

# PiPI μMASS 1

## Zadatak #1

Napisati funkciju koja prima polje cijelih brojeva i veličinu polja kao argumente. Funkcija treba vratiti vrijednost najmanjeg člana polja. Nije potrebno napisati glavni program.

```
#include <stdio.h>

int najmanji(int *polje, int velicina) {
    int i, najmanji;

    najmanji = polje[0];
    for (i = 1; i < velicina; ++i) {
        if (polje[i] < najmanji)
            najmanji = polje[i];
    }
    return najmanji;
}
```

## Zadatak #2

Napisati funkciju koja prima polje cijelih brojeva i veličinu polja kao argumente. Polje treba popuniti nasumičnim vrijednostima u rasponu od 7 do 55. Napisati glavni program koji stvara polje od deset cijelih brojeva i poziva ovu funkciju.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void popuniPolje(int *polje, int velicina) {
    int i;
    for (i = 0; i < velicina; ++i)
        polje[i] = (rand() % (55 - 7 + 1)) + 7;
}

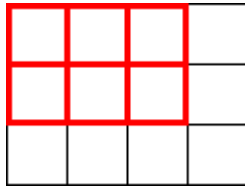
int main() {
    int polje[10], i;

    srand(time(NULL));
    popuniPolje(polje, 10);
    for (i = 0; i < 10; ++i)
        printf("%d ", polje[i]);

    return 0;
}
```

### Zadatak #3

Napisati funkciju koja prima matricu, veličinu matrice i maksimalni broj stupaca kao argumente. Funkcija treba vratiti sumu svih elemenata matrice. Prvo napisati prototip (potpis) funkcije, onda funkciju `main` i tek onda tijelo tražene funkcije.



Veličina matrice: 4x3  
Iskorištena veličina: 3x2

**Argumenti funkcije:**  
Veličina matrice: 3x2  
Maksimalni broj stupaca: 4

```
#include <stdio.h>

#define MAXR 10
#define MAXS 10

int suma(int *matrica, int redaka, int stupaca, int maxStupaca);

int main() {
    int matrica[MAXR][MAXS];
    int redaka, stupaca, i, j;

    scanf("%d %d", &redaka, &stupaca);
    for (i = 0; i < redaka; ++i) {
        for (j = 0; j < stupaca; ++j)
            scanf("%d", &matrica[i][j]);
    }
    printf("%d\n", suma(&matrica[0][0], redaka, stupaca, MAXS));
    return 0;
}

int suma(int *matrica, int redaka, int stupaca, int maxStupaca) {
    int i, j;
    int suma = 0;

    for (i = 0; i < redaka; ++i) {
        for (j = 0; j < stupaca; ++j) {
            suma += matrica[i * maxStupaca + j];
        }
    }
    return suma;
}
```

## Zadatak #4

Napisati funkciju koja prima matricu, veličinu matrice i maksimalni broj stupaca kao argumente. Matricu treba popuniti nasumičnim vrijednostima u rasponu od 0 do 99 tako da redci s parnim indeksom sadržavaju parne brojeve, a s neparnim indeksom neparne (indeksi počinju od nule). Nije potrebno napisati glavni program.

### Izlaz:

```
10 18 46 92 74
27 71 53 21 19
44 36 80 14 54
43 87 91 45 23
```

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void popuniMatricu(int *matrica, int redaka, int stupaca, int maxStupaca) {
    int i, j;
    for (i = 0; i < redaka; ++i) {
        for (j = 0; j < stupaca; ++j) {
            if (i % 2 == 0)
                matrica[i * maxStupaca + j] = (rand() % 50) * 2;
            else
                matrica[i * maxStupaca + j] = (rand() % 50) * 2 + 1;
        }
    }
}

/*
 * Pretpostavlja se da glavni program sadrži:
 * srand(time(NULL));
 */
```

