



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

OLIMPIADA KOMBËTARE E INFORMATIKËS
NË ARSIMIN E MESËM TË LARTË

Klasa 12

Faza e dytë

Viti shkollor: 2023-2024

ZGJIDHJE

Ushtrimi 1. Shkruani një program ku numri i rreshtave merret nga përdoruesi dhe afishon figurën e mëposhtme.

Input: 4

5 pikë

Output: ****

**
*

Zgjidhja e ushtrimit 1:

```
#include <stdio.h>

int main() {
    int n;

    // Merr numrin e rreshtave nga përdoruesi
    printf("Shkruani numrin e rreshtave: ");
    scanf("%d", &n);

    // Krijimi dhe shfaqja e figurës
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < i; j++) {
            printf(" ");
        }
        for (int k = 0; k < n - i; k++) {
            printf("**");
        }
    }
}
```

```

}

printf("\n");

}

return 0;

}

```

Ushtrimi 2. Jepet n , një numër i plotë pozitiv. Shkruani një program për të kontrolluar nëse një grup prej n fjalësh të vendosura nga përdoruesi janë anagramë të njëra-tjetrës. *Anagramë janë fjalë të cilat kanë të njëjtat karaktere, por në renditje të ndryshme.*

8 pikë

Zgjidhja e ushtrimit 2:

```

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

// Funkcion për të kontrolluar nëse dy fjalë janë anagramë
int areAnagrams(char *word1, char *word2) {
    int len1 = strlen(word1);
    int len2 = strlen(word2);

    // Nëse fjalët kanë gjatësi të ndryshme, ato nuk janë anagramë
    if (len1 != len2) {
        return 0;
    }

    // Krijo një varg për të mbajtur numrin e shkronjave të secilës fjalë
    int count1[26] = {0};
    int count2[26] = {0};

    // Mbush vargun e numrave për fjalën e parë
    for (int i = 0; i < len1; i++) {
        count1[word1[i] - 'a']++;
    }

    // Mbush vargun e numrave për fjalën e dytë
    for (int i = 0; i < len2; i++) {
        count2[word2[i] - 'a']++;
    }

    // Kontrolllo nëse numrat e shkronjave janë të njëjta për secilën fjalë
    for (int i = 0; i < 26; i++) {
        if (count1[i] != count2[i]) {
            return 0;
        }
    }
}

```

```

    }
}

// Nëse nuk ka dallime, fjalët janë anagramë
return 1;
}

int main() {
    int n;
    printf("Shkruani numrin e fjalëve: ");
    scanf("%d", &n);

    char words[n][100];

    // Merr fjalët nga përdoruesi
    for (int i = 0; i < n; i++) {
        printf("Shkruani fjalen %d: ", i + 1);
        scanf("%s", words[i]);
    }

    // Kontrollon nëse çdo çift i fjalëve është anagramë
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (areAnagrams(words[i], words[j])) {
                printf("%s dhe %s janë anagramë.\n", words[i], words[j]);
            }
        }
    }

    return 0;
}

```

Ushtrimi 3. Jepen m dhe n , dy numra të plotë pozitivë. Shkruani një program i cili gjeneron matricën A [$m \times n$] dhe matricën e transpozuar B [$n \times m$] të matricës A . **10 pikë**

Zgjidhja e ushtrimit 3:

```

#include <stdio.h>

int main() {
    int m, n;

    // Merr vlerat e m dhe n nga përdoruesi
    printf("Shkruani numrin e rreshtave (m): ");
    scanf("%d", &m);
    printf("Shkruani numrin e kolonave (n): ");
    scanf("%d", &n);

```

```
// Krijohet matricën A[m x n] dhe mbush vlerat e saj
int A[m][n];

printf("Mbushja e matricës A[m x n]:\n");
for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++) {
        printf("A[%d][%d]: ", i, j);
        scanf("%d", &A[i][j]);
    }
}

// Shfaq matricën A
printf("\nMatrica A[m x n]:\n");
for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++) {
        printf("%d\t", A[i][j]);
    }
    printf("\n");
}

// Krijohet matricën B[n x m] dhe mbush vlerat e saj nga matrica A
int B[n][m];

for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        B[i][j] = A[j][i];
    }
}

// Shfaq matricën e transpozuar B
printf("\nMatrica e transpozuar B[n x m]:\n");
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        printf("%d\t", B[i][j]);
    }
    printf("\n");
}

return 0;
}
```

Ushtrimi 4. Jepet vektori N [1:n] me emrat e n punonjësve dhe vektori O [1:n] me numrin e orëve të punës jashtë orarit zyrtar. Një orë pune nga orët e punës jashtë orarit zyrtar paguhet me 13\$. Shkruani një program i cili afishon punonjësin me më shumë orë jashtë orari dhe shumën e parave që fitoi ai.

12 pikë

Zgjidhja e ushtrimit 4:

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

#define PAGA_PER_ORE 13

int main() {
    int n;

    // Merr numrin e punonjësve nga përdoruesi
    printf("Shkruani numrin e punonjësve: ");
    scanf("%d", &n);

    // Deklaro vektoret për emrat dhe orët e punës jashtë orarit
    char emrat[n][50];
    int oret[n];

    // Mbushja e vektoreve me emrat dhe orët e punës jashtë orarit
    for (int i = 0; i < n; i++) {
        printf("Emri i punonjësit %d: ", i + 1);
        scanf("%s", emrat[i]);
        printf("Numri i orëve jashtë orarit për %s: ", emrat[i]);
        scanf("%d", &oret[i]);
    }

    // Gjeje punonjësin me më shumë orë jashtë orarit
    int indeksiMaxOre = 0;
    for (int i = 1; i < n; i++) {
        if (oret[i] > oret[indeksiMaxOre]) {
            indeksiMaxOre = i;
        }
    }

    // Llogarit shumën e parave të fituara nga punonjësi me më shumë orë jashtë orarit
    int shumaParave = oret[indeksiMaxOre] * PAGA_PER_ORE;

    // Afisho rezultatin
    printf("\nPunonjësi me më shumë orë jashtë orarit:\n");
    printf("Emri: %s\n", emrat[indeksiMaxOre]);
```

```

printf("Numri i orëve: %d\n", orete[indeksiMaxOre]);
printf("Shuma e parave të fituara: $%d\n", shumaParave);

return 0;
}

```

Ushtrimi 5. Të ndërtohet programi i cili analizon të dhënat shëndetësore për 3 pacientë, për 10 ditë. Përdoruesit duhet të kenë mundësi të shtojnë për çdo ditë totalin e hapave të kryera nga pacientët dhe kalorive të konsumuara. Programi duhet të afishojë:

15 pikë

- totalin e hapave të të gjithë pacientëve për çdo ditë.
- ditën me konsumin më të lartë të kalorive për çdo pacient.
- pacientin me hapat mesatarë ditorë më të lartë.

Zgjidhja e ushtrimit 5:

```

#include <stdio.h>

#define NUMRIPACIENTEVE 3
#define NUMRIDITEVE 10

int main() {
    // Deklaro vektoret për të mbajtur të dhënat për secilin pacient dhe secilën ditë
    int hapat[NUMRIDITEVE][NUMRIPACIENTEVE];
    int kalorite[NUMRIDITEVE][NUMRIPACIENTEVE];

    // Mbushja e vektoreve me të dhënat nga përdoruesi
    for (int i = 0; i < NUMRIDITEVE; i++) {
        printf("Dita %d:\n", i + 1);
        for (int j = 0; j < NUMRIPACIENTEVE; j++) {
            printf("Pacienti %d - Numri i hapave: ", j + 1);
            scanf("%d", &hapat[i][j]);
            printf("Pacienti %d - Numri i kalorive: ", j + 1);
            scanf("%d", &kalorite[i][j]);
        }
    }

    // a) Totali i hapave për çdo ditë
    printf("\nTotali i hapave per cdo dite:\n");
    for (int i = 0; i < NUMRIDITEVE; i++) {
        int totalHapatDites = 0;
        for (int j = 0; j < NUMRIPACIENTEVE; j++) {
            totalHapatDites += hapat[i][j];
        }
        printf("Dita %d: %d hapa\n", i + 1, totalHapatDites);
    }
}

```

// b) Ditën me konsumin më të lartë të kalorive për çdo pacient

```
printf("\nDita me konsumin me te larte te kalorive per cdo pacient:\n");
for (int j = 0; j < NUMRIPACIENTEVE; j++) {
    int maxKalorite = kalorite[0][j];
    int ditaMaxKalorive = 1;
    for (int i = 1; i < NUMRIDITEVE; i++) {
        if (kalorite[i][j] > maxKalorite) {
            maxKalorite = kalorite[i][j];
            ditaMaxKalorive = i + 1;
        }
    }
    printf("Pacienti %d: Dita %d me %d kalori\n", j + 1, ditaMaxKalorive, maxKalorite);
}
```

// c) Pacienti me hapat mesatarë ditorë më të lartë

```
printf("\nPacienti me hapat mesatarë ditorë më të lartë:\n");
float mesatarjaMax = 0;
int pacientiMaxMesatarja = 0;
for (int j = 0; j < NUMRIPACIENTEVE; j++) {
    float mesatarjaHapave = 0;
    for (int i = 0; i < NUMRIDITEVE; i++) {
        mesatarjaHapave += hapat[i][j];
    }
    mesatarjaHapave /= NUMRIDITEVE;

    printf("Pacienti %d: Mesatarja e hapave = %.2f\n", j + 1, mesatarjaHapave);

    if (mesatarjaHapave > mesatarjaMax) {
        mesatarjaMax = mesatarjaHapave;
        pacientiMaxMesatarja = j + 1;
    }
}
printf("Pacienti me hapat mesatarë ditorë më të lartë: Pacienti %d\n", pacientiMaxMesatarja);

return 0;
}
```