Kolokvijum – april 2013

1.

Prvi argument komandne linije je ime programa, a drugi niska s. Napisati program koji na standardni izlaz prepisuje one linije datoteke(pretpostaviti da ne sadrze vise od 80 karaktera) koje sadrze nisku s.

ULAZ	IZLAZ
program ulaz.txt P2 ulaz.txt Red jedan P02 Drugi red P2 Red treci bez P1 Cetvrti red:P2	Drugi red P2 Cetvrti red:P2

```
#include <stdio.h>
#include <string.h>
#define MAX 81
int main(int argc, char *argv[])
{
if(argc!=3)
{
   fprintf(stderr,"Greska!Nedovoljan broj argumenata komandne linije!\n");
   return 0;
}

FILE *in;
char s[MAX];

if((in=fopen(argv[1],"r"))==NULL){
  fprintf(stderr,"Greska prilikom otvaranja ulazne datoteke!\n");
  return 0;
}

while(fgets(s, MAX, in)!=NULL)
{
if(strstr(s,argv[2])!=NULL)
printf("%s", s);

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

2. **Napisati rekurzivnu funkciju kojom se izracunava na koliko pozicija se binarni zapis neoznacenih celih brojeva a i b poklapa.**

```
#include <stdio.h>

int f(unsigned a,unsigned b,unsigned mask)
{
  if(mask == 0)
    return 0;
  if((a&mask)==(b&mask))
    return 1+f(a,b,mask<<1);
  else
    return f(a,b,mask<<1);
```

```
}


int main()
{
  unsigned a,b;

  scanf("%u%u",&a,&b);

  printf("poklapa se na %d pozicija\n",f(a,b,1));

 return 0;
}
```

3. **Struktura razlomak predstavljena je neoznacenim brojiocem i imeniocem. Napisati funkciju koja skracuje razlomak (za NZD koristiti Euklidov algoritam). Napisati program koji ucitava niz razlomaka(njih najvise 10000), skracuje ih i ispisuje sortirano opadajuce(ili rastuce ukoliko je u komandnoj liniji navedena opcija -r).**

ULAZ	IZLAZ
5 6 3 4 5 100 2 1 3 3 9	50 /1 2 /1 4 /5 1 /3 1 /3
ULAZ -r 5 6 3 4 5 100 2 1 3 3 9	1 /3 1 /3 4 /5 2 /1 50 /1

```
#include <stdio.h>
#include <stdlib.h>
#include<string.h>
#define MAX 10000
typedef struct{
                    unsigned br;
                    unsigned im;
                    }razlomak;

unsigned NZD(unsigned br,unsigned im)
{  /* NZD(18,12)=NZD(12,18%12)=NZD(12, 6)=NZD(6,0)=6*/
 if(im==0) return br;
 else NZD(im,br % im);
}

void skrati(razlomak *z)
{
unsigned a;
if(z->br>z->im) a=NZD(z->br,z->im);
else a=NZD(z->im,z->br);
z->br=z->br/a;
z->im=z->im/a;
```

```
}

int rastuca(const void *a ,const void *b)
{
        razlomak *x= (razlomak *)a;
        razlomak *y= (razlomak *)b;
        if(((float)x->br)/x->im - ((float)y->br)/y->im  > 0)
                        return 1;
        else if((((float)x->br)/x->im - ((float)y->br)/y->im) ==0)
                        return 0;
        else return -1;
}

int opadajuca(const void *a ,const void *b)
{
        razlomak *x= (razlomak *)a;
        razlomak *y= (razlomak *)b;
         if(((float)x->br)/x->im - ((float)y->br)/y->im  > 0)
                        return -1;
        else if((((float)x->br)/x->im - ((float)y->br)/y->im) ==0)
                        return 0;
        else return 1;
}

int main(int argc, char*argv[])
{
razlomak z[MAX];
int i,n;

scanf("%d", &n);
if (n>MAX)
n=MAX;


for(i=0; i<n; i++)
{
scanf("%u%u",&z[i].br,&z[i].im);
skrati (&z[i]);
}

if(argc>1 && !strcmp(argv[1], "-r"))
qsort(z, n, sizeof(razlomak), rastuca);
else
qsort(z, n, sizeof(razlomak), opadajuca);

for(i=0; i<n; i++)
printf("%u /%u \n ",z[i].br,z[i].im);

return 0;
}
```