

Napomene:

- Savjetuje se navedene zadatke riješiti ubrzo nakon predavanja
- Savjetuje se ne gledati rješenja prije nego se pokuša samostalno riješiti zadatke

14. vježbe uz predavanja

1. Što će se ispisati sljedećim programom:

```
#include <stdio.h>
int main (void) {
    int a = 2, x = 10;
    int *p = &a;
    x += *p * 3;
    printf ("%d %d\n", *p, x);
    return 0;
}
```

2. Što će se ispisati sljedećim programom:

```
#include <stdio.h>
int main (void) {
    int a = 10, b = 20, *c = &a, *d = &b;
    c = &b;
    d = &a;
    *c = *d + 5;
    *d = *c + 3;
    printf ("%d %d %d %d\n", a, b, *c+1, *d+1);
    return 0;
}
```

3. Što je neispravno u sljedećem programu:

```
#include <stdio.h>
int main (void) {
    int a, b, *c = &a;
    short x, y = 10, *z = &x;
    c = &b;
    z = &y;
    *c = 20;
    *z = 30;
    c = &y;
    printf ("%d %d\n", *c, *z);
    return 0;
}
```

4. Što će se ispisati sljedećim programom:

```
#include <stdio.h>
int main (void) {
    int polje[4] = {3, 6, 9, 12};
    int *p1, *p2, *p3;
    p1 = &polje[0];
    p2 = &polje[1];
    p3 = &polje[3];
    printf ("%d %d %d %d %d\n", *p1, *p2, *p3, *(p2 - 1), *p2 - 1);
    return 0;
}
```

5. Što će se ispisati sljedećim programom:

```
#include <stdio.h>
int main (void) {
    int polje[4][2] = {3, 6, 9, 12, 15, 18, 21, 24};
    int *p1, *p2, *p3;
    p1 = &polje[0][0];
    p2 = &polje[1][0];
    p3 = &polje[3][1];
    printf ("%d %d %d %d %d\n", *p1, *p2, *p3, *(p1 + 1), *p1 + 1);
    return 0;
}
```

6. Što će se ispisati sljedećim programom:

```
#include <stdio.h>
int main (void) {
    int polje[3][2] = {3, 6, 9, 12, 15, 18};
    int *p = polje[0];
    int i1 = *p;
    int i2 = (*p)++;
    int i3 = *p;
    int i4 = *++p;
    int i5 = *p;
    int i6 = *p++;
    int i7 = *(p-1);
    int i8 = *p;
    printf ("%d %d %d %d %d %d %d %d\n", i1, i2, i3, i4, i5, i6, i7, i8);
    return 0;
}
```

7. Što će se ispisati sljedećim programom:

```
#include <stdio.h>
int main (void) {
    int polje[3][2] = {1, 2, 3, 4, 5, 6};
    int *p = &polje[1][1], i = 5;
    printf ("%d\n", *(p + --i));
    printf ("%d\n", *--p + --i);
    return 0;
}
```

8. Izrazima zamijenite upitnike u printf naredbi kojom će se ispisati članovi glavne dijagonale matrice x. Članovima matrice **mora** se pristupiti preko pokazivača p:

```
#include <stdio.h>
int main(void) {
    int x[4][4] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16};
    int *p = &x[0][0];
    printf("%d %d %d %d\n", ?, ?, ?, ?);
    return 0;
}
```

9. S tipkovnice upisati cijeli broj **n** iz intervala [5, 10]. Generirati kvadratnu matricu dimenzija **n x 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. **Elementima matrice pristupati isključivo pomoću pokazivača.** 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
```

Ovaj zadatak se od zadatka br. 5 iz prethodne vježbe razlikuje jedino u tome što se elementima matrice mora pristupiti pomoću pokazivača.

10. 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. Zatim jednodimenzijsko polje **nps** napuniti vrijednostima najmanjih članova matrice u pojedinim stupcima, a jednodimenzijsko polje **npr** napuniti vrijednostima najmanjih članova matrice u pojedinim retcima. Ispisati učitanu matricu, a zatim članove polja **nps** i članove polja **npr**. Članovima svih polja mora se pristupati isključivo preko pokazivača.

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
---	---	---	---

Ovaj zadatak se od zadatka br. 6 iz prethodne vježbe razlikuje jedino u tome što se elementima matrice mora pristupati pomoću pokazivača.

Rješenja

Rješenje 3. zadatka

Neispravna je naredba

```
c = &y;
```

`c` je "pokazivač na `short`", a pridružuje mu se adresa varijable koja je tipa `int`.

Rješenje 8. zadatka

Kad bi bilo dopušteno članovima polja pristupiti pomoću indeksa, tada bi se napisalo:

```
printf("%d %d %d %d\n", x[0][0], x[1][1], x[2][2], x[3][3]);
```

Članovima polja se također može pristupiti preko pokazivača. Pokazivač `p` sadrži adresu elementa `x[0][0]`. Do adrese elementa `x[1][1]` dolazimo tako da preskočimo sve elemente u prvom retku (ima ih $1 \cdot 4$) i jedan element iz drugog retka, tj. adresi `p` pribrojimo $1 \cdot 4 + 1$. Do adrese elementa `x[2][2]` dolazimo tako da preskočimo sve elemente u prva dva retka (ima ih $2 \cdot 4$) i dva elementa iz trećeg retka, tj. adresi `p` pribrojimo $2 \cdot 4 + 2$. Itd.

```
printf("%d %d %d %d\n",  
      *(p+4*0+0), *(p+4*1+1),  
      *(p+4*2+2), *(p+4*3+3));
```

ili

```
printf("%d %d %d %d\n", *(p), *(p+5), *(p+10), *(p+15));
```

Rješenje 9. zadatka

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

int main(void) {
    int i, j, n;
    int mat[MAXDIM][MAXDIM];
    int *p = &mat[0][0]; /* ili: int *p = mat[0] */

    /* 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)
                *(p + i*MAXDIM + j) = 1;
            else
                *(p + i*MAXDIM + j) = 8;

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

    return 0;
}
```

Česta pogreška pri rješavanju ovog zadatka (i sličnih zadataka) jest da se članu matrice `mat[i][j]` pokuša pristupiti pomoću izraza

`*(p + i*n + j)`

To je pogrešno! Redak matrice `mat` sadrži `MAXDIM` članova, a ne `n` članova. Izraz bi bio dobar jedino u slučaju kada korisnik za `n` upiše vrijednost jednaku `MAXDIM`.

Rješenje 10. zadatka

```
#include <stdio.h>

#define MAXRED 6
#define MAXSTUP 10

int main(void) {
    int i, j, m, n;
    int mat[MAXRED][MAXSTUP], *pmat = &mat[0][0];
    int nps[MAXSTUP], *pnps = &nps[0], npr[MAXRED], *pnpr = &npr[0];

    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", pmat + i*MAXSTUP+j);

    /* odredi najmanji clan u svakom stupcu */
    for (j = 0; j < n; ++j) {
        /* u svakom stupcu j obavi sljedece */

        /* pretpostavi da je prvi clan u j-tom stupcu najmanji clan tog stupca */
        *(pnps + j) = *(pmat + 0*MAXSTUP+j);

        for (i = 1; i < m; ++i)
            if (*(pmat + i*MAXSTUP+j) < *(pnps + j))
                *(pnps + j) = *(pmat + i*MAXSTUP+j);
    }

    /* odredi najmanji clan u svakom retku */
    for (i = 0; i < m; ++i) {
        /* u svakom retku i obavi sljedece */

        /* pretpostavi da je prvi clan u i-tom retku najmanji clan tog retka */
        *(pnpr + i) = *(pmat + i*MAXSTUP+0);
        for (j = 1; j < n; ++j)
            if (*(pmat + i*MAXSTUP+j) < *(pnpr + i))
                *(pnpr + i) = *(pmat + i*MAXSTUP+j);
    }

    printf("\nmatrica:\n");
    for (i = 0; i < m; ++i) {
        for (j = 0; j < n; ++j)
            printf("%4d", *(pmat + i*MAXSTUP+j));
        printf("\n");
    }
    printf("\npolje nps:\n");
    for (j = 0; j < n; ++j)
        printf("%4d", *(pnps + j));
    printf("\n");

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

    return 0;
}
```

Česte pogreške pri rješavanju ovog zadatka (i sličnih zadataka) jesu da se članu matrice `mat[i][j]` pokuša pristupiti pomoću izraza

```
*(pmat + i*n + j)
```

ili

```
*(pmat + i*MAXRED + j)
```