n-i-1][n-j-1];
            matrix[n-i-1][n-j-1] = matrix[j][n-i-1];
            matrix[j][n-i-1] = temp;
        }
    }
    printf("Matrica e kthyer është:\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

Ushtrimi 6.

15 pikë

```

#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
struct TrainRecord
{
    int pasagjerë;
    int ora;
};
int main() {
    std::ifstream infile("travel.logs");
    std::vector<TrainRecord> records;
    int total_pasagjerë = 0;
    for (int i = 0; i < 7; ++i) {
        int pasagjerë, ora;
        while (infile >> pasagjerë >> ora) {
            records.push_back({ pasagjerë, ora });
            total_pasagjerë += pasagjerë;
        }
    }
    infile.close();
    // Llogarit totalin e pasagjerëve per çdo ditë
    std::vector<int> pasagjerë_ditor(7, 0);
    for (const auto& record : records) {
        int dita = (record.ora / 24) % 7;
        pasagjerë_ditor[dita] += record.pasagjerë;
    }
    // Llogarit numrin mesatar të pasagjerëve në një udhëtim
    int avg_pasagjerë = records.empty() ? 0 : total_pasagjerë / records.size();

    // Grupon trenat sipas mëngjesit, pasdites dhe mbrëmjes
    std::vector<std::vector<TrainRecord>> groups(3);
    for (const auto& record : records) {
        if (record.ora >= 7 && record.ora < 12) {
            groups[0].push_back(record); // Trenat e mëngjesit
        } else if (record.ora >= 12 && record.ora < 18) {
            groups[1].push_back(record); // Trenat e pasdites
        } else {
            groups[2].push_back(record); // Trenat e mbrëmjes
        }
    }
}

```

```

// Llogarit statistikat per secilin grup trenash
std::vector<std::pair<int, int>> group_stats(3);
for (int i = 0; i < 3; ++i) {
    int group_total = 0;
    for (const auto& record : groups[i]) {
        group_total += record.pasagjerë;
    }
    int group_avg = groups[i].empty() ? 0 : group_total / groups[i].size();
    group_stats[i] = { group_avg, group_total };
}
// Llogarit orarin më të ngarkuar dhe më pak të ngarkuar
std::vector<int> ora_totals(24, 0);
for (const auto& record : records) {
    ++ora_totals[record.ora];
}

auto busiest_time = std::max_element(ora_totals.begin(), ora_totals.end());
auto least_busy_time = std::min_element(ora_totals.begin(), ora_totals.end());
// Paraqet rezultatet ne output file
std::ofstream outfile("output.txt");
outfile << "Totali i pasagjerëve në javë: " << total_pasagjerë << std::endl;
outfile << "Mesatarja e pasagjerëve për udhëtim: " << avg_pasagjerë << std::endl;
outfile << "Total pasagjerë per ditën: ";
for (int i = 0; i < 7; ++i) {
    outfile << pasagjerë_ditor[i] << " ";
}
outfile << std::endl;
outfile << "Mesatarja e pasagjerëve për trenat e mëngjesit: " << group_stats[0].first << std::endl;
outfile << "Totali i pasagjerëve për trenat e mëngjesit: " << group_stats[0].second << std::endl;
outfile << " Mesatarja e pasagjerëve për trenat e pasdites: " << group_stats[1].first << std::endl;
outfile << " Totali i pasagjerëve për trenat e pasdites:" << group_stats[1].second << std::endl;
outfile << " Mesatarja e pasagjerëve për trenat e mbrëmjes: " << group_stats[2].first << std::endl;
outfile << " Totali i pasagjerëve për trenat e mbrëmjes:" << group_stats[2].second << std::endl;
}

```

*Kjo është një nga zgjidhjet e ushtrimit 6, ku në kolonën e orëve është vendosur numri total i orëve që nga nisja e qarkullimit të trenave të stacionit nga Dita 1, ora 00:00.

Shënim 1: Pranohet çdo zgjidhje tjetër e saktë, që nuk është parashikuar më lart, të cilin komisioni i vlerësimit e gjykon si të tillë.

Shënim 2: Në folderin e zgjidhjeve të ushtrimeve duhet programi dhe source code për çdo ushtrim që pikët të jenë të plota.

Shënim 3: Ushtrimet të cilat janë cilësuar si plagjaturë (kopjim) të kodit në faqe të ndryshme në internet, kanë ulje pikësh.