



1. zadatak

Napisati program koji učitava cijeli broj n sve dok se ne unese broj u intervalu $[1,100]$, a zatim učitava n članova cjelobrojnog jednodimenzionalnog polja.

Za svaki učitani član polja treba ispisati s koliko je brojeva u tom polju koji su učitani nakon njega (imaju veći indeks) djeljiv.

Npr.

$n=8$ i učitani su brojevi: 1, 4, 6, 2, 3, 0, 2, 2
program treba ispisati: 0, 3, 4, 2, 0, 2, 1, 0



Nadopunite (1)

```
#include <stdio.h>
int main(){
    int polje[100];
    int n, i, j, brDjel;
    do{
        scanf("%d", &n);
    }while(_____);
    for(____; i<n; i++){
        scanf("%d", _____);
    }
}
```

Napisati program koji učitava cijeli broj n sve dok se ne unese broj u intervalu $[1,100]$, a zatim učitava n članova cjelobrojnog jednodimenzionalnog polja.



Nadopunite (2)

```
for(i=0;i<n;____){
    brDjel = 0;
    for(____; j<n; j++){
        if (polje[j] && (polje[i] % polje[j] == 0))
            _____;
    }
    printf("%d ", brDjel);
}
return 0;
}
```

Za svaki učitani član polja treba ispisati s koliko je brojeva u tom polju koji su učitani nakon njega (imaju veći indeks) djeljiv.

npr. n=8 i učitani su brojevi:
1, 4, 6, 2, 3, 0, 2, 2
program treba ispisati:
0, 3, 4, 2, 0, 2, 1, 0



Rješenje (1)

```
#include <stdio.h>
int main(){
    int polje[100];
    int n, i, j, brDjel;
    do{
        scanf("%d", &n);
    }while(n<1 || n>100);
    for(i=0; i<n; i++){
        scanf("%d", &polje[i]);
    }
}
```

Napisati program koji učitava cijeli broj n sve dok se ne unese broj u intervalu [1,100], a zatim učitava n članova cjelobrojnog jednodimenzionalnog polja.



Rješenje (2)

```
for(i=0;i<n;i++){  
    brDjel = 0;  
    for(j=i+1; j<n; j++){  
        if (polje[j] && (polje[i] % polje[j] == 0))  
            brDjel++;  
    }  
    printf("%d ", brDjel);  
}  
return 0;  
}
```

Za svaki učitani član polja treba ispisati s koliko je brojeva u tom polju koji su učitani nakon njega (imaju veći indeks) djeljiv.

npr. n=8 i učitani su brojevi:
1, 4, 6, 2, 3, 0, 2, 2
program treba ispisati:
0, 3, 4, 2, 0, 2, 1, 0



2. zadatak

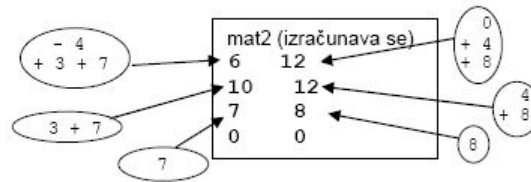
Napisati glavni program u kojem se definiraju dvije cjelobrojne matrice `mat1` i `mat2` (matrice smiju imati najviše 50 redaka i 100 stupaca). Pri definiranju dimenzija matrica obavezno koristiti simboličke konstante.

S tipkovnice učitavati stvarne dimenzije matrice (broj redaka `m` i broj stupaca `n`) dok god ne budu ispravno zadane. U matricu `mat1` po retcima učitati $m \times n$ elemenata.

Elemente matrice `mat2` treba izračunati na temelju elemenata matrice `mat1`. Element matrice `mat2` koji se nalazi u retku `r` i stupcu `s` izračunava se kao suma elemenata matrice `mat1` koji se nalaze ispod retka `r` u stupcu `s`.

Ispisati matricu `mat2`.

mat1 (učitava se)	
2	6
-4	0
3	4
7	8



Nadopunite (1)

Napisati glavni program u kojem se definiraju dvije cjelobrojne matrice `mat1` i `mat2` (matrice smiju imati najviše 50 redaka i 100 stupaca).

S tipkovnice učitavati stvarne dimenzije matrice (broj redaka `m` i broj stupaca `n`) dok god ne budu ispravno zadane.

```
#include<stdio.h>
#define _____
#define _____

int main() {
    int i, j, k, suma, m, n;
    int mat1[MAXRED][MAXSTUP], mat2[MAXRED][MAXSTUP];
    do {
        printf("Upisite broj redaka: ");
        scanf("%d", &m);
        printf("Upisite broj stupaca: ");
        scanf("%d", _____);
    } while ( _____ || _____ || n < 1 || _____ );
```



Rješenje (1)

Napisati glavni program u kojem se definiraju dvije cjelobrojne matrice mat1 i mat2 (matrice smiju imati najviše 50 redaka i 100 stupaca). S tipkovnice učitavati stvarne dimenzije matrice (broj redaka m i broj stupaca n) dok god ne budu ispravno zadane.

```
#include<stdio.h>
#define MAXRED 50
#define MAXSTUP 100

int main() {
    int i, j, k, suma, m, n;
    int mat1[MAXRED][MAXSTUP], mat2[MAXRED][MAXSTUP];
    do {
        printf("Upisite broj redaka: ");
        scanf("%d", &m);
        printf("Upisite broj stupaca: ");
        scanf("%d", &n);
    } while (m < 1 || m > 50 || n < 1 || n > 100);
```



Nadopunite (2)

U matricu mat1 po retcima učitati m × n elemenata.

```
printf("Upisite clanove matrice po retcima:\n");
for (i = 0; i < __; i++)
    for (j = 0; j < __; j++){
        printf("mat1[%d][%d]=", i, j);
        scanf("%d", _____);
    }
```



Rješenje (2)

U matricu mat1 po retcima učitati $m \times n$ elemenata.

```
printf("Upisite clanove matrice po retcima:\n");
for (i = 0; i < m; i++)
    for (j = 0; j < n; j++){
        printf("mat1[%d][%d]=", i, j);
        scanf("%d", &mat1[i][j]);
    }
```



Nadopunite (3)

Element matrice mat2 koji se nalazi u retku r i stupcu s izračunava se kao suma elemenata matrice mat1 koji se nalaze ispod retka r u stupcu s.

```
for (i = 0; i < m; i++)
    for (j = 0; j < n; j++) {
        suma = __;
        for (k = _____; k < m; k++)
            suma += mat1[___][___];
        mat2[i][j] = _____;
    }
```



Rješenje (3)

Element matrice `mat2` koji se nalazi u retku `r` i stupcu `s` izračunava se kao suma elemenata matrice `mat1` koji se nalaze ispod retka `r` u stupcu `s`.

```
for (i = 0; i < m; i++)
    for (j = 0; j < n; j++) {
        suma = 0;
        for (k = i + 1; k < m; k++)
            suma += mat1[k][j];
        mat2[i][j] = suma;
    }
```



Nadopunite (4)

Ispisati matricu `mat2`.

```
for (i = 0; i < m; i++) {
    for (j = 0; _____; j++)
        printf("%d ", _____);
    printf("\n_____");
}
return 0;
}
```



Rješenje (4)

Ispisati matricu mat2.

```
for (i = 0; i < m; i++) {
    for (j = 0; j < n; j++)
        printf("%d ", mat2[i][j]);
    printf("\n");
}
return 0;
}
```



Zadatak 3 (PZI 30.1.2007. 1. zad)

Napisati funkciju koja za zadano jednodimenzionalno cjelobrojno polje u pozivajući program vraća aritmetičku sredinu i sumu članova polja.
Napisati glavni program u kojem je potrebno s tipkovnice uz kontrolu učitati broj elemenata polja (mora biti u intervalu [5,100]) i zatim elemente polja. Koristeći zadanu funkciju ispisati aritmetičku sredinu i sumu članova polja.

Zadatak 3 - nadopunite!

```
#include <stdio.h>
void arSredSuma( _____ ) {
    int i=0;
    /* DOVRŠITE FUNKCIJU arSredSuma koja u pozivajući program
    vraća aritmetičku sredinu i sumu članova polja */
}
int main(){
    int polje[100];
    int br, i, suma;
    float arsred;
    do {
        printf("Unesite broj elemenata polja:");
        scanf("%d", &br);
    } while ( _____ ); // br je u intervalu [5,100]
    printf("\nUnos elemenata polja:\n");
    for (i= 0; i < br; ++i){
        printf("polje[%d]:",i);
        scanf("%d",&polje[i]);
    }
    arSredSuma( _____ );
    printf("\nsuma=%d arsred=%f\n", suma, arsred);
    return 0;
}
```

Zadatak 3 - rješenje

```
#include <stdio.h>
void arSredSuma(int *niz, int br, int *suma, float *arsred){
    int i=0;
    *suma = 0;
    for (i = 0; i < br ; ++i){
        *suma += *(niz+i);           // ili: niz[i]
    }
    *arsred = (float)*suma/br;
}
int main(){
    int polje[100];
    int br, i, suma;
    float arsred;
    do {
        printf("Unesite broj elemenata polja:");
        scanf("%d", &br);
    } while (br<5 || br>100);
    printf("\nUnos elemenata polja:\n");
    ...
    arSredSuma(polje, br, &suma, &arsred); //ili: &polje[0]
    printf("\nsuma=%d arsred=%f\n", suma, arsred);
    return 0;
}
```