

# Strukture podataka i algoritmi 2

školska 2009/10

Napisati program u programskom jeziku C za rad sa kompleksnim brojevima. Program treba da sadrži i potprogram za sabiranje dva kompleksna broja i potprogram za konjugovanje kompleksnog broja.

```
#include<stdio.h>
#include<ctype.h>
#include<stdlib.h>
#define TRUE 1
#define FALSE 0
typedef struct Scomplex {
    double Re, Im;
} TComplex;
int DaLiJeBroj(char *pstr) {
    int bTacka = FALSE;
    if (*pstr=='+' || *pstr=='-') ++pstr;
    while(*pstr) {
        if (!isdigit(*pstr) && *pstr!='.') return FALSE;
        if (*pstr == '.') {
            if (bTacka) return FALSE;
            bTacka = TRUE; }
        pstr++; }
    return TRUE;
}
```

Napisati program u programskom jeziku C za rad sa kompleksnim brojevima. Program treba da sadrži i potprogram za sabiranje dva kompleksna broja i potprogram za konjugovanje kompleksnog broja.

```
TComplex Zbir(TComplex z1, TComplex z2) {
    TComplex cplx;
    cplx.Re = z1.Re + z2.Re;
    cplx.Im = z1.Im + z2.Im;
    return cplx;
}

void Konjug(TComplex *pZ) {
    pZ->Im = - pZ->Im;
}

void Ispis(TComplex z) {
    if (z.Im >= 0) printf("%5.4f + j%5.4f", z.Re, z.Im);
    else printf("%5.4f - j%5.4f", z.Re, -z.Im);
}
```

Napisati program u programskom jeziku C za rad sa kompleksnim brojevima. Program treba da sadrži i potprogram za sabiranje dva kompleksna broja i potprogram za konjugovanje kompleksnog broja.

```
int main(int argc, char *argv[]) {
    TComplex z1, z2, zbir;
    int i;
    if (argc != 5) {
        printf("Unesite tacno cetiri argumenata pri startu programa!\n");
        return -1;
    }
    printf("Uneli ste sledece argumente preko komandne linije:\n");
    for (i=1; i < argc; i++) printf("%d %s\n", i, argv[i]);
    for (i=1; i < argc; i++) {
        if (!DaLiJeBroj(argv[i])) {
            printf("Uneli ste argument koji nije broj!\n");
            return -1;
        }
    }
}
```

Napisati program u programskom jeziku C za rad sa kompleksnim brojevima. Program treba da sadrži i potprogram za sabiranje dva kompleksna broja i potprogram za konjugovanje kompleksnog broja.

```
z1.Re = atof(argv[1]);
z1.Im = atof(argv[2]);
z2.Re = atof(argv[3]);
z2.Im = atof(argv[4]);
printf("\nPrvi broj: "); Ispis(z1);
printf("\nDrugi broj: "); Ispis(z2);
zbir = Zbir(z1, z2);
Konjug(&z1); Konjug(&z2);
printf("\nZbir: "); Ispis(zbir);
printf("\nKonjug(z1): "); Ispis(z1);
printf("\nKonjug(z2): "); Ispis(z2);
return 0;
}
```

# Zadaci za uvod

- Elektronski sat pokazuje tekuće vreme u satima, minutima i sekundama. Ispisati sve momente vremena u toku jednog dana takve da je suma cifara na časovniku jednaka datom broju  $n$ . Ispis treba da je u obliku (za  $n=6$ ):

0 sat 11 min 31 sec

1 sat 32 min 0 sec

...

# Zadaci za uvod

```
#include <stdio.h>
main() {
    int sat,min,sec,n;
    scanf("%d",&n);
    for(sat=0;sat<24;sat++)
        for(min=0;min<60;min++)
            for(sec=0;sec<60;sec++) {
                if(((sat%10)+(sat/10)+(min%10)+(min/10)+
                    (sec%10)+(sec/10))==n)
                    printf("%d sat %d min %d sec\n",sat,min,sec);
            }
}
```

# Zadaci za uvod

- Napisati program kojim se na osnovu rezultata trkača na 100m datih nizom:  $a_1, a_2, \dots, a_n$  (indeks elementa niza odgovara startnom broju) formira niz:  $b_1, b_2, \dots, b_n$  gde je  $b_i$  indeks (startni broj) takmičara koji se plasirao na  $i$ -to mesto.

Na primer, ako je niz  $a$ : 13.6, 13.3, 12.9, 13.1, tada je niz  $b$ : 3, 4, 2, 1.



# Zadaci za uvod

```
#include <stdio.h>
main() {
    int n,i,j,p,b[100];
    double a[100],q;
    scanf("%d",&n);
    for(i=0;i<n;i++) {
        scanf("%f",&a[i]);
        b[i]=i+1;
    }
    for(i=0;i<n-1;i++)
        for(j=i+1;j<n;j++)
            if(a[i]>a[j]) {
                q=a[i]; a[i]=a[j]; a[j]=q;
                p=b[i]; b[i]=b[j]; b[j]=p;
            }
    for(i=0;i<n;i++) printf("%5d %7.2f\n",b[i],a[i]);
}
```

# Zadaci za uvod

- Napisati program kojim se generiše niz od  $N$  slučajnih brojeva na zatvorenom intervalu  $[a,b]$  i određuje vrednost elementa niza najbližeg srednjoj vrednost  $L$ .

# Zadaci za uvod

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
main() {
    int i,n,k,a,b,x[20],ind; double s;
    printf("Unesi broj clanova niza koji se generisu \n");
    scanf("%d",&n);
    printf("Unesi interval [a,b] na kom se generisu:\n");
    scanf("%d%d",&a,&b);
    srand(time(NULL));
    s=0;
    for(i=0;i<n;i++) {
        x[i]=a+(b-a)*rand()/RAND_MAX;
        s=s+x[i];
    }
    for(i=0;i<n;i++) printf("%d ",x[i]); printf("\n");
    s=s/n; k=0;
    for(i=1;i<n;i++)
        if(fabs(x[k]-s) < fabs(x[i]-s)) k=i;
    printf("Najblizi srednjoj vrednosti: %.2f je: %d\n",s,x[k]);
}
```

# Zadaci za uvod

- Malog Mirka je mama poslala u kupovinu sa spiskom od  $N$  proizvoda koje treba da kupi i nekoliko **bonova za popust**. Mirko ima ukupno  $K$  bonova za popust, i svaki od tih  $K$  bonova uvek smanjuje cenu proizvoda, koji misli kupiti, za **15%**. Mirko želi što jeftinije da kupi sve proizvode sa spiska, pa je zaključio da treba što pametnije da rasporediti bonove za popust kako bi sa što manje novca uspeo sve da kupi. Mirko sme iskoristiti **više bonova na istom proizvodu** kako bi smanjio cenu toga proizvoda. Ukoliko se koristi više bonova za popust na istom proizvodu, onda svako novo sniženje smanjuje cenu za 15% u odnosu na cenu nastalu predhodnim korišćenjem bonova. Napiši program koji uz spisak cena proizvoda i broja bonova izračunava najmanju sumu potrebnu da kupi sve proizvode sa spiska ukoliko na najbolji način rasporedi bonove za popust.

|        |         |
|--------|---------|
| 2 3    | 4 1     |
| 100 85 | 4 6 9 5 |
| 144.5  | 22.65   |