5 GSL продолжение

1. Да се напише програма која ќе дозволи внесување на 10 реални вредности (до 5та децимална вредност). Да се најдат сите броеви што се исти со првиот до прва, втора, трета и четврта децимална вредност. На располагање е функцијата gsl_fcmp за споредба на реални броеви. Како би се решила задачата доколку не беше на располагање оваа функција?

```
#include <stdio.h>
#include <gsl/gsl_math.h>
#include <gsl/gsl_poly.h>

int main(){

double niza[10]={1.12345, 1.12346, 1.12347, 1.1235, 1.1236, 1.124, 1.125, 1.13, 1.14, 1.2};
int i;
/*
for(i=0;i<10;i++){
scanf("%lg",&niza[i]);
}*/
// baranje do nulta decimalna vrednost
printf("\n\nBroevi isti do nulta decimalna vrednost so brojot %g\n",niza[0]);
for(i=1;i<10;i++){
if(gsl_fcmp(niza[0],niza[i],0.1)==0) printf("%lg ", niza[i]);
}
// baranje do prva decimalna vrednost
printf("\n\nBroevi isti do prva decimalna vrednost so brojot %g\n",niza[0]);
for(i=1;i<10;i++){
if(gsl_fcmp(niza[0],niza[i],0.01)==0) printf("%lg ", niza[i]);
}
// baranje do vtora decimalna vrednost
printf("\n\nBroevi isti do vtora decimalna vrednost so brojot %g\n",niza[0]);
for(i=1;i<10;i++){
if(gsl_fcmp(niza[0],niza[i],0.001)==0) printf("%lg ", niza[i]);
}
// baranje do treta decimalna vrednost
printf("\n\nBroevi isti do treta decimalna vrednost so brojot %g\n",niza[0]);
for(i=1;i<10;i++){
if(gsl_fcmp(niza[0],niza[i],0.0001)==0) printf("%lg ", niza[i]);
}
// baranje do cetvrta decimalna vrednost
printf("\n\nBroevi isti do cetvrta decimalna vrednost so brojot %lg\n",niza[0]);
for(i=1;i<10;i++){
if(gsl_fcmp(niza[0],niza[i],0.00001)==0) printf("%lg ", niza[i]);
}

return 0;
}
```

2. Да се напише програма која ќе овозможи читање (од тастатура) на вектор (поле) со должина N, каде што N исто така се внесува од тастатура. Програмот треба да креира нов вектор користејќи го постоечкиот така што, на секоја позиција во векторот ќе го смести збирот на тековниот елемент и елементот десно од него (последниот елемент не ја менува својата вредност). Потоа, да се најдат и испечатат на екран минималниот и максималниот елемент на двата вектори, заедно со позициите на кои се наоѓаат.

   НАПОМЕНА: влезниот вектор на почеток е пополнет со нули, додека пак новиот вектор е пополнет со единици.

```
#include <stdio.h>
#include <gsl/gsl_vector.h>

int main(){
gsl_vector *vlez;
gsl_vector *transform;
double vrednost;
double min1,max1, min2, max2;
int poz_min1, poz_min2, poz_max1, poz_max2;
int n;
int i,j;

printf("Vnesi ja dolzinata na vektorot");
scanf("%d",&n);
vlez=gsl_vector_calloc(n);
transform=gsl_vector_calloc(n);
gsl_vector_add_constant(transform,1.0);

for(i=0;i<n;i++){
scanf("%lf",&vrednost);
gsl_vector_set(vlez,i,vrednost);
}
for(i=0;i<n-1;i++){
vrednost = gsl_vector_get(vlez,i)+gsl_vector_get(vlez,i+1);
gsl_vector_set(transform,i,vrednost);
}
gsl_vector_set(transform,n-1,gsl_vector_get(vlez,n-1));

for(i=0;i<n;i++){
printf("%.1lf ",gsl_vector_get(vlez,i));
}
printf("\n\n");
for(i=0;i<n;i++){
printf("%.1lf ",gsl_vector_get(transform,i));
}
printf("\n\n");
min1=gsl_vector_min(vlez);
min2=gsl_vector_min(transform);
max1=gsl_vector_max(vlez);
max2=gsl_vector_max(transform);
poz_min1=gsl_vector_min_index(vlez);
poz_min2=gsl_vector_min_index(transform);
poz_max1=gsl_vector_max_index(vlez);
poz_max2=gsl_vector_max_index(transform);
```

```
printf("min[%d]=%lf       max[%d]=%lf\n",poz_min1, min1, poz_max1, max1);
printf("min[%d]=%lf       max[%d]=%lf\n",poz_min2, min2, poz_max2, max2);
gsl_vector_free(vlez);
gsl_vector_free(transform);
return 0;
}
```

3. Да се напише програма која од датотека со име „vlez.txt" ќе прочита два вектори. На почетокот на редот се наоѓа секогаш должината на векторот, додека пак во продолжение на истиот ред се наоѓа векторот. Вториот вектор се наоѓа во вториот ред. Програмата треба да изврши сортирање на векторите во растечки редослед. Потоа, програмот треба да ги запише во излезна датотека „izlez.txt" векторите најпрво во растечки а потоа во опаѓачки редослед.

```
#include<stdio.h>
#include<gsl/gsl_vector.h>

int main(){

gsl_vector *ve1;
gsl_vector *ve2;
FILE *vlez,*izlez;
int dolzina1, dolzina2,i,j;
if((vlez=fopen("vlez.txt","r"))==NULL){printf("vlez.txt nemozam da ja otvoram za citanje\n");return -1;}
if((izlez=fopen("izlez.txt","w"))==NULL){printf("izlez.txt nemozam da ja otvoram za zapisuvanje\n");return -1;}

fscanf(vlez,"%d ",&dolzina1);
ve1=gsl_vector_calloc(dolzina1);
gsl_vector_fscanf(vlez,ve1);
fscanf(vlez,"\n%d ",&dolzina2);
ve2=gsl_vector_calloc(dolzina2);
gsl_vector_fscanf(vlez,ve2);

for(i=0;i<dolzina1;i++)
{printf("%lg ",gsl_vector_get(ve1,i));}
printf("\n");
printf("%d\n",dolzina2);
for(i=0;i<dolzina2;i++)
{printf("%lg ",gsl_vector_get(ve2,i));}
printf("\n");

gsl_sort_vector(ve1);
gsl_sort_vector(ve2);
gsl_vector_fprintf(izlez,ve1,"%lg ");
fprintf(izlez,"\n");
gsl_vector_fprintf(izlez,ve2,"%lg ");
fprintf(izlez,"\n");

gsl_vector_reverse(ve1);
gsl_vector_reverse(ve2);
gsl_vector_fprintf(izlez,ve1,"%lg ");
fprintf(izlez,"\n");
gsl_vector_fprintf(izlez,ve2,"%lg ");
```

```
fprintf(izlez,"\n");
for(i=0;i<dolzina1;i++)
{printf("%lg ",gsl_vector_get(ve1,i));}
printf("\n");
for(i=0;i<dolzina2;i++)
{printf("%lg ",gsl_vector_get(ve2,i));}
printf("\n");

gsl_vector_free(ve1);
gsl_vector_free(ve2);
fclose(vlez);fclose(izlez);
}
```

4. Да се напише програма која од тастатура ќе ги прочита димензиите на една матрица. Потоа, да креира три матрици со истите димензии, користејќи ги функциите на GSL дефинирани во gsl_matrix.h. Првата матрица треба да ги има сите елементи единици, втората нули, додека пак третата да биде кориснички внесена од тастатура. Првата матрица да се трансформира во идентитет матрица. На крај, сите матрици да се запишат во датотеката „vlez.txt“.

```
#include<stdio.h>
#include<gsl/gsl_matrix.h>

int main(){
gsl_matrix *m1,*m2,*m3;
FILE *dat;
int i,j;
int n,m;
double temp;
if((dat=fopen("vlez1.txt","w"))==NULL){printf("nemozam da ja otvoram datotekata za zapisuvanje\n");return -1;}
printf("Vnesi gi dimenziite na matricite\n");
scanf("%d%d",&n,&m);

m1=gsl_matrix_alloc(n,m);
m2=gsl_matrix_calloc(n,m);
m3=gsl_matrix_alloc(n,m);
gsl_matrix_set_all(m1,1.0);
gsl_matrix_set_zero(m2);
gsl_matrix_set_identity(m1);
for(i=0;i<n;i++){
for(j=0;j<m;j++){
printf("a[%d][%d]=",i,j);
scanf("%lg",&temp);
gsl_matrix_set(m3,i,j,temp);
}
}
for(i=0;i<n;i++){
for(j=0;j<m;j++){
printf("a[%d][%d]=%lg\t",i,j,gsl_matrix_get(m1,i,j));}
}
for(i=0;i<n;i++){
for(j=0;j<m;j++){
printf("a[%d][%d]=%lg\t",i,j,gsl_matrix_get(m2,i,j));
}
```

```
printf("\n");
}
printf("\n");
for(i=0;i<n;i++){
for(j=0;j<m;j++){
printf("a[%d][%d]=%lg\t",i,j,gsl_matrix_get(m3,i,j));}
}
gsl_matrix_fprintf(dat,m1,"%lg ");
gsl_matrix_fprintf(dat,m2,"%lg ");
gsl_matrix_fprintf(dat,m3,"%lg ");
fclose(dat);
gsl_matrix_free(m1);
gsl_matrix_free(m2);
gsl_matrix_free(m3);
return 0;
}
```

5. Од датотека со име „vlez.txt“ да се прочита матрица со големина 3x3. Најпрвин, елементите од секоја колона одделно да бидат сортирани во растечки редослед. Потоа, да се дефинира вектор во кој ќе се сместат најмалите елементи од секоја редица. Елементите на онаа редица што ќе го има најмалиот елемент од сите, да се намалат за 1. Да се провери дали дадениот елемент е најмал елемент за целата матрица и да се испечати.

```
#include<stdio.h>
#include<gsl/gsl_matrix.h>
#include<gsl/gsl_vector.h>
int main(){
FILE *dat;
gsl_matrix *m=gsl_matrix_alloc(3,3);
gsl_vector *v=gsl_vector_alloc(3);
gsl_vector *mali=gsl_vector_alloc(3);
int i,j,najmal;

if((dat=fopen("vlez1.txt","r"))==0){printf("Nemozam da ja otvoram za citanje\n");return -1;}

gsl_matrix_fscanf(dat,m);

for(i=0;i<3;i++){

gsl_matrix_get_col(v,m,i);
gsl_sort_vector(v);
gsl_matrix_set_col(m,i,v);
}
for(i=0;i<3;i++){
gsl_matrix_get_row(v,m,i);
gsl_vector_set(mali,i,gsl_vector_min(v));
printf("%lg ",gsl_vector_get(mali,i));
}
najmal=gsl_vector_min_index(mali);
for(i=0;i<3;i++){
gsl_matrix_set(m,najmal,i,gsl_matrix_get(m,najmal,i)-1);
}
for(i=0;i<3;i++)
for(j=0;j<3;j++)
printf("%lg ",gsl_matrix_get(m,i,j));
```

```
gsl_matrix_free(m);
gsl_vector_free(v);
gsl_vector_free(mali);
return 0;
}
```

6. Да се напише програма за решавање на систем равенки со непознати. Програмата да дозволи најпрво внесување на бројот на равенки од кои е составен системот, а потоа, наведување само на коефициентите на равенките. Решението да се прикаже на екран.

```
#include<stdio.h>
#include<gsl/gsl_matrix.h>
#include<gsl/gsl_vector.h>
#include<gsl/gsl_linalg.h>
#include<gsl/gsl_cblas.h>
```

$$\begin{cases} x+y+z=6 \\ 2y+5z=-4 \\ 2x+5y-z=27 \end{cases} \quad \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 5 \\ 2 & 5 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 6 \\ -4 \\ 27 \end{bmatrix} \quad \text{Решение: } \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 5 \\ 2 & 5 & -1 \end{bmatrix}^{-1} \begin{bmatrix} 6 \\ -4 \\ 27 \end{bmatrix}$$

$$\mathrm{A} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 5 \\ 2 & 5 & -1 \end{bmatrix} \quad X = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad B = \begin{bmatrix} 6 \\ -4 \\ 27 \end{bmatrix} \quad Ainv = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 5 \\ 2 & 5 & -1 \end{bmatrix}^{-1}$$

$$X = \begin{bmatrix} 5 \\ 3 \\ -2 \end{bmatrix}$$

```
int main(){
gsl_matrix *A;
gsl_matrix *X;
gsl_matrix *B;
gsl_matrix *Ainv;
gsl_permutation *perm;
double temp;
int i,j, n;
printf("vnesi go redot na sistemot ravenki\n");
scanf("%d",&n);
A=gsl_matrix_alloc(n,n);
X=gsl_matrix_calloc(n,1);
B=gsl_matrix_alloc(n,1);

perm=gsl_permutation_alloc(n);
Ainv=gsl_matrix_alloc(n,n);
for(i=0;i<n;i++){
printf("\nVnesi gi koef za ravenka %d",i);
for(j=0;j<n;j++){
scanf("%lg",&temp);
gsl_matrix_set(A,i,j,temp);
}
printf("\nVnesi go reshenieto:");
scanf("%lg",&temp);
gsl_matrix_set(B,i,0,temp);
}
gsl_linalg_LU_decomp(A, perm,&j);
```

```
gsl_linalg_LU_invert(A,perm,Ainv);
for(i=0;i<n;i++){
for(j=0;j<n;j++)printf("%lg ",gsl_matrix_get(Ainv,i,j));
printf("\n");
}
for(i=0;i<n;i++){
temp=0;
for(j=0;j<n;j++){
temp+=gsl_matrix_get(Ainv,i,j)*gsl_matrix_get(B,j,0);
}
gsl_matrix_set(X,i,0,temp);
}
printf("\n");

for(j=0;j<n;j++)printf("%lg ",gsl_matrix_get(X,j,0));
printf("\n");

gsl_matrix_free(A);
gsl_matrix_free(X);
gsl_matrix_free(B);
gsl_matrix_free(Ainv);
gsl_permutation_free(perm);
return 0;
}
```