

## 2. međuispit iz Programiranja i programskog inženjerstva 5. prosinca 2006.

Napomene za sve zadatke:

- Nije dopušteno korištenje goto naredbe, te statičkih i globalnih varijabli.
- **Studentima na pitanja ne smiju odgovarati asistenti koji ih čuvaju u učionici. Na pitanja će odgovarati isključivo asistenti koji za vrijeme ispita obilaze prostorije**

1. (6 bodova) Napisati **glavni program** (tj. funkciju `main`) koji će s tipkovnice učitavati cijele brojeve dok god se ne upiše negativan broj. Poznato je da korisnik programa sigurno neće upisati više od 1000 nenegativnih brojeva.

Za svaki cijeli broj  $k \in [1, 50]$ , na zaslone ispisati koliko je učitanih brojeva djeljivo brojem  $k$ . Učitani negativan broj koji uzrokuje prekid učitavanja ne uzima se u obzir (vidi se iz primjera).

### Primjer izvođenja programa:

```
Upisite brojeve:
23 1 6 490 11 0 4900 12 -2
djeljivih s 1 ima:8
djeljivih s 2 ima:5
djeljivih s 3 ima:3
... itd.
djeljivih s 49 ima:3
djeljivih s 50 ima:2
```

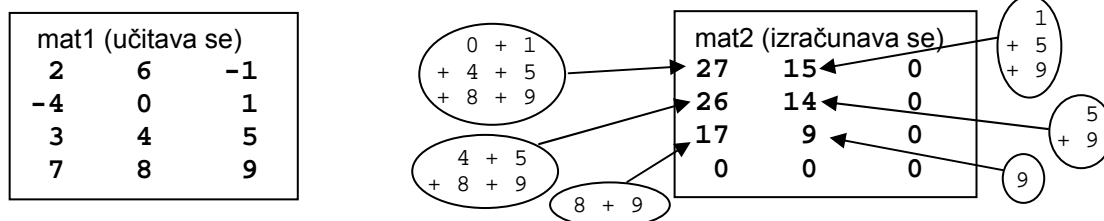
2. (6 bodova) Napisati **funkciju** `prethodnik` koja u pozivajući program vraća ASCII prethodnik zadanog znaka (tj. znak koji se u ASCII tablici nalazi na poziciji neposredno ispred zadanog znaka). Npr. ako se funkciji zada znak 'K', funkcija treba vratiti znak 'J'. Nije potrebno voditi računa o tome što će se desiti ako se funkciji zada znak '\0'.

Zatim napisati **glavni program** u kojem će se s tipkovnice, pomoću funkcije `gets`, učitati niz znakova (može se pretpostaviti da korisnik sigurno neće zadati niz dulji od 50 znakova). Na zaslon ispisati učitani niz znakova. Zatim niz treba izmijeniti: iz niza izbaciti znakove praznine (*blank*), a preostale znakove zamijeniti njihovim ASCII prethodnicima (ASCII prethodnike odrediti pomoću funkcije `prethodnik`). Na kraju, **nakon što je niz izmijenjen**, na zaslon ispisati izmijenjeni niz.

### Primjer izvođenja programa:

```
Upisite niz:Ovo- je niz+
Ucitani niz:Ovo- je niz+
Izmijenjeni niz: Nun,idmhy*
```

3. (6 bodova) 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  $r$  i **desno** od stupca  $s$ . Nije potrebno ispisivati matrice. Primjer:



4. (7 bodova) Napisati **prototip funkcije** i **funkciju** `rastavi` koja za zadani pozitivni četveroznamenasti broj u pozivajući program vraća **dva cijela broja**: cijeli broj koji predstavlja prve dvije znamenke i cijeli broj koji predstavlja zadnje dvije znamenke zadanog broja. Npr. za zadani broj 3725 funkcija vraća brojeve 37 i 25, a za zadani broj 3007 vraća brojeve 30 i 7.

Napisati **glavni program** koji s tipkovnice učitava samo jedan cijeli broj. Ako je učitani pozitivan četveroznamenasti broj, tada učitani broj rastaviti pomoću funkcije `rastavi` i ispisati rezultat, u suprotnom ispisati poruku "Broj nije ispravan".

# RJEŠENJA:

## 1. Zadatak

```
#include <stdio.h>
#define MAXCLAN 50

int main() {
    int djeljiv[MAXCLAN] = {0};
    int i, broj;

    printf("Upisite brojeve:\n");
    do {
        scanf("%d", &broj);
        if (broj >= 0)
            for (i = 1; i <= 50; i++)
                if (broj % i == 0)
                    djeljiv[i-1]++;
    } while (broj >= 0);

    for (i = 1; i <= 50; i++)
        printf("djeljivih s %d ima:%d\n", i, djeljiv[i-1]);
    return 0;
}
```

iii

```
#include <stdio.h>
#define MAXPOLJE 1000

int main() {
    int i, j, broj, brojDjeljivih, brojUcitanih = 0;
    int ulaznoPolje[MAXPOLJE];

    printf("Upisite brojeve:\n");
    do {
        scanf("%d", &broj);
        if (broj >= 0)
            ulaznoPolje[brojUcitanih++] = broj;
    } while (broj >= 0);

    for (i = 1; i <= 50; i++) {
        brojDjeljivih = 0;
        for (j = 0; j < brojUcitanih; j++)
            if (ulaznoPolje[j] % i == 0)
                brojDjeljivih++;
        printf("djeljivih s %d ima:%d\n", i, brojDjeljivih);
    }
    return 0;
}
```

## 2. Zadatak

```
#include <stdio.h>
#define MAXZNAK 50

char prethodnik(char znak) {
    return znak - 1;
}

int main() {
    char niz[MAXZNAK+1];
    int i = 0, j = 0;
    printf("Upisite niz:");
    gets(niz);
    printf("Ucitani niz:%s\n", niz);

    while (niz[i]) {
        if (niz[i] != ' ') {
            niz[j] = prethodnik(niz[i]);
            j++;
        }
        i++;
    }
    niz[j] = '\0';

    printf("Izmijenjeni niz:%s\n", niz);
    return 0;
}
```

### 3. Zadatak

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

int main() {
    int i, j, i2, j2, suma, m, n;
    int mat1[MAXRED][MAXSTUP], mat2[MAXRED][MAXSTUP];
    do {
        printf("Upisite m i n:");
        scanf("%d %d", &m, &n);
    } while (m < 1 || m > 50 || n < 1 || n > 100);

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

    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            scanf("%d", &mat1[i][j]);

    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++) {
            suma = 0;
            for (i2 = i + 1; i2 < m; i2++)
                for (j2 = j + 1; j2 < n; j2++)
                    suma += mat1[i2][j2];
            mat2[i][j] = suma;
        }
    /* kontrolni ispis (ne trazi se u zadatku!)
       4 3      2 6 -1 -4 0 1 3 4 5 7 8 9
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++)
            printf("%d ", mat2[i][j]);
        printf("\n");
    }
    */
    return 0;
}
```

### 4. Zadatak

```
#include <stdio.h>

void rastavi(int broj, int *prviDio, int *drugiDio);

int main () {
    int n, prvi, drugi;
    printf("Upisite 4-znamenkasti pozitivni broj: ");
    scanf("%d", &n);
    if (n >= 1000 && n <= 9999) {
        rastavi(n, &prvi, &drugi);
        printf("Trazeni rastav broja %d jest %d i %d\n", n, prvi, drugi);
    }
    else
        printf("Broj nije ispravan\n");
    return 0;
}

void rastavi(int broj, int *prviDio, int *drugiDio) {
    *prviDio = broj / 100;
    *drugiDio = broj % 100;
    return;
}
```