

**Zadaci za vježbu za 3.MI**

# Kad bi pogađali...

- funkcije, pokazivači
  - prijenos 2D polja u funkciju
  - ugrađene funkcije, posebice za rad s zn.nizovima
  - slučajni brojevi
  - slijedne datoteke
  - direktne datoteke
- 
- Ali: u završni međuispit ulazi cjelokupno gradivo!

# Kako rješavati zadatak

- Pažljivo pročitati zadatak
- Pažljivo pogledati primjera ako postoje – ponekad je zadatak dodatno definiran primjerom
- Rubni uvjeti (prazan zn.niz, zn.niz duljine 1 , matrica 1x1 i sl.)
- Probati na papiru
- Struktura programa

# Kada treba koristiti pokazivače u funkciji?

- Funkcije **preko imena** može vratiti **samo jednu** vrijednost!
- npr. `x2 = pow(x, 2);`
- Što ako želimo funkciju koja će izračunati i korijen i kvadrat zadanog broja?
- 

```
void f1(double x, double *xKv, double *xKo){
    *xKv = pow(x, 2);
    *xKo = pow(x, 1./2);
}
```

```
int main(){
    double x = 2, xKvadrat, xKorijen;
    f1(x, &xKvadrat, &xKorijen);
    return 0;
}
```

```
double f2(double x, double *xKo){
    *xKo = pow(x, 1./2);
    return pow(x, 2);
}
```

```
int main(){
    double x = 2, xKvadrat, xKorijen;
    xKvadrat = f2(x, &xKorijen);
    return 0;
}
```

# Primjer

Napisati funkciju koja će za zadani niz vratiti broj suglasnika i samoglasnika koje niz sadrži. Napisati glavni program koji učitava znakovni niz, pomoću funkcije određuje broj samoglasnika i suglasnika i zatim ispisuje broj samoglasnika i suglasnika na zaslon.

Npr.

Unesite niz: abacus!!!

Zadani niz ima 3 samoglasnika i 3 suglasnika.

Za funkciju koja ne vraća broj samoglasnika/suglasnika nego ispisuje na zaslon – **0 bodova** !!

# Analiza

Moramo vratiti 2 vrijednosti – pomoću pokazivača.

Odredimo da li je slovo samoglasnik, a inače je ...?

# Rješenje

```
#include <stdio.h>
#include <ctype.h>

void brSuglSamogl(char *niz, int *brSamogl, int *brSugl){
    char zn;
    *brSamogl = 0; // OBAVEZNO U FUNKCIJI!!
    *brSugl = 0; // OBAVEZNO U FUNKCIJI!!
    while (*niz) {
        zn = tolower(*niz);
        if (zn == 'a' || zn == 'e' || zn == 'i' || zn == 'o' || zn == 'u' ) {
            ++(*brSamogl);
        } else if (islower(zn)) {
            ++(*brSugl);
        }
        ++niz;
    }
}

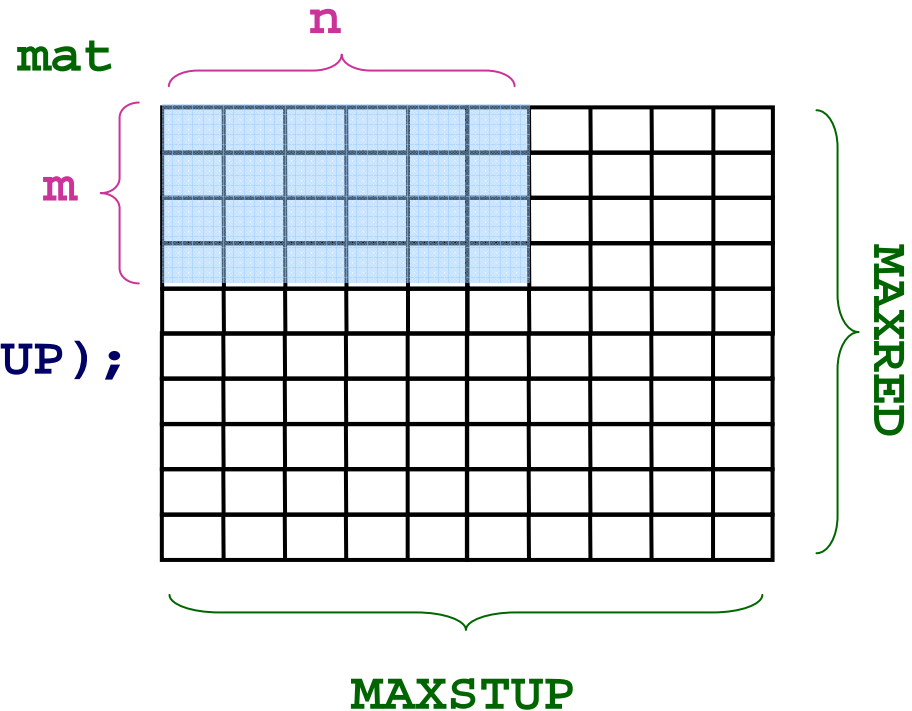
int main(){
    char niz[100];
    int brSamogl, brSugl;
    gets(niz);
    brSuglSamogl(niz, &brSamogl, &brSugl);
    printf("Zad. niz ima %d samoglas. i %d suglasnika.", brSamogl, brSugl);
    return 0;
}
```

# Prijenos matrice u funkciju



# Funkcija s $m \times n$ matricom ako se koristi dio prostora definirane matrice

```
int main() {  
    int mat[MAXRED][MAXSTUP]  
    ...  
    fun(&mat[0][0], m, n, MAXSTUP);  
    ...  
}
```



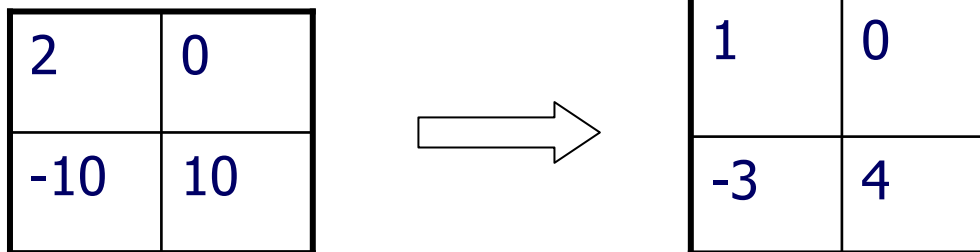
```
tipfun fun(int *p, int m, int n, int brstup) {  
    for (i=0; i<m; i++)  
        for (j=0; j<n; j++)  
            *(p + i*brstup + j) ili p[i*brstup + j]
```

**NE OVAKO:**  $*(p + i*MAXSTUP + j)$  ili  $p[i*MAXSTUP + j]$

# Primjer

Napisati funkciju koja mijenja zadanu cjelobrojnu matricu proizvoljnih dimenzija tako da svaki element postavi na neku slučajnu vrijednost u intervalu:

- ako je  $x_{ij} \geq 0$  onda  $[0, x_{ij}]$
- inače  $[x_{ij}, 0]$



# Slučajni brojevi

`int rand (void) → [0, 32767]`

Kako jednoliko preslikati cijele brojeve  $x$  iz intervala  $[a, b]$  u interval  $[c, d]$ ?

$$y = \underbrace{(x - a) / (b - a + 1) * (d - c + 1)}_{\text{skaliranje}} + \underbrace{c}_{\text{translacija}}$$

tj. `y = (double)rand() / (RAND_MAX+1) * (d-c+1) + c;`

ili (jednostavnije!):

`y = rand() % (d - c + 1) + c`

# Analiza

- "...proizvoljnih dimenzija" -> 4 argumenta!

za svaki element matrice

zapamtiti predznak

odrediti slučajan broj u intervalu  $[0, \text{abs}(m_{ij})]$

$m_{ij} = \text{predznak} * \text{slučajan broj}$

# Rješenje

```
void matSlBroj(int *m, int r, int s, int maxstu){
    int i, j, p, br;
    for (i = 0; i < r; ++i) {
        for (j = 0; j < s; ++j){
            br = m[i * maxstu + j];
            p = (br >= 0) ? 1 : -1; // odredimo predznak
            br *= p; // ovime radimo abs, mogli smo i pomocu f-je
            m[i * maxstu + j] = p * ( rand() % (br - 0 + 1) + 0 );
        }
    }
}
```

## Rješenje – glavna funkcija – što nedostaje!?

```
int main(){
    int m[MAXRED][MAXSTU], i, j, r, s;
    puts("Unesite broj redaka i stupaca <= 10");
    scanf("%d", &r);
    scanf("%d", &s);
    for (i = 0; i < r; ++i) {
        for (j = 0; j < s; ++j) {
            scanf("%d", &m[i][j]);
        }
        printf("\n");
    }
    matSlBroj(&m[0][0], r, s, MAXSTU);
    for (i = 0; i < r; ++i) {
        for (j = 0; j < s; ++j) {
            printf("%3d", m[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

srand ((unsigned) time(NULL));

# Primjer – ugrađene funkcije

Napisati funkciju koja u zadanom znakovnom nizu sve zadane podnizove premješta na kraj niza.

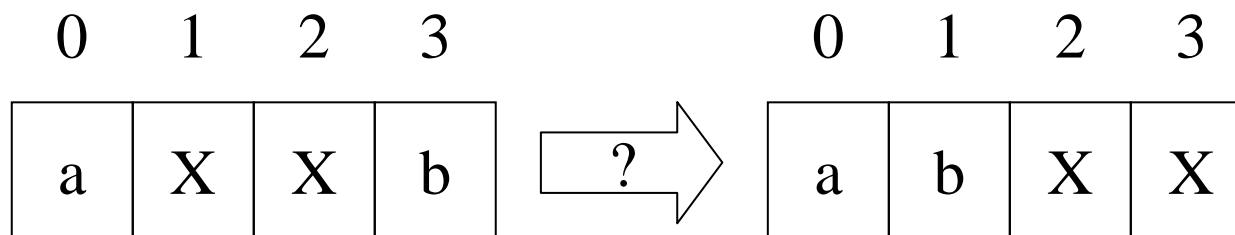
Npr.

za niz AXXBXXC i podniz XX

funkcija treba promijeniti prvi niz u ABCXXXX

# Analiza

- U petlji pomoću `strstr` tražimo zadani podniz. (budući da u zadatku nije ništa rečeno smatra se da je pretraga *case-sensitive*)
- `while ( (p = strstr(niz, podniz)) != NULL ) { ...`
- Kad nađemo zadani podniz premjestimo ga na kraj?





# Rješenje

```
void podnizNaKraj(char *niz, char *podniz){
    char *p;
    int br = 0;
    while ( (p = strstr(niz, podniz)) != NULL ) {
        strcpy(p, p + strlen(podniz));
        ++br;
    }
    while (br--){
        strcat(niz, podniz);
    }
}

int main(){
    char niz[100], podniz[100];
    gets(niz);
    gets(podniz);
    podnizNaKraj(niz, podniz);
    printf("\n%s", niz);
    return 0;
}
```

Datoteke

# Formatirane (tekstualne) datoteke

# Primjer – formatirane

U svakom retku slijedne formatirane datoteke "sprint" nalazi se ime, prezime i rezultat natjecatelja utrke na 100m:

Npr:

```
Maurice Greene 9.87
Ato Boldon    9.99
Obadele Thompson 10.04
...
```

Napisati funkciju koja će u datoteku "ispod10" prepisati zapise o rezultatima ispod 10 sekundi, u istom formatu.

*Napomena: ime i prezime ne sadrži praznine.*

# Rješenje

```
void prebaci(){
    FILE *fin, *fout;
    char im[50], pr[50];
    char c;
    float rez;
    if ((fin=fopen("sprint.txt","r"))==NULL){
        printf("\nNe mogu otvoriti ulaznu datoteku!");
        return;
    }
    if ((fout=fopen("ispod10.txt","w"))==NULL){
        printf("\nNe mogu otvoriti izlaznu datoteku!");
        return;
    }
    while( fscanf(fin,"%s%s%f",im,pr,&rez) == 3 ){
        if (rez<10.){
            fprintf(fout,"%s %s %f\n",im,pr,rez);
        }
    }
    fclose(fin);
    fclose(fout);
}
```

**Neformatirane (binarne) datoteke**

## Primjer – neformatirane (27.9.00.)

Direktna neformatirana datoteka "Syndey2000" sadrži podatke o dizačima utega:

šifra natjecatelja	long
masa	float

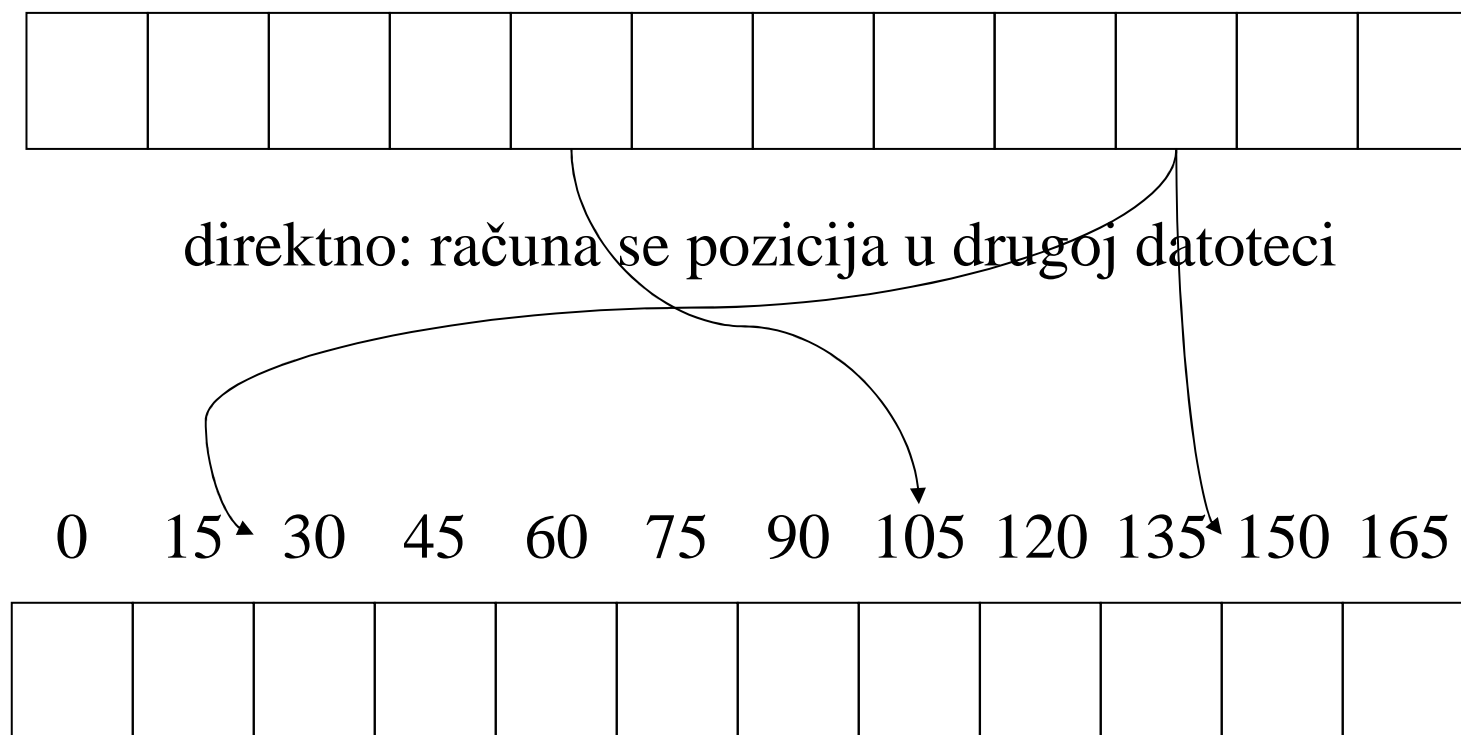
U toj datoteci je za svakog natjecatelja upisana najveća masa koju je podigao. Potrebno je napisati funkciju koja će vratiti maksimalnu, minimalnu i prosječnu podignutu masu.

## Rješenje:

```
void statistika(float *min, float *max, float *prosj){
    cvor zapis;
    int br_zapisa;
    FILE *fin;
    if ((fin=fopen("Sydney2000","rb"))==NULL){
        printf("\nNe mogu otvoriti ulaznu datoteku!");
        return;
    }
    if (fread(&zapis, sizeof(cvor),1,fin)==1){
        *min=*max=*prosj=zapis.masa; // u varijablu *prosj prvo
        br_zapisa=1;                // sumiramo pa na kraju
    } else {                        // podijelimo sa br_zapisa
        printf("\nPrazna datoteka!");
        return;
    }
    while(fread(&zapis, sizeof(cvor),1,fin)==1){
        if (zapis.masa>*max){
            *max=zapis.masa;
        } else if (zapis.masa<*min){
            *min=zapis.masa;
        }
        *prosj+=zapis.masa;
        br_zapisa++;
    }
    *prosj/=br_zapisa;
}
```



# Često se čita jedna datoteka slijedno a druga direktno



`sizeof(struct) = 15 byte`

## Primjer (15.9.05)

Neformatirana datoteka "roditelji.dat" sadrži podatke o roditeljima:

imePrezime	polje od 50 znakova
brDjece	int
sifreDjece	polje od 50 long brojeva

Svaki roditelj ima polje koje pokazuje na njegovu djecu u datoteci "djeca.dat".

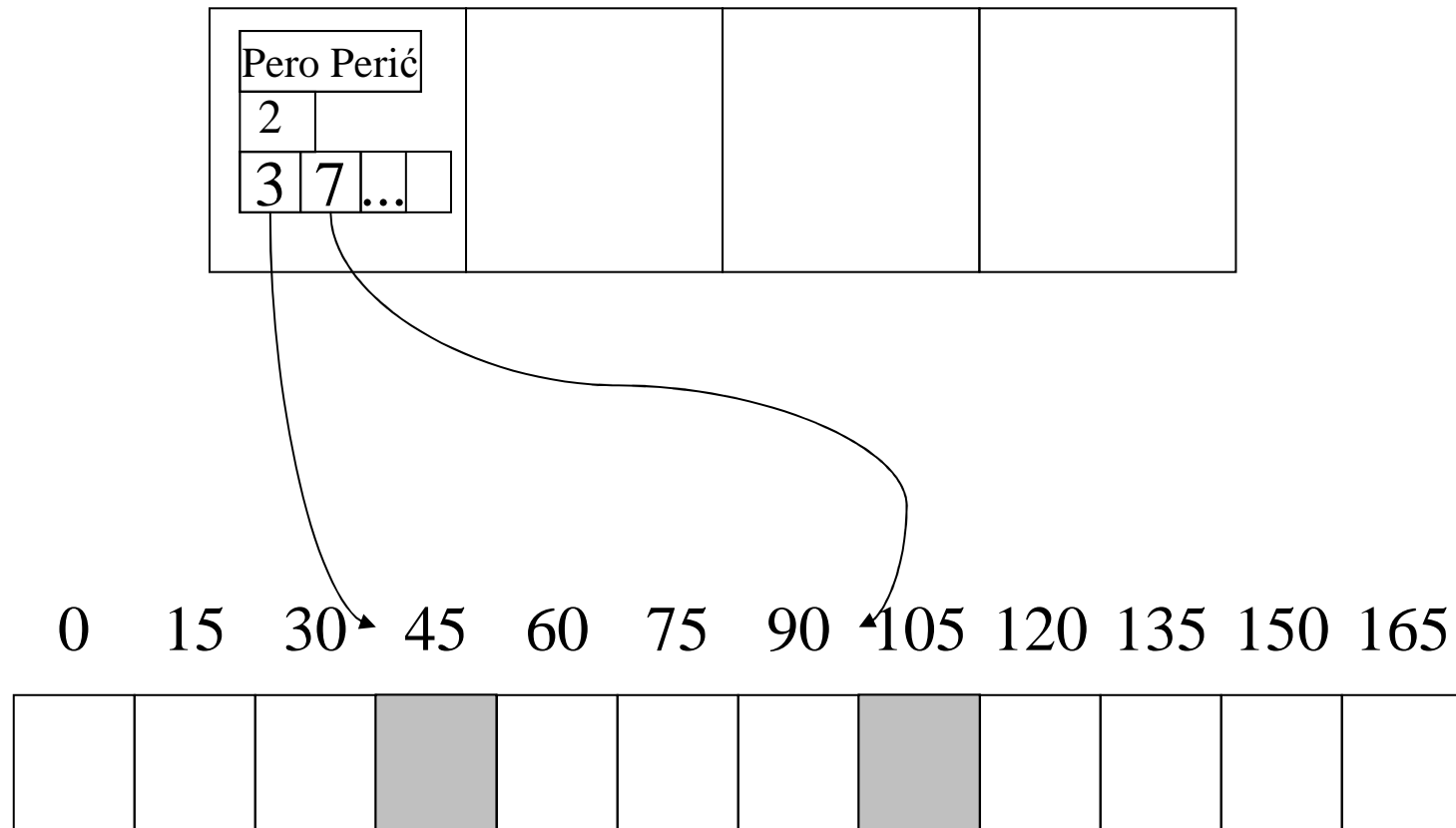
Direktna neformatirana datoteka "djeca.dat" sadrži podatke o tome koliko djeci ispita nedostaje za prolaz godine na fakultetu na kojem studiraju:

sifraDjeteta	long
imePrezime	polje od 50 znakova
brIspita	int

pri čemu sifra\_djeteta odgovara rednom broju zapisa u datoteci.

Potrebno je napisati program koji će ispisati ime i prezime roditelja čijoj djeci ukupno nedostaje najviše ispita za prolaz. Ukoliko ima više takvih, ispisati bilo kojega.

# Analiza



`sizeof(struct) = 15 byte`

## Rješenje (1/2):

```
typedef struct{  
    char        imePrezime[50];  
    int         brDjece;  
    long        sifreDjece[50];  
} roditelj;
```

```
typedef struct{  
    long        sifraDjeteta;  
    char        imePrezime[50];  
    int         brIspita;  
} dijete;
```

## Rješenje (2/2)

```
int main(){
    int i, sum, max=-1;
    char maxRoditelj[50];
    roditelj r;
    dijete d;
    FILE *fr, *fd;
    fr = fopen("roditelji.dat", "rb");
    fd = fopen ("djeca.dat", "rb");
    if (fd==NULL || fr==NULL)
        return;
    while (fread(&r, sizeof(r), 1, fr)==1){
        sum=0;
        for (i=0; i < r.brDjece; i++){
            fseek(fd, sizeof(d) * r.sifreDjece[i], SEEK_SET);
            fread(&d, sizeof(d), 1, fd);
            sum += d.brIspita;
        }
        if (sum > max){
            max = sum;
            strcpy(maxRoditelj, r.imePrezime);
        }
    }
    printf("%s", maxRoditelj);
    return 0;
}
```

# Zadatak s međuispita

Svaki zapis postojeće slijedne formatirane (tekstualne) datoteke "polozio.txt" sadrži šifru studenta (cijeli broj od najviše 5 znamenaka), prezime studenta (niz znakova bez praznina duljine najviše 20 znakova) i broj položenih ispita (cijeli broj iz zatvorenog intervala [0, 10]).

Napisati glavni program koji će za svaki cijeli broj iz intervala [0, 10], na zaslon ispisati broj studenata koji su položili taj broj ispita. U slučaju pogreške kod otvaranja toka podataka (datoteke) prekinuti izvođenje programa.

U primjeru na slici prikazan je mogući izgled datoteke (Horvat je položio 2 ispita, Novak je položio 7 ispita, itd.) i mogući izgled ispisa na zaslonu (niti jedan ispit nije položilo 47 studenata, jedan ispit položilo je 63 studenata, itd.)

## **polozio.txt**

```
12362 Horvat 2
162 Novak 7
52351 Kolar 0
1564 Kos 3
...
```

## **Izgled ispisa na zaslonu:**

```
0 47
1 63
2 78
...
8 0
9 12
10 2
```

# Rješenje

```
#include <stdio.h>

int main () {
    FILE *du;
    int mbr, brojIspita, i;
    int brojStud[11] = {0};
    char prez[20+1];
    du = fopen("polozio.txt", "r");

    if (du == NULL)
        exit(-1);

    while (fscanf(du, "%d%s%d", &mbr, prez, &brojIspita) == 3)
        brojStud[brojIspita] += 1;

    for (i = 0; i <= 10; i++)
        printf("%d %d\n", i, brojStud[i]);

    fclose(du);
    return 0;
}
```

?