

3. masovne instrukcije 2010./11.

Programiranje i programsko inženjerstvo

Održali:

Kristijan Franković

Alen Rakipović

05. siječanj 2011

Funkcije, jednodimenzionalna polja, generiranje pseudoslučajnih brojeva

Zadatak 1.

Napišite funkciju `vратиBroj` koja vraća jedan pseudo-slučajan cijeli broj preko imena funkcije. Funkcija prima donju i gornju granicu intervala iz kojeg će brojevi biti birani. Napišite i glavni program u kojem se generira 100 pseudo-slučajnih brojeva iz intervala `[0,50]` uz pomoć funkcije `vратиBroj`. Program na standardni izlaz ispisuje koji se brojevi iz prije navedenog intervala nisu pojavili niti jednom, rastućim redoslijedom. Koristite simboličke konstante.

Primjer:

Brojevi koji se nisu pojavili niti jednom su:

3 24 35 36 46

Rješenje:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define DG 0
#define GG 50
#define BROJ 100

int vratiBroj (int donja, int gornja)
{
    int broj;
    broj = rand ()%(gornja-donja+1)+donja;
    return broj;
}

int main ()
{
    int i, broj, brojevi[GG+1] = {0};
    srand ((unsigned)time(NULL));
    for (i=0; i<BROJ;i++)
    {
        broj = vratiBroj (DG, GG);
        brojevi[broj]++;
    }

    printf ("Brojevi koji se nisu pojavili niti jednom su:\n");

    for (i=0;i<=GG;i++)
        if (brojevi[i] == 0)
            printf ("%3d", i);
    printf ("\n");
    return 0;
}
```

Zadatak 2.

Napišite funkciju `genLozinku`, koja prima duljinu lozinke, prazan znakovni niz gdje će biti smještena lozinka koju će funkcija generirati, te niz znakova koji sadrži znakove koji mogu biti odabrani kao znakovi koji će se naći u generiranoj lozinci! Glavnim programom demonstrirati ponašanje funkcije! Također preko imena funkcije vratiti cijeli broj 1 ukoliko je lozinka uspješno generirana, a 0 ukoliko generiranje nije bilo uspješno. Generiranje lozinke nije uspješno ukoliko je niz znakova iz kojega se bira prazan znakovni niz. Ukoliko je potrebno, možete koristiti ugrađene funkcije. Dimenzije polja odaberite sami.

Primjer:

Unesite znakove: AbCD123

Unesite duljinu: 10

Generirana lozinka: D1311DCCb3

Rješenje:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

int genLozinku (int duljina, char *lozinka, const char *ulaz)
{
    int dg = 0, gg, i, indeks;
    if (strlen (ulaz) == 0)
        return 0;
    gg = strlen (ulaz) - 1;
    for (i=0; i<duljina; i++)
    {
        indeks = rand() % (gg-dg+1)+dg;
        *(lozinka+i) = *(ulaz+indeks);
    }
    *(lozinka+duljina) = '\0';
    return 1;
}

int main ()
{
    int n;
    char polje[50+1], lozinka [100+1];
    srand ((unsigned)time(NULL));
    printf ("Unesite znakove! ");
    gets(polje);
    printf ("Unesite duljinu lozinke, ne vecu od 100! ");
    scanf ("%d", &n);
    if (genLozinku (n, lozinka, polje) == 1)
    {
        printf ("Lozinka je uspjesno generirana i glasi: %s\n", lozinka);
    }
    else
        printf ("Lozinku nije bilo moguće generirati!\n");
    return 0;
}
```

Zadatak 3.

Napišite funkciju koja obavlja jednak posao kao i ugrađena funkcija `strstr` iz biblioteke `string.h`. Dakle funkcija prima dva polja znakova, `str1` i `str2`. Kao rezultat funkcija vraća pokazivač na prvo pojavljivanje niza znakova `str2` u nizu znakova `str1`. Ukoliko traženi niz ne postoji, vratiti `NULL`. Također napisati i glavni program kojim se demonstrira rad funkcije. Za učitavanje obavezno koristiti funkciju `scanf`, učitavanje prekinuti pritiskom na tipku enter (odnosno nailaskom znaka za novi red). Dimenzije polja odredite sami prema vlastitom izboru. Zabranjeno je koristiti ugrađene funkcije osim `strlen`.

Primjer:

Ulaz: abcdef fgh [ENTER] def

Izlaz: def fgh

Rješenje:

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

#define MAX 100

char* mojStr (char *niz1, char *niz2)
{
    int i = 0, jednaki = 1, j;

    while (*(niz1+i) != '\0')
    {
        j = 0;

        if (strlen (niz1+i) < strlen(niz2))
            break;

        jednaki = 1;
        while (*(niz2+j) != '\0')
        {
            if (*(niz1+i+j) != *(niz2+j))
            {
                jednaki = 0;
                break;
            }
            j++;
        }

        if (jednaki == 1)
            return niz1+i;

        i++;
    }
    return NULL;
}

int main ()
{
    char prvoPolje [MAX];
    char drugoPolje [MAX];
    char smece;

    scanf ("%s", prvoPolje);
    scanf ("%c", &smece);
    printf ("%s", mojStr (prvoPolje, drugoPolje));
    return 0;
}
```

Zadatak 4.

Napišite funkciju koja prima dva znakovna niza. Prvi niz je komprimirano potrebno zapisati u drugi (koji je prazan) na slijedeći način: Prvo je potrebno zapisati znak koji se komprimira pa potom broj pojavljivanja znaka, osim ukoliko je broj pojavljivanja znaka 1, tada se zapisuje samo znak! Primjerice: A A A B C B B se prevodi u A 3 B C B 2. Drugi niz je potrebno zaključiti null znakom! Broj slova koja se mogu uzastopno pojaviti nije ograničen osim ukupnom duljinom niza! Također može se pretpostaviti da će se u ulaznom nizu nalaziti samo slova, bilo velika bilo mala. Također napišite glavni program koji demonstrira rad funkcije. Pritom je maksimalni broj znakova 200, (ponuditi odabir korisniku). Znakove generirati pomoću generatora pseudoslučajnih brojeva, tako da se bira između tri znaka. A, B, C. Pritom je vjerojatnost pojavljivanja slova A 0.6, slova B 0.2, slova C 0.2. Prema potrebi, ukoliko se za to ukaže potreba možete napisati i dodatne funkcije!

Primjer:

Ulaz: 5

Izlaz: AAcCC A2cC2

Rješenje:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define MAX 200

void vratiZnamenke (int broj, char *polje)
{
    int koef, pom, i = 0;
    pom = broj;
    koef = 1;

    while (pom != 0)
    {
        koef *= 10;
        pom /= 10;
    }

    pom = broj;
    while (pom != 0)
    {
        *(polje+i) = pom % koef / (koef / 10) + '0';
        i++;
        koef = koef / 10;
        pom %= koef;
    }
    *(polje+i) = '\\0';
    return;
}
```

```

void komprimiraj (char *niz1, char *niz2){
    int i=0, j=0, k, ponavljanja = 0;
    char znak = *(niz1);
    char pom[4];

    while (1)
    {
        if (*(niz1+i) == znak)
            ponavljanja++;
        else
        {
            *(niz2+j) = znak;
            j++;
            if (ponavljanja != 1)
            {
                vratiZnamenke (ponavljanja, pom);
                k = 0;
                while (*(pom+k) != '\0')
                {
                    *(niz2+j) = *(pom+k);
                    k++;
                    j++;
                }
            }
            znak = *(niz1+i);
            ponavljanja = 1;
        }

        if (*(niz1+i) == '\0')
            break;

        i++;
    }
    *(niz2+j) = '\0';
    return;
}

int main ()
{
    int i, odabir;
    char polje [MAX+1], komprimirano[MAX+1];
    int n;
    srand((unsigned)time(NULL));
    scanf ("%d", &n);

    for (i=0;i<n;i++)
    {
        odabir = rand()%(1-0+1)+0;
        if (odabir == 0)
        {
            polje[i] = rand()%('C'-'A' + 1)+'A';
        }
        else
        {
            polje[i] = rand()%('c'-'a' + 1)+'a';
        }
    }
    polje[n] = '\0';
    puts (polje);

    komprimiraj (polje, komprimirano);
    puts(komprimirano);
    return 0;
}

```

Zadatak 5.

Napišite funkciju koja prima tri znakovna niza. Funkcija treba u prvom znakovnom nizu pronaći sva pojavljivanja drugog znakovnog niza te izvršiti zamjenu sa trećim znakovnim nizom i to do duljine drugog znakovnog niza (prevedno: nije potrebno proširivati polje). Također može se pretpostaviti da će niz2 i niz3 biti najmanje jednake duljine (ipak, niz3 može biti veći od niz2). Pritom se do tada zamijenjeni dio prvog niza smatra obrađenim, odnosno nije potrebno ponovo vršiti zamjenu ukoliko treći niz sadrži drugi niz kao svoj podniz. Demonstrirati rad programa glavnim programom! Obavezno je koristiti ugrađene funkcije! Dimenzije nizova odredite proizvoljno.

Primjer:

Ulaz: strstrestrlolstre [ENTER] str [ENTER] lololol

Izlaz: lollolelollole

Rješenje:

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

#define NIZ 100
#define PODN 10

void suludaZamjena (char *niz1, char *niz2, char *niz3)
{
    char *p = niz1;
    int i, duljina, j;
    duljina = strlen (niz2);

    while (1)
    {
        p = strstr (p,niz2);
        if (p == NULL)
            break;
        else
        {
            for (i=0;i<duljina;i++)
            {
                *(p+i) = *(niz3 + i);
            }
            p = p + duljina;
        }
    }
    return;
}

int main ()
{
    char prvi[NIZ+1];
    char drugi[PODN+1];
    char treci[PODN+1];
    gets (prvi);
    gets (drugi);
    gets (treci);
    suludaZamjena (prvi, drugi, treci);
    puts (prvi);
    return 0;
}
```

Funkcije i dvodimenzionalna polja, generiranje slučajnih brojeva

Zadatak 6.

Napišite glavni program u kojem se generira znakovna matrica proizvoljnih dimenzija. (Sami odredite dimenzije matrice). Potrebno je generirati samo velika slova. Potom ispisati generiranu matricu. Napišite funkciju koja će vratiti indeks retka i stupca u kojem započinje niz znakova predanih kao argument funkciji, ali samo ako je niz pronađen u cijelosti u matrici. Niz se traži u stupcima! Ukoliko traženi niz nije bilo moguće pronaći, vratiti -1 kao indekse retka i stupca. Ukoliko postoji više takvih nizova vratiti onaj sa najmanjim indeksom.

Primjer:

Ulaz: 3 3

Izlaz:

```
W S W
D D M
V V J
```

Ulaz: DV

Izlaz: 1 1

Rješenje:

```
int main ()
{
    int m,n,i,j, redak, stupac;
    char matrica[MAX][MAX];
    char niz [MAX];
    char smece;

    scanf ("%d %d", &m, &n);

    srand (time(NULL));

    for (i = 0; i<m;i++)
        for (j=0;j<n;j++)
            matrica[i][j] = rand()%('Z'-'A'+1)+'A';

    for (i = 0; i<m;i++)
    {
        for (j=0;j<n;j++)
            printf ("%3c", matrica[i][j]);
        printf ("\n");
    }

    scanf ("%c", &smece);

    gets (niz);

    matStr (&redak, &stupac, &matrica[0][0], niz, m,n,MAX);

    printf ("%d %d\n", redak, stupac);

    return 0;
}
```



```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

#define MAX 10

void matStr (int* brred, int* brstup, char *matrica, char* niz, int m, int n,
int maxs)
{
    int i, j, k, jednaki = 0;
    *brred = -1;
    *brstup = -1;

    if (strlen(niz) > m)
        return;

    for (i=0;i<n;i++)
    {
        for (j=0;j<m;j++)
        {
            if (strlen(niz) > m-j)
                break;

            k = 0;
            jednaki = 1;
            while (*(niz+k) != '\0')
            {
                if (*(niz+k) != (*(matrica + (j+k)*maxs+i)))
                {
                    jednaki = 0;
                    break;
                }
                k++;
            }
            if (jednaki == 1)
            {
                *brred = j;
                *brstup = i;
                break;
            }
        }
        if (jednaki == 1)
            break;
    }
}

```

Zadatak 7.

Napišite funkciju koja kao rezultat prilikom poziva vraća jedan pseudoslučajan cijeli broj iz intervala [DONJA, GORNJA]. Pomoću te funkcije popunite u glavnom programu matricu proizvoljnih dimenzija (korisniku ponuditi da uči stvarne dimenzije matrice, može se pretpostaviti da neće biti veća od 10 X 10, ne mora biti kvadratna, ne treba provjeravati ispravnost podataka, koristiti simboličke konstante). Brojevi neka budu iz intervala [-20,20]. Matricu potom ispišite na ekran. Nakon toga napišite funkciju koja će matricu presložiti na način da se redak sa najmanjom sumom nalazi na najmanjem indeksu, a redak sa najvećom sumom na najvećem indeksu. Posao si olakšajte pisanjem funkcije koja računa sumu. Nakon svih ovih transformacija matricu ispisati na ekran u glavnom programu.

Primjer:

Ulaz: 3 3

Izlaz:

```
-15  -18  11
 10   -1  20
 -9   19   9

 10   -1  20
 -9   19   9
-15  -18  11
```

Rješenje:

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

#define MAX 10

int vratiBroj (int donja, int gornja)
{
    int broj = rand()%(gornja-donja+1)+donja;
    return broj;
}

int suma (int* redak, int n)
{
    int i, suma=0;
    for (i=0;i<n;i++)
        suma += *(redak+i);
    return suma;
}
```

```

int main ()
{
    int matrica[10][10];
    int m, n, i, j, max, indeks, sumaRetka, pom;

    srand((unsigned)time(NULL));

    scanf ("%d %d", &m, &n);

    for (i=0;i<m;i++)
        for (j=0;j<n;j++)
            matrica[i][j] = vratiBroj (-20,20);

    for (i=0;i<m;i++)
    {
        for (j=0;j<n;j++)
            printf ("%4d",matrica[i][j]);
        printf ("\n");
    }

    for (i=0;i<m;i++)
    {
        max = suma (matrica[i],n);
        indeks = i;
        for (j=i;j<m;j++)
        {
            sumaRetka = suma (matrica[j],n);
            if (sumaRetka > max)
            {
                max = sumaRetka;
                indeks = j;
            }
        }

        for (j=0;j<n;j++)
        {
            pom = matrica[i][j];
            matrica[i][j] = matrica[indeks][j];
            matrica[indeks][j] = pom;
        }
    }

    printf ("\n");

    for (i=0;i<m;i++)
    {
        for (j=0;j<n;j++)
            printf ("%4d",matrica[i][j]);
        printf ("\n");
    }

    return 0;
}

```

Zadatak 8.

Napišite funkciju `matcat` koja prima matricu te polje znakova. U matrici se nalazi niz rečenica (svaki redak matrice predstavlja jednu rečenicu zaključenu null-znakom `'\0'`). Potrebno je svaki redak matrice nalijepiti jedan iza drugoga u polje znakova (koje je na početku prazno!) počevši od prvog retka matrice. Funkcija preko imena vraća koliko je ukupno rečenica bilo kraće od 5 znakova. Obavezno koristite ugrađene funkcije iz biblioteke `string.h`. Dodatno, razmislite kako bi se zadatak riješio bez ugrađenih funkcija. Također napišite i glavni program, može se pretpostaviti da matrica neće biti veća od 10 x 30. Učitavanje matrice raditi dok god se ne učitava niz „kraj“ ili dok se ne učitava deset redaka. Također učitati maksimalno 29 znakova (1 znak ćemo sačuvati za NULL karakter). Nakon poziva funkcije `matcat`, ispisati novopopunjeno polje i broj učitanih rečenica manjih od 5!

Primjer:

Ulaz: asd [ENTER] asdef

Izlaz:

Manjih od 5 je bilo: 1
asd
asdef

Rješenje:

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

#define MAXR 10
#define MAXS 30

int matcat (char *matrica, int m, char *polje, int maxs)
{
    int i, brojac = 0;
    for (i=0; i<m; i++)
    {
        if (strlen (matrica+i*maxs) < 5)
            brojac++;
        strcat (polje, (matrica+i*maxs));
    }

    return brojac;
}

int main ()
{
    char matrica[MAXR][MAXS], smece, polje[MAXR * MAXS];
    int i=0, manjih, ucitanoRedaka;

    polje[0] = 0;

    while (i < MAXR)
    {
        scanf ("%29[^\n]%c", matrica[i], &smece);
        if (strcmp (matrica[i], "kraj") == 0)
            break;
        i++;
    }

    ucitanoRedaka = i;

    manjih = matcat (&matrica[0][0], ucitanoRedaka, polje, MAXS);

    printf ("Manjih od pet je bilo %d\n", manjih);
    puts (polje);

    return 0;
}
```

Datoteke

Zadatak 9.

Za potrebe izgradnje informatičkog sustava za Konzum Ivica Todorčić se obratio FER-u i FER-ovcima. Do sada su djelatnici Konzuma ručno baratali svim podacima, a sve podatke držali na papirima. Pomognite Konzumu da postigne učinkovitije poslovanje, Todorčiću da se još više besramno bogati, a da blagajnice imaju još više posla tako što:

- a) Napišite program koji će omogućiti djelatnicima Konzuma unos artikala u direktnu binarnu datoteku. Potrebno je zapisati šifru artikla (cijeli broj), naziv (najviše 20 znakova), količinu (cijeli broj) te cijenu (realni broj). Sve šifre artikala su 4 znamenkaste i počinju brojkom 9. Dakle mogu biti iz intervala [9 000, 9 999] Za pristup zapisu koristite šifru artikla. Učitajte 5 artikala.

```
#include <stdio.h>
#define BROJ 5

typedef struct
{
    int sifra;
    char naziv[20+1];
    int kolicina;
    float cijena;
} artikl;

int main ()
{
    char smece;
    int i;
    FILE *tokPodataka;
    artikl zapis;

    tokPodataka = fopen ("artikli.bin", "wb");

    if (tokPodataka == NULL)
    {
        printf ("Greška - nije moguće otvoriti datoteku!");
        return -1;
    }

    for (i=0;i<BROJ;i++)
    {
        printf ("Sifra %d. artikla ", i+1);
        scanf ("%d%c", &zapis.sifra, &smece);
        printf ("Naziv %d. artikla ", i+1);
        gets (zapis.naziv);
        printf ("Kolicina i cijena %d. artikla ", i+1);
        scanf ("%d %f", &zapis.kolicina, &zapis.cijena);

        if (fseek(tokPodataka, (long)(zapis.sifra-9000)*sizeof(zapis), SEEK_SET) != 0)
        {
            printf ("Nije uspjelo pozicioniranje u datoteku\n");
            return -1;
        }

        if (fwrite(&zapis, sizeof(zapis), 1, tokPodataka) != 1)
        {
            printf ("Nije uspjelo zapisivanje u datoteku!\n");
            return -1;
        }
    }
    fclose (tokPodataka);
}
```

b) Ispišite sve artikle na ekran iz takve direktne binarne datoteke!

```
#include <stdio.h>

typedef struct
{
    int sifra;
    char naziv[20+1];
    int kolicina;
    float cijena;
} artikl;

int main ()
{
    char smece;
    int i, rbZapis;
    FILE *tokPodataka;
    artikl zapis;

    tokPodataka = fopen ("artikli.bin", "rb");

    if (tokPodataka == NULL)
    {
        printf ("Greška - nije moguće otvoriti datoteku!");
        return -1;
    }

    rbZapis = 9000;
    while (fread(&zapis,sizeof(zapis),1,tokPodataka) == 1)
    {
        if (rbZapis == zapis.sifra)
            printf ("%d %s %d %f\n", zapis.sifra,zapis.naziv,zapis.kolicina,zapis.cijena);
        rbZapis++;
    }
    fclose (tokPodataka);
}
```