Programiranje I

Beleške sa vežbi

Smer *Informatika*

Matematički fakultet, Beograd

Sana Stojanović

December 28, 2007

Sadržaj

1 Funkcije za rad sa niskama

1. Napisati funkciju koja za dva data stringa proverava da li se drugi string nalazi u prvom i vraća poziciju na kojoj se nalazi. U slučaju da se string ne nalazi funkcija vraća -1.

```
/* Proverava da li string str sadrzi string sub.
   Vraca poziciju na kojoj sub pocinje, odnosno -1 ukoliko ga nema
*/
int string_string(char str[], char sub[])
{
        int i, j;
        /* Proveravamo da li sub pocinje na svakoj poziciji i */
        for (i = 0; str[i]; i++)
                /* Poredimo sub sa str pocevsi od poziciji i
                   sve dok ne naidjemo na razliku */
                for (j = 0; str[i+j] == sub[j]; j++)
                        /* Nismo naisli na razliku a ispitali smo
                           sve karaktere niske sub */
                        if (sub[j+1]=='\0')
                                return i;
        /* Nije nadjeno */
        return -1;
}

main()
{
        printf("%d\n",string_string("racunari", "rac"));
        printf("%d\n",string_string("racunari", "ari"));
        printf("%d\n",string_string("racunari", "cun"));
        printf("%d\n",string_string("racunari", "cna"));
}
```

2. Napisati program koji za tekst *ulaz.txt* ispisuje sve linije u kojima se pojavljuje reč *for* ili reč *while*. Program pozivati sa preusmeravanjem ulaza (npr. `./a.out < ulaz.txt`

```
#include <stdio.h>

/* Maksimalna duzina linije */
#define MAX_LINE 100

/* Funkcija ucitava liniju sa standardnog ulaza i smesta je u datu nisku
   karaktera s. lim oznacava maksimalnu duzinu niske i funkcija vodi
   racuna da se taj limit ne premasi.
   Oznaka za kraj reda se smatra sastavnim delom linije.
```

```
   Funkcija vraca broj procitanih karaktera.
*/
int getline(char s[], int lim)
{
        int c;

        /* Cita se karakter po karakter i smesta u nisku sve dok se ne
           dodje do EOF, '\n' ili dok se ne premasi duzina niza
           (ostavljajuci mesto za '\0')
           Napomena: zbog lenjog izracunavanja, ukoliko je duzina niza
           premasena ne cita se sledeci karakter.
        */
        int i = 0;
        while (i < lim - 1 && (c=getchar()) != EOF && c != '\n')
                s[i++] = c;

        /* Posto je '\n' sastavni deo linije i on se smesta u niz */
        if (c == '\n')
                s[i++] = c;

        /* Terminira se niska na kraju */
        s[i] = '\0';

        return i;
}

int string_string(char str[], char sub[])
{
        int i, j;
        /* Proveravamo da li sub pocinje na svakoj poziciji i */
        for (i = 0; str[i]; i++)
                /* Poredimo sub sa str pocevsi od poziciji i sve dok ne
                   naidjemo na razliku */
                for (j = 0; str[i+j] == sub[j]; j++)
                        /* Nismo naisli na razliku a ispitali smo sve
                           karaktere niske sub */
                        if (sub[j+1]=='\0')
                                return i;
        /* Nije nadjeno */
        return -1;
}

main()
{
    /* Ispisujemo sve linije koje sadrze petlju */
```

```
    /* Niska karaktera koja ce da sadrzi tekucu liniju.
       Alociramo prostor od MAX_LINE karaktera */
    char line[MAX_LINE];

    /* Citamo liniju - getline vraca 0 kada dodje do kraja ulaza */
    while(getline(line, MAX_LINE)>0)
       /* Ispisujemo one linije koje sadrze rec for ili rec while */
       if (string_string(line, "for") != -1
           || string_string(line, "while") != -1)
            printf("%s",line);
}
```

3. Napisati funkciju koja učitava jednu reč sa standardnog ulaza. Za proveru da li smo došli do kraja reči koristimo funkciju `isspace` iz zaglavlja `ctype.h`

```
/* Demonstrira prenos nizova u funkciju - preneti niz se moze menjati */

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

/* Funkcija ucitava rec sa standardnog ulaza i smesta je u niz karaktera s.
   Ovo uspeva zbog toga sto se po vrednosti prenosi adresa pocetka niza,
   a ne ceo niz */
void get_word(char s[])
{
        int c, i = 0;
        while (!isspace(c=getchar()))
                s[i++] = c;
        s[i] = '\0';
}

main()
{
        /* Obavezno je alocirati memoriju za niz karaktera */
        char s[100];

        get_word(s);
        printf("%s\n", s);
}
```

2 Konverzije

1. Napisati program koji konvertuje nisku cifara u njihovu odgovarajuću brojevnu vrednost (na primer, niska "123" predstavlja ceo broj 123).

```
/* atoi - konverzija niske cifara u brojnu vrednost */
/* Program omogucuje citanje celog broja bez koriscenja funkcije scanf("%d")
   Napomena : funkcija atoi vec postoji u zaglavlju <stdio.h> i ovde
   je implementirana samo za vezbu
*/

#include <stdio.h>
#include <ctype.h> /* Zbog funkcije isspace */

/* Maksimalna duzina reci */
#define MAX_DIGITS 10


/* AsciiTOInteger
   Funkcija izracunava vrednost celog broja koji je zapisan u datom
   nizu karaktera. Za izracunavanje se koristi Hornerova shema. */

int atoi(char s[])
{
        int sum = 0;

        /* Obradjuju se karakteri sve dok su cifre */
        int i;
        for (i = 0; isdigit(s[i]); i++)
                sum = 10*sum + (s[i]-'0');

        return sum;
}


main()
{
        char s[MAX_DIGITS];
        int num;

        /* Ucitavamo broj u vidu niske karaktera */
        printf("Unesi ceo broj : ");

        /* Obratiti paznju da uz s ne stoji & */
        scanf("%s", s);

        /* Izracunava se njegova vrednost */
        num  = atoi(s);

        /* Ispisuje se vrednost */
        printf("Uneo si broj : %d\n", num);
```

```
        printf("Broj za tri veci je : %d\n",  num+3);

}
```

2. Napisati program koji datu nisku karaktera koja predstavlja broj zapisan u proizvoljnom sistemu, konvertuje u dekadni sistem.

```
/* btoi - konverzija iz datog brojnog sistema u dekadni */
#include <stdio.h>
#include <ctype.h>

/* Pomocna funkcija koja izracunava vrednost koju predstavlja karakter u
   datoj osnovi. Funkcija vraca -1 ukoliko cifra nije validna.

   Npr.
   cifra 'B' u osnovi 16 ima vrednost 11
   cifra '8' nije validna u osnovi 6

*/

int digit_value(char c, int base)
{
        /* Proveravamo obicne cifre */
        if (isdigit(c) && c < '0'+base)
                return c-'0';

        /* Proveravamo slovne cifre za mala slova */
        if ('a'<=c && c < 'a'+base-10)
                return c-'a'+10;

        /* Proveravamo slovne cifre za velika slova */
        if ('A'<=c && c < 'A'+base-10)
                return c-'A'+10;

        return -1;
}

/* Funkcija izracunava vrednost celog broja koji je zapisan u datom
   nizu karaktera u datoj osnovi. Za izracunavanje se koristi Hornerova shema.
*/
int btoi(char s[], int base)
{
        int sum = 0;

        /* Obradjuju se karakteri sve dok su cifre */
        int i, vr;
```

```
        for (i = 0; (vr = digit_value(s[i], base)) != -1; i++)
                sum = base*sum + vr;

        return sum;

}

main()
{
        char bin[] = "11110000";
        char hex[] = "FF";

        printf("Dekadna vrednost binarnog broja %s je %d\n", bin, btoi(bin, 2));
        printf("Dekadna vrednost heksadekadnog broja %s je %d\n", hex,
            btoi(hex, 16));

}
```

3 Lenjo izračunavanje

1. ```
   /* Ilustracija lenjog izracunavanja logickih operatora */

   /* Prilikom izracunavanja izraza - A && B, ukoliko je A netacno, izraz B
      se ne izracunava.
      Prilikom izracunavanja izraza - A || B, ukoliko je A tacno, izraz B
      se ne izracunava.
   */

   #include <stdio.h>

   int b = 0;

   /* Funkcija ispisuje da je pozvana i uvecava promenjivu b.
      Funkcija uvek vraca vrednost 1 (tacno)
   */
   int izracunaj()
   {
       printf("Pozvano izracunaj()\n");
       b++;
       return 1;
   }

   main()
   {
       /* Funkcija izracunaj() ce se pozivati samo za parne vrednosti a */
   ```
   ```

```
    int a;
    for (a = 0; a < 10; a++)
        if (a%2 == 0 && izracunaj())
            printf("Uslov ispunjen : a = %d, b = %d\n", a, b);
        else
            printf("Uslov nije ispunjen : a = %d, b = %d\n", a, b);

    printf("----------------------------\n");

    /* Funkcija izracunaj() ce se pozivati samo za neparne vrednosti a */
    b = 0;
    for (a = 0; a < 10; a++)
        if (a%2 == 0 || izracunaj())
            printf("Uslov ispunjen : a = %d, b = %d\n", a, b);
        else
            printf("Uslov nije ispunjen : a = %d, b = %d\n", a, b);
}
```

```
Izlaz:
Pozvano izracunaj()
Uslov ispunjen : a = 0, b = 1
Uslov nije ispunjen : a = 1, b = 1
Pozvano izracunaj()
Uslov ispunjen : a = 2, b = 2
Uslov nije ispunjen : a = 3, b = 2
Pozvano izracunaj()
Uslov ispunjen : a = 4, b = 3
Uslov nije ispunjen : a = 5, b = 3
Pozvano izracunaj()
Uslov ispunjen : a = 6, b = 4
Uslov nije ispunjen : a = 7, b = 4
Pozvano izracunaj()
Uslov ispunjen : a = 8, b = 5
Uslov nije ispunjen : a = 9, b = 5
----------------------------
Uslov ispunjen : a = 0, b = 0
Pozvano izracunaj()
Uslov ispunjen : a = 1, b = 1
Uslov ispunjen : a = 2, b = 1
Pozvano izracunaj()
Uslov ispunjen : a = 3, b = 2
Uslov ispunjen : a = 4, b = 2
Pozvano izracunaj()
Uslov ispunjen : a = 5, b = 3
Uslov ispunjen : a = 6, b = 3
Pozvano izracunaj()
```

```
Uslov ispunjen : a = 7, b = 4
Uslov ispunjen : a = 8, b = 4
Pozvano izracunaj()
Uslov ispunjen : a = 9, b = 5
```

4 Primena prostih brojeva

1. Napisati program koji prikazuje sve proste brojeve u datom intervalu kojima je zbir cifara složen broj. Interval se zadaje učitavanjem gornje i donje granice (dva prirodna broja). Brojeve prikazati u opadajucem poretku.

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

int prost (int n);  /*testira da li je broj n prost broj */
int zbirCifara (int n);   /*vraca zbir cifara broja n */

main()
{
    int donja,gornja;  /*granice intervala */
    int i;  /*brojac u petlji */
    int pom;   /*posrednik u eventualnoj zameni */

  /*ucitavanja granice intervala */
    scanf("%d%d", &donja, &gornja);
     if (donja > gornja)   /*obezbedjivanje relacije: donja <=gornja */
    {
        pom=donja;
        donja=gornja;
         gornja=pom;
     }
     for(i=gornja;i>=donja; i--)
     if (prost (i) && !prost(zbirCifara(i) )  ) printf("%d\n",i);
}

int prost(int n)
/* Ispituje se da li je broj n prost tako to se proverava da li ima
   delioce medju brojevima od 2 do n/2.  Pri implementaciji se koristi
   tvrdjenje da je broj prost ako je jednak 2, ili ako je neparan i ako
   nema delitelja medju neparnim brojevima od 3 do n/2 */
{
    int prost;  /*indikator slozenosti broja n */
    int i;    /*potencijalni delitelj broja n */
    if (n==1) return 0;
     /*parni brojevi razliciti od  od dva nisu prosti brojevi */
```

```
     prost= (n%2!=0) || (n==2);

     /*najmanji potencijalni kandidat za delitelje medju
      neparnim brojevima razlicitim od jedan */
     i=3;
    while ( (prost) && (i<=n/2) )
    {
      prost=n%i != 0;
      i=i+2; /*proveravamo kandidate za delitelje samo medju neparnim brojevma */
    }
 return prost;
}
int zbirCifara (int n)
{
   int  Suma=0;
   while (n>0)
   {
     Suma+= n%10;  /* dodavanje tekuce cifre, krecemo od cifre jedinica  */
     n=n/10;      /* da bi smo mogli da dobijemo sledecu cifru */
   }
 return Suma;
}

Ulaz:
1 20
Izlaz:
19
17
13
```

2. Napisati program koji koristeći funkciju `prost()` štampa sve brojeve blizance do datog prirodnog broja n. Dva prirodna broja su blizanci ako su prosti i razlikuju se za 2 (npr. 3 i 5, 5 i 7, 11 i 13, itd.)*/

```
#include <stdio.h>

main()
{
int n, i;
scanf("%d", &n);

for (i = 3; i <= n-2; i+=2)
    if (prost(i) && prost(i+2))
        printf("%d %d ", i, i+2);
}
```

```
II nacin:
sa smanjenim brojem poziva funkcije prost()

main()
{
int n, i, prethodni, tekuci;
scanf("%d", &n);

prethodni = 0;
for(i = 3; i <=n; i+=2)
{
    //Obratiti paznju na to da zbog lenjog izracunavanja sledeci
    //uslov u if naredbi mora biti zapisan bas ovako a ne obrnutim redosledom
    if (tekuci = prost(i) && prethodni)
        printf("%d %d", i-2, i);
    prethodni = tekuci;
}
```

5 Treći i četvrti domaći zadatak

Fajlove *domaci3.c*, *domaci4.c* i *domaci.tex* poslati u roku od 2 nedelje od objavljivanja zadataka.

1. Napisati program koji za uneto n sa standardnog ispisuje zvezdice na sledeći način. Na primer, za uneto $n = 5$ program treba da ispiše:

```
* * * * *
 * * * *
  * * *
   * *
    *
```

2. Broj je *Niveov* ako je deljiv sumom svojih cifara. Napisati program koji za uneto n sa standardnog ulaza ispisuje prvih n Niveovih brojeva.