

Programiranje

vježba za 2MI

1. zadatak

- Napisati glavni program koji će s tipkovnice učitavati cijele brojeve dok god se ne upiše nula.
- Za svaki učitani broj provjeriti da li ga je moguće dobiti kao sumu **svih naknadno** učitanih brojeva.

Primjer izvođenja programa:

Upisite brojeve:

14 7 3 4 0

14: DA

7: DA

3: NE

4: NE

1. zadatak

```
#include <stdio.h>
#define MAXNIZ 100

int main() {

    int niz[MAXNIZ];
    int i=0, j, k, suma;

    printf("Upisite brojeve: ");
    do {
        scanf("%d", &niz[i++]);
    } while (niz[i-1]!=0);

    for (j=0; j<i-1; j++) {
        suma = 0;
        for (k=j+1; k<i; k++)
            suma+= niz[k];
        printf("%d %s\n", niz[j], (niz[j]==suma)?"DA":"NE");
    }
}
```

2. zadatak

- Napisati program koji pomoću gets naredbe učitava niz znakova. Pretpostaviti da učitani niz neće imati više od 80 znakova. Ispisati koje se **malo** slovo abecede pojavilo najviše puta u učitanoj nizu znakova, te ispisati broj pojavljivanja tog slova. Ukoliko su se različita slova pojavila jednak broj puta, ispisati ono slovo koje se nalazi bliže početku abecede.
- *Primjer:*
- za učitani niz: PIPi je moj najdrazi predmet
- treba ispisati: Slovo e se pojavilo 3 puta

2. zadatak

```
#include <stdio.h>
#define MAXZNAK 80
int main() {

    char niz[MAXZNAK+1];
    int brojSlova['z'-'a' +1] = {0};
    int i, d, index, max = 0;

    printf("Unesite recenicu: ");
    gets(niz);
    d = 0;

    //mala slova
    d = 0;
    while (niz[d]){
        if (niz[d] >= 'a' && niz [d] <= 'z') {
            index = niz[d] - 'a';
            brojSlova[index]++;
        }
        d++;
    }
}
```

2. zadatak

```
for (i = 'a'; i <= 'z'; i++){
    if (brojSlova[i-'a'] > brojSlova[max])
        max = i-'a';
}

printf ("Slovo %c, pojavilo se %d puta.\n",
        max + 'a', brojSlova[max]);

return 0;
}
```

3. zadatak

- Napisati **funkciju** prethodnik koja u pozivajući program vraća ASCII prethodnik zadanog znaka.
 - Npr. za 'C' fja vrati 'B', za '4' vrati '3' itd...
- Napisati **glavni program** u kojem će se s tipkovnice, učitati niz znakova (do 50 znakova). Ispisati originalni niz, a ispod njega ispisati niz sa ASCII prethodnicima svih znakova iz učitanoog niza, **osim razmaka**.
 - Npr. za upisano "Stvarno volim PIPI", program će ispisati:

```
Stvarno volim PIPI  
Rsu`qmn unkh1 OHOH
```

3. zadatak

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

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

int main(){
    char niz[MAXZNAK+1]; int i=0;
    printf("Niz: ");
    gets(niz);
    printf("%s\n", niz);
    while(niz[i]) {
        if (niz[i]!=' ')
            printf ("%c",prethodnik(niz[i]));
        else
            printf (" ");
        i++;
    }
    return 0;
}
```

4. zadatak

- Napisati program kojim se učitava dvodimenzionalna matrica cijelih brojeva maksimalnih dimenzija 10 x 15.

Učitati stvarne dimenzije matrice (broj redaka i broj stupaca).

Učitati elemente matrice.

Ukoliko je suma elementa matrice jednaka 0 ispisati: Suma elemenata matrice je 0. Inače ispisati sve elemente manje od aritmetičke sredine.

4. zadatak

```
#include <stdio.h>
#define MAXRED 10
#define MAXSTUP 15

int main() {

int mat[MAXRED][MAXSTUP];
int r, s;
int i, j, suma=0;
float as;

printf("Unesi broj redaka i broj stupaca: ");
scanf("%d %d", &r, &s);

for (i=0; i<r; i++) {
    for (j=0; j<s; j++) {
        scanf ("%d", &mat[i][j]);
        suma = suma + mat[i][j];
    }
}
```

4. zadatak

```
if (suma==0) {
    printf("Suma elementa matrice je 0");
}

else {
    as=(float) suma/(r*s);
    printf("Brojevi manji od %f su: ", as);

    for (i=0;i<r;i++)
        for (j=0;j<s;j++)
            if (mat[i][j]<as)
                printf("%d , ", mat[i][j]);

}

return 0;

}
```

5. zadatak

- Napisati program u kojem se definiraju dvije cjelobrojne matrice maksimalnih dimenzija 20 x 20. Pri definiranju dimenzija matrica obavezno koristiti simboličke konstante.
- S tipkovnice učitati stvarne dimenzije prve matrice (broj redaka i broj stupaca), pri čemu postupak učitavanja dimenzija treba ponavljati sve dok se ne učitaju valjane dimenzije.
- Učitati elemente prve matrice.
- Elemente druge matrice treba izračunati na temelju elemenata prve matrice, na način da se $mat2[i,j]$ računa kao suma svih elemenata u i -tom retku matrice $mat1$ **minus** suma svih elemenata j -tog stupca $mat1$.

5. zadatak

```
#include <stdio.h>
#define MAXRED 20
#define MAXSTUP 20

int main() {

int mat1[MAXRED][MAXSTUP], mat2[MAXRED][MAXSTUP];
int r, s, i, j, si, sj;

do {
    printf("Unesite broj redaka i broj stupaca: ");
    scanf("%d %d", &r, &s);
} while (r<0 || s<0 || r>MAXRED || s>MAXSTUP);

printf("Unesite elemente matrice mat1: \n");
for (i=0; i<r; i++) {
    for (j=0; j<s; j++) {
        printf("mat[%d,%d]: ", i, j);
        scanf ("%d", &mat1[i][j]);
    }
}
}
```

5. zadatak

```
for (i=0;i<r;i++) {
    for (j=0;j<s;j++) {
        mat2[i][j] = 0;
        for (sj=0; sj<s; sj++) {
            mat2[i][j] += mat1[i][sj];
        }
        for (si=0; si<r; si++)
            mat2[i][j] -= mat1[si][j];
        printf("%d ", mat2[i][j]);
    }

    printf("\n");
}

return 0;
}
```

6. zadatak

- Napisati funkciju koja će za zadani višeznamenasti cijeli broj u glavni program vratiti **dva broja** - sumu i umnožak njegovih znamenki.
- Ukoliko predani parametar nije višeznamenasti broj, funkcija preko povratne vrijednosti signalizira pogrešku, tj. vraća -1, inače vrati 0.

6. zadatak

```
#include <stdio.h>

int racunajSU(int br, int *suma, int *umnozак) {
    if (br < 0 ) br = -br;
    if (br < 10) return -1;
    *suma = 0; *umnozак = 1;
    while (br) {
        *suma += br%10;
        *umnozак *= br%10;
        br/=10;
    }
    return 0;
}
```

6. zadatak

```
int main () {
    int broj, s, u;
    printf("Upisite broj: ");
    scanf("%d", &broj);
    if (racunajSU(broj, &s, &u) != -1)
        printf("Za %d suma zn. je %d, a umnozак %d.\n", broj, s, u);
    else
        printf("Broj nije viseznamenkast\n");

    system("PAUSE");

    return 0;
}
```