

Zadaci za vježbu

Završni ispit
(ugrađene funkcije)

Zadatak 1

- Napisati funkciju čiji je prototip:
void zamijeni (char *nizStari, char *nizNovi);
koja će niz znakova nizStari zadan u formatu "PREZIME, IME" prepisati u niz nizNovi, i to tako da niz nizNovi bude formata "IME PREZIME". Nije dozvoljeno mijenjanje niza nizStari.
- Napisati glavni program u kojem se učitava niz znakova u formatu "PREZIME, IME" (pretpostavite da će niz biti unesen u tom formatu) te ispisuje u formatu "IME PREZIME".
- **Primjer:** Za učitani niz "ANIC, ANA" potrebno je ispisati "ANA ANIC".

Zadatak 1 – rješenje (1. način)

```
void zamijeni (char *nizStari, char *nizNovi)
{
    int duljina;
    char *p;
    p = strchr(nizStari, ',');
    duljina = p - nizStari + 1;

    strcpy(nizNovi, nizStari + duljina + 1);
    strcat(nizNovi, " ");
    strncat(nizNovi, nizStari, duljina - 1);
}
```

Zadatak 1 – rješenje (2. način – 1/2)

```
void zamijeni (char * nizStari, char * nizNovi)
{
    char pom1[100], pom2[100];
    int i = 0, j = 0, k = 0, flag = 0;

    while(nizStari [i] != '\0')
    {
        ...
    }
    pom1[k] = '\0';
    pom2[j] = '\0';

    strcpy(nizNovi, pom2);
    strcat(nizNovi, " ");
    strcat(nizNovi, pom1);
}
```

Zadatak 1 – rješenje (2. način – 2/2)

```
while(nizStari [i] != '\0'){  
  
    if (nizStari [i] == ',') flag = 1;  
  
    if (!flag){  
        pom1[k] = nizStari [i];  
        k++;  
    }  
  
    else if ((flag) && (nizStari [i] != ' ')  
            && (nizStari [i] != ',')){  
        pom2[j] = nizStari [i];  
        j++;  
    }  
    i++;  
}
```

Zadatak 1 – rješenje (gl. program)

```
int main()
{
    char nizStari [100], nizNovi [100];
    gets(nizStari);
    zamijeni(nizStari, nizNovi);
    printf("%s %s\n", nizStari, nizNovi);
    return 0;
}
```

Zadatak 2

- Napisati funkciju koja zadani niz mijenja tako da iz njega izbacuje sva pojavljivanja zadanog podniza. Funkcija vraća 1 ako je podniz pronađen u zadanom nizu, a inače 0.
- Napisati glavni program u kojem se učitavaju niz i podniz, poziva funkcija, te ispisuje niz.
- Primjer :
Niz prije poziva funkcije: Oko za oko, zub za zub.
Niz poslije poziva funkcije, za podniz "za":
Oko oko, zub zub.

Zadatak 2 – rješenje

```
int IzbaciPodniz(char *niz, char *podniz) {
    char *pom = strstr(niz, podniz);
    if (pom == NULL) return 0;
    while (pom != NULL) {
        strcpy(pom, pom + strlen(podniz));
        pom = strstr(pom, podniz);
    }
    return 1;
}
```

Zadatak 2 – rješenje (gl. program)

```
int main()
{
    char niz[100];
    char podniz[30];
    gets(niz);
    gets(podniz);
    IzbaciPodniz(niz, podniz);
    printf("%s", niz);
    return 0;
}
```

Zadatak 3

- Napisati funkciju koja će vratiti pseudoslučajni broj u intervalu [-100, 100]. Funkcija treba imati prototip:
`int fun1(void);`
- Napisati funkciju koja će testirati ponašanje funkcije iz **a)** dijela zadatka. Funkciju `fun1` treba pozvati 1000 puta te za tako generirane brojeve izračunati aritmetičku sredinu i standardnu devijaciju. Izračunate vrijednosti vratiti u glavni program. Standardna devijacija računa se po formuli:

$$StDev = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

, gdje je \bar{x} aritmetička sredina

Zadatak 3 – rješenje (1/3)

```
int fun1(void) {  
    return rand()%201 - 100;  
}
```

**Nikako ne pisati `srand`
unutar ove funkcije!**

Još jedan način:

```
int fun1(void) {  
    return ((double)rand()) / (RAND_MAX + 1) * 201 - 100;  
}
```

Po formuli:

$$y = ((\text{double})\text{rand}()) / (\text{RAND_MAX} + 1) * (d - c + 1) + c$$

(c = -100, d = 100)

Zadatak 3 – rješenje (2/3)

```
void fun2(float *AS, float *SD) {
    int i, x;
    int polje[1000];
    float arsr = 0, stdev = 0;
    srand(time(NULL));

    for (i=0; i<1000; i++) {
        polje[i] = fun1();
        arsr += polje[i];
    }
    arsr /= 1000;
    ...
}
```

Zadatak 3 – rješenje (3/3)

```
for (i=0; i<1000; i++) {  
    stdev += pow(polje[i]-arsr, 2);  
}  
stdev = sqrt(stdev/1000);  
  
*AS = arsr;  
*SD = stdev;  
}
```

Zadatak 4

- Napisati funkciju vratiNiz koja na slučajan način generira riječ proizvoljne duljine. Riječ se sastoji od malih slova engleske abecede. Unutar riječi se ista slova smiju ponavljati (npr. aeertte i ghhz su ispravne riječi). Funkciji se zadaje tražena duljina riječi (cijeli broj) i niz u kojeg će upisati generiranu riječ.
- Također napisati glavni program koji će pomoću te funkcije generirati i na zaslone ispisati dvadeset riječi, čije su duljine od 1 do 20 znakova. **Napomena:** ispis riječi na zaslon obavlja glavni program, a ne funkcija.
- Primjer ispisa:
h
ot
...
kaskbfbaoeujghsoissk

Zadatak 4 – rješenje

```
void vratiNiz(char *niz, int d) {  
    while (d-->0) {  
        *niz = rand()%('z'-'a'+1) + 'a';  
        ++niz;  
    }  
    *niz = 0;  
}
```

Zadatak 4 – rješenje (gl. program)

```
int main() {
    char niz[20+1];
    int i;
    srand((unsigned int)time(NULL));
    for (i=1; i <= 20; ++i ) {
        vratiNiz(niz, i);
        puts(niz);
    }
    return 0;
}
```