107. NCP koji ucitava iz datoteke ulaz.txt nisku sa ne vise od 20 karaktera i ispisuje na standardni izlaz da li je uneta niska palindrom.

```
#include <stdio.h>
#include <string.h>
int palindrom (char niska[])
/*vraca 0 ako niska nije palindrom, vraca 1 ako niska jeste palindrom */
{    int i,kraj; /* brojaci u ciklusu */
     kraj=strlen(niska)-1;
    for(i=0; i<=kraj; i++, kraj--)
        if(niska[i] != niska[kraj] ) return 0;
    return 1;
}
 main ()
{ char s[21];
  FILE *f;
  f=fopen("ulaz.txt","r");
  fscanf(f,"%s",s);
  if (palindrom(s)) printf("\nJeste"); else printf("\nNije");
  printf(" palindrom\n");
  fclose(f);
}
```

108. U svakoj ulaznoj liniji datoteke ulaz.txt nalazi se veliki prirodan broj bez vodecih nula (do 100 cifara). Ispisati u datoteku izlaz.txt najveci broj.

```
#include <stdio.h>
#include <string.h>
#define MAX 101
int veci(char *p, char *d)
{ if(strlen(p)==strlen(d))return strcmp(p,d)>0;
            else  return strlen(p)>strlen(d);}
main()
{
   FILE *ul,*izl;
   char s[MAX],max[MAX];
   max[0]=0;
```

```
   ul=fopen("ulaz.txt", "r");
   izl=fopen("izlaz.txt", "w");
   while(fgets(s,100,ul)!=NULL)
    if (veci(s,max)) strcpy(max,s);
   fprintf(izl,"%s",max);
   fclose(ul);
   fclose(izl);
return 0;
}
```

109. NCP koji iz datoteke ulaz.txt ucitava datume u obliku dd.mm.gggg. (do nailaska na oblezje kraja datoteke) i ispisuje na standardni izlaz najveci datum. U slucaju da postoji vise takvih ispisati tacno jedan od njih.

Napomena pre resavanja: U tekstu zadatka nisu date nikakve pretpostavke o ukupnom broju datuma u datoteci !!! Dakle, Vi kao programer ne znate da li ih ima 30 ili 300. Vodite racuna da ne upotrebite strukturu podataka koja cuva datume i zahteva od Vas da u programu navedete pretpostavljeni broj clanova strukture.

```
#include <stdio.h>
#include <stdlib.h>
int veci(int d1, int m1, int g1, int d2, int m2, int g2);
/*vraca 1, ako je prvi datum d1.m1.g1 veci od drugog d2.m2.g2.
   vraca 2, ako je drugi datum veci
  vraca 0, ako su datumi jednaki */
void poruka(int d, int m, int g);  /*stampa rezultat u formi d.m.g. */
main()
{
    FILE *f;
    int d1,m1,g1;  /* 1.datim d1.m1.g1. */
    int maxd=0,maxm=0,maxg=0;  /* najveci datum u obliku maxd.maxm.maxg. Polazna pretpostavka:
0.0.0. */
    f=fopen("ulaz.txt", "r");
    if (f==NULL) {fprintf(stderr, "Greska u citanju datoteke\n"); exit(1);}
```

```
 while (!feof(f))
   {  fscanf(f, "%d.%d.%d.", &d1, &m1, &g1);
      if (veci(d1,m1,g1,maxd,maxm,maxg)==1)
      {  /* postavi nove vrednosti za najveci datum, jer je d1.m1.g1. novi najveci datum */
          maxd=d1; maxm=m1; maxg=g1;
       }
  }
 poruka(maxd,maxm,maxg);
 fclose(f);
}
void poruka(int d, int m, int g)
{     printf("\n%d.%d.%d.\n",d,m,g);}
int veci(int d1, int m1, int g1, int d2, int m2, int g2)
{    int status;
 /* status = 0,1,2 u zavisnosti da li prvi datum je jednak drugom, veci od drugog, manji od drugog */
    if (g1 > g2) status=1; /*jer prvi datum je tada veci od drugog*/
   else if (g1 < g2) status=2;
   else /* ako su godine u oba datuma jednaka, porede se meseci */
      if (m1 > m2) status=1;
      else if (m1 < m2) status=2;
      else  /*porede se dani ako su meseci u oba datuma jednaki */
          if (d1 > d2) status =1;
          else if (d1 < d2) status =2;
          else status=0;
 return status;
}
```

110. NCP za određivanje broja velikih slova, malih slova i cifara u tekstu koji se učitava iz datotele ulaz.txt. Tekst se završava znakom za kraj datoteke.

```
#include <stdio.h>
#include <ctype.h> /* treba za funkcije vezane za ispitivanje slova i cifara*/
main()
{
```

```
   FILE *f;
        int znak, vel_sl=0, mal_sl=0,cifra=0;

        f=fopen("ulaz.txt","r");

        while ((znak=fgetc(f))!=EOF)
        {
                vel_sl += isupper(znak) != 0; /*  ctype.h */
                mal_sl += islower(znak) != 0; /*  ctype.h */
                cifra  += isdigit(znak) != 0; /*  ctype.h */
/* Ove funkcije su veoma jednostavne. isupper pita (znak>='A') && (znak<='Z')
  islower pita (znak>='a') && (znak<='a').isdigit pita (znak>='0')&&(znak<='9')
  Sve is_ funkcije vracaju 0 za logicku neistinu, a razlicito od 0 (ne obavezno 1) za
  logicku istinu*/
        }
        printf("Velikih slova ima %d\n",vel_sl);
        printf("Malih slova ima %d\n",mal_sl);
        printf("Cifara ima %d\n",cifra);
        fclose(f);
}
```

ZADATAK ZA OCENU 5 (samostalni rad, bez pomoci profesora)

1. Neka se pod sažetim rečnikom kompjuterskih termina podrazumeva datoteka sledećeg oblika:
a) prvi red sadrži termin, a naredni redovi sadrže objašnjenje tog termina
b) nakon objašnjenja sledi prazan red, pa sledeći termin
c) termini su poredjani u poretku engleske abecede
d) termini nisu duži od 40 karaktera
Napisati program koji spaja dva sažeta rečnika kompjuterskih termina i to *recnik1.dat* i *recnik2.dat* u datoteku *recnik.dat* oblika navedenog pod a) b) c) d). Ako postoji duplikat termina, ponoviti samo objašnjenje bez termina i objašnjenja istog termina
razdvojiti jednim redom koji sadrži znake '<', '=', '>'.

ZADACI ZA VEZBU (samostalni rad, obratiti se za pomoc profesoru ako nesto nije jasno tokom resavanja)

1. NCP koji će na standardni izlaz ispisati koliko ima samoglasnika (A,E,I,O,U,a, e, i, o, u) u ulaznoj datoteci ulaz.txt.

2. NCP koji ispisuje na standardni izlaz da li su jednake prva rec datoteke dat1.txt i prva rec druge datoteke dat2.txt. Smatrati da reci nemaju vise od 20 karaktera. U slucaju greske u otvaranju datoteka, prijaviti odgovarajuci komentar na stderr.

TEST primer

dat1. txt

Programiranje je moj omiljeni predmet.

dat2.txt

Programiranje nije moj omiljeni predmet.

IZLAZ

Prva rec 1. datoteke je Programiranje, prva rec 2. datoteke je Programiranje i te dve reci su jednake.

3. NCP koji iz ulazne datoteke DAT1.txt kopira u izlaznu datoteku DAT2.txt svaki drugi karakter polazeci od drugog procitanog karaktera. U slucaju greske u otvaranju i zatvaranju datoteka, prijaviti odgovarajuci komentar na stderr.

TEST primer

DAT1. txt

Programiranje.

IZLAZ

rgaiaj.

4. NCP koji ucitava prvu rec datoteke dat1.txt i prvu rec druge datoteke dat2.txt i proverava da li se prva rec pojavljuje u drugoj reci. Na kraju ispisati odgovarajucu poruku. (na primer, rec **aka** se pojavljuje u reci **baka**, rec **aka** se ne pojavljuje u reci **bajka**, rec **majka** se ne pojavljuje u reci **bajka**). Smatrati da reci nemaju vise od 20 karaktera.