

13. Dodatne vježbe

1. Ponovljeni zadatak s predavanja: definirati i inicijalizirati trodimenzionalno cjelobrojno polje s dimenzijama koje sami odaberite (npr. 3, 4, 5). Koristiti inicijalizator s vitičastim zagradama (svaki sloj unutar svojih vitičastih zagrada, svaki redak sloja unutar svojih vitičastih zagrada). Ispišite polje po slojevima, u sljedećem obliku:

```
x x x x x
x x x x x
x x x x x
x x x x x
```

```
x x x x x
x x x x x
x x x x x
x x x x x
```

```
x x x x x
x x x x x
x x x x x
x x x x x
```

2. Korisnik odabire dimenzije matrice koju želi učitati (brRed, brStup). Broj redaka koje korisnik može učitati ne smije biti manji od 1 niti veći od 20, a broj stupaca ne smije biti manji od 1 niti veći od 10. U matricu treba učitati vrijednosti elemenata i ispisati matricu. Nakon toga treba retke matrice posmaknuti prema gore: redak i dobiva vrijednosti iz retka $i+1$, a posljednji redak matrice dobiva vrijednosti 0-tog retka. Npr. ako je korisnik učitao: brRed=4, brStup=5, te elemente matrice:

```
1 2 3 4 5
2 1 3 7 1
8 1 1 4 3
4 2 2 3 3
```

nakon posmaka redaka, matrica treba izgledati ovako:

```
2 1 3 7 1
8 1 1 4 3
4 2 2 3 3
1 2 3 4 5
```

Nakon obavljenog posmaka redaka, ispisati novu matricu. **Zadatak riješite bez upotrebe pomoćnog polja.**

3. Korisnik odabire dimenzije matrice koju želi učitati (brRed, brStup). Broj redaka koje korisnik može učitati ne smije biti manji od 1 niti veći od 4, a broj stupaca ne smije biti manji od 1 niti veći od 5. S tipkovnice se učitaju vrijednosti elemenata matrice. Nakon toga u elemente u retku brRed zapisati sume stupaca, u elemente u stupcu brStup zapisati sume redaka, a u element (brRed,brStup) sumu svih elemenata matrice. Ispisati dobivenu matricu. Npr. ako je korisnik učitao:

brRed=4, brStup=5, te elemente matrice:

```
1 2 3 4 5
2 1 3 7 1
8 1 1 4 3
4 2 2 3 3
```

Nakon izračuna, ispis matrice izgleda ovako:

```
1 2 3 4 5 15
2 1 3 7 1 14
8 1 1 4 3 17
4 2 2 3 3 14
15 6 9 18 12 60
```

Zadatak riješite bez upotrebe pomoćnog polja.

4. S tipkovnice upisati cijeli broj n iz intervala $[5, 10]$. Generirati kvadratnu matricu dimenzija $n \times n$ kojoj su svi elementi glavne i sporedne dijagonale, elementi prvog i zadnjeg retka, te prvog i zadnjeg stupca postavljeni na vrijednost 1, a svi ostali elementi matrice postavljeni na vrijednost 8. Generiranu matricu ispisati na zaslon. Npr. ako korisnik preko tipkovnice za n unese broj 7, na zaslon treba ispisati:

```
1 1 1 1 1 1 1
1 1 8 8 8 1 1
1 8 1 8 1 8 1
1 8 8 1 8 8 1
1 8 1 8 1 8 1
1 1 8 8 8 1 1
1 1 1 1 1 1 1
```

5. S tipkovnice, uz kontrolu, učitati broj redaka m i broj stupaca n cjelobrojne matrice. Broj redaka mora biti iz intervala $[3, 6]$, a broj stupaca iz intervala $[4, 10]$. Nakon toga učitati članove matrice. Jednodimenzionalno polje nps napuniti vrijednostima najmanjih članova matrice u pojedinim stupcima, a jednodimenzionalno polje npr napuniti vrijednostima najmanjih članova matrice u pojedinim retcima. Ispisati učitanoj matrici, a zatim članove polja nps i članove polja npr .

Primjer: ako je učitana matrica ($m=4$, $n=5$)

```
1 2 3 4 5
2 0 3 7 2
8 1 5 4 3
4 2 2 9 3
```

program treba ispisati:

matrica:

```
1 2 3 4 5
2 0 3 7 2
8 1 5 4 3
4 2 2 9 3
```

polje nps:

```
1 0 2 4 2
```

polje npr:

```
1 0 1 2
```

Rješenja svih zadataka provjeriti prevođenjem i testiranjem vlastitih programa!

Rješenja: NE GLEDATI prije nego sami pokušate riješiti zadatke

Rješenje 1. zadatka

```
#include <stdio.h>

#define MAXSLOJ 3
#define MAXRED 4
#define MAXSTUP 5

int main() {
    int i, j, k;
    int tridim[MAXSLOJ][MAXRED][MAXSTUP] =
    {
        { { 101, 102, 103, 104, 105},
          { 106, 107, 108, 109, 110},
          { 111, 112, 113, 114, 115},
          { 116, 117, 118, 119, 120 }
        },
        { { 201, 202, 203, 204, 205},
          { 206, 207, 208, 209, 210},
          { 211, 212, 213, 214, 215},
          { 216, 217, 218, 219, 220 }
        },
        { { 301, 302, 303, 304, 305},
          { 306, 307, 308, 309, 310},
          { 311, 312, 313, 314, 315},
          { 316, 317, 318, 319, 320 }
        }
    };

    for (i = 0; i < MAXSLOJ; i++) {
        for (j = 0; j < MAXRED; j++) {
            for (k = 0; k < MAXSTUP; k++)
                printf("%d ", tridim[i][j][k]);
            printf("\n");
        }
        printf("\n");
    }

    return 0;
}
```

Rješenje 2. zadatka

```
#include <stdio.h>

#define MAXRED 20
#define MAXSTUP 10

int main() {
    int i, j, m, n, pom;
    int mat[MAXRED][MAXSTUP];
    do {
        printf("Upisite m i n:");
        scanf("%d %d", &m, &n);
    } while (m < 1 || m > 20 || n < 1 || n > 10);

    printf("Upisite clanove matrice po retcima:");

    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            scanf("%d", &mat[i][j]);

    for (j = 0; j < n; j++) {
        /* u svakom stupcu j obavi sljedece */
        pom = mat[0][j]; /* spasi clan iz 0-tog retka */

        for (i = 0; i < m-1; i++)
            mat[i][j] = mat[i+1][j];

        mat[m-1][j] = pom; /* u zadnji redak stavi spaseni clan iz 0-tog retka */
    }

    printf("\n\nIspis nove matrice:\n");

    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++)
            printf("%d ", mat[i][j]);
        printf("\n");
    }

    return 0;
}
```

Rješenje 3. zadatka

```
#include <stdio.h>

#define MAXRED 4
#define MAXSTUP 5

int main() {
    int i, j, m, n;
    int mat[MAXRED+1][MAXSTUP+1] = {0};
    /* dimenzije su uvecane za 1 jer treba osigurati
       prostor za dodatni redak i stupac */
    do {
        printf("Upisite m i n:");
        scanf("%d %d", &m, &n);
    } while (m < 1 || m > MAXRED || n < 1 || n > MAXSTUP);

    printf("Upisite clanove matrice po retcima:");

    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            scanf("%d", &mat[i][j]);

    /* kontrolni ispis učitane matrice */
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++)
            printf("%4d ", mat[i][j]);
        printf("\n");
    }

    /* proci cemo kroz sve elemente i svaki element
       pribrojiti u odgovarajuće elemente u dodatnom retku
       i dodatnom stupcu. Clanovi matrice su inicijalizirani
       na 0 pri definiciji (inace bi trebalo clanove retka m i
       stupca n prvo postaviti na 0) */

    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++) {
            /* dodaj ga u zbroj odgovarajućeg retka */
            mat[i][n] += mat[i][j];
            /* dodaj ga u zbroj odgovarajućeg stupca */
            mat[m][j] += mat[i][j];
            /* dodaj ga u zbroj svih elemenata matrice */
            mat[m][n] += mat[i][j];
        }

    printf("\n\nIspis nove matrice:\n");
    for (i = 0; i <= m; i++) {
        for (j = 0; j <= n; j++)
            printf("%4d ", mat[i][j]);
        printf("\n");
    }
}
```

Rješenje 4. zadatka

```
#include <stdio.h>
#define MAXDIM 10

int main() {
    int i, j, n;
    int mat[MAXDIM][MAXDIM];

    /* učitavanje n */
    do {
        printf("Upisite n:");
        scanf("%d", &n);
    } while (n < 1 || n > MAXDIM);

    /* generiranje matrice */
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            if (i == j || i == n-1-j || j == 0 || i == 0 || j == n-1 || i == n-1)
                mat[i][j] = 1;
            else
                mat[i][j] = 8;

    /* ispis matrice */
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++)
            printf("%d ", mat[i][j]);
        printf("\n");
    }

    return 0;
}
```

Rješenje 5. zadatka

```
#include <stdio.h>

#define MAXRED 6
#define MAXSTUP 10

int main() {
    int i, j, m, n;
    int mat[MAXRED][MAXSTUP];
    int nps[MAXSTUP], npr[MAXRED];

    do {
        printf("Upisite m[3,6] i n[4,10]: ");
        scanf("%d %d", &m, &n);
    } while (m < 3 || m > 6 || n < 4 || n > 10);

    printf("Upisite clanove matrice po retcima: ");

    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            scanf("%d", &mat[i][j]);

    /* odredi najmanji clan u svakom stupcu */
    for (j = 0; j < n; j++) {
        /* u svakom stupcu j obavi sljedece */
        nps[j] = mat[0][j]; /* pretpostavi da je prvi clan u j-tom stupcu
                             najmanji clan tog stupca */
        for (i = 1; i < m; i++)
            if (mat[i][j] < nps[j])
                nps[j] = mat[i][j];
    }

    /* odredi najmanji clan u svakom retku */
    for (i = 0; i < m; i++) {
        /* u svakom retku i obavi sljedece */
        npr[i] = mat[i][0]; /* pretpostavi da je prvi clan u i-tom retku
                             najmanji clan tog retka */
        for (j = 1; j < n; j++)
            if (mat[i][j] < npr[i])
                npr[i] = mat[i][j];
    }

    printf("\nmatrica:\n");
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++)
            printf("%4d", mat[i][j]);
        printf("\n");
    }

    printf("\npolje nps:\n");
    for (j = 0; j < n; j++)
        printf("%4d", nps[j]);
    printf("\n");

    printf("\npolje npr:\n");
    for (i = 0; i < m; i++)
        printf("%4d", npr[i]);
    printf("\n");

    return 0;
}
```