

8. STRUCTURI SI UNIUNI

A. Probleme rezolvate

1. Se considera un vector de n componente complexe: z_1, z_2, \dots, z_n , unde $z_i = a_i + j * b_i$, $i=1, \dots, n$. Folosind un tip de date structura pentru reprezentarea numerelor complexe, sa se determine norma vectorului data de relatia: $d = |z_1 + z_2 + \dots + z_n|$, unde $|z|$ reprezinta modulul numarului complex z .

Rezolvare

```
#include <stdio.h>
#include <math.h>
struct compl
{
    double re, im;
};
double norma(struct compl z)
{
    return sqrt(z.re*z.re + z.im*z.im);
}
struct compl suma(struct compl z1, struct compl z2)
{
    struct compl z;
    z.re = z1.re + z2.re;
    z.im = z1.im + z2.im;
    return z;
}
int main() {
    int n, i;
    struct compl z[10], s;
    double d;
    printf("\nNumarul de elemente al vectorului: ");
    scanf("%d", &n);
    printf("\nElementele vectorului:\n");
    for(i=0; i<n; ++i) {
        printf("\nz[%d].re: ", i+1);
        scanf("%lf", &z[i].re);
        printf("\nz[%d].im: ", i+1);
        scanf("%lf", &z[i].im);
    }
    s.re = s.im = 0;
    for(i=0; i<n; ++i)
        s = suma(s, z[i]);
    d = norma(s);
    printf("\nNorma = %lf", d);
    return 0;
}
```

2. Se considera multimile de numere intregi A, B, C si D. Sa se determine multimea:

$$E = (A \cup B) - (A \cap B).$$

Se va utiliza un tip de date structura pentru reprezentarea multimilor alocate dinamic.

Rezolvare

```
#include <stdio.h>
#include <stdlib.h>
#define M 10
typedef struct set {
    int *val;
    int n;
} Set;
typedef Set* PSet;
int Inclus(int x, Set a) {
    int i;
    for (i=0; i<n ; ++i)
        if (x == a.val[i])
            return 1;
    return 0;
}
PSet MultimeVida() {
    PSet pa;
    pa = (PSet)malloc(sizeof(Set));
    pa->val = 0;
    pa-> n=0;
    return pa;
}
PSet AdaugaElement(PSet pa, int x) {
    if(!Inclus(x, *pa)) {
        int i;
        int *pv = (int*)malloc(sizeof(int)*(pa->n+1));
        for (i=0; i<pa->n; ++i) {
            pv[i] = pa->val[i];
        }
        pv[pa->n++] = x;
        free(pa->val);
        pa->val = pv;
    }
    return pa;
}
Set CitireMultime() {
    PSet pa;
    int x, i, n;
    pa = MultimeVida();
    printf("\nNumarul de elemente: ");
    scanf("%d",&n);
    for(i=0; i<n; ++i) {
        printf("\nElement = ");
        scanf("%d",&x);
        pa = AdaugaElement(pa, x);
    }
    return *pa;
}
void ScrieMultime(Set a) {
    int i=0;
    printf ("{ ");
    for (i=0; i<a.n; ++i)
```

```

        printf ("%d ", a.val[i]);
    printf (" }\n");
}
Set Reuniune(Set a, Set b) {
    int i;
    PSet pc;
    pc = MultimeVida();
    for(i=0; i<a.n; ++i)
        pc = AdaugaElement(pc, a.val[i]);
    for (i=0; i<b.n; ++i)
        pc = AdaugaElement(pc, b.val[i]);
    return *pc;
}
Set Intersectie(Set a, Set b) {
    int i;
    PSet pc;
    pc = MultimeVida();
    for (i=0; i<a.n; ++i)
        if(Inclus(a.val[i], b))
            pc = AdaugaElement(pc, a.val[i]);
    return *pc;
}
Set Diferenta(Set a, Set b) {
    int i;
    PSet pc;
    pc = MultimeVida();
    for (i=0; i<a.n; ++i)
        if(!Inclus(a.val[i], b))
            pc = AdaugaElement(pc, a.val[i]);
    return *pc;
}
int main () {
    struct set a, b, c, d, e ;
    printf("\nCitire multimea A:\n");
    a = CitireMultime ();
    printf("\nCitire multimea B:\n");
    b = CitireMultime();
    printf("\nMultimea A: ");
    ScrieMultime(a);
    printf("\nMultimea B: ");
    ScrieMultime(b);
    c = Reuniune(a, b);
    d = Intersectie(a, b);
    e = Diferenta(c, d);
    printf("\nReuniune: ");
    ScrieMultime(c);
    printf("\nIntersectie: ");
    ScrieMultime(d);
    printf("\nDiferenta simetrica: ");
    ScrieMultime(e);
    return 0;
}

```

B. Probleme propuse

1. Se considera un vector x de n componente complexe: z_1, z_2, \dots, z_n , unde $z_i = a_i + j \cdot b_i$, $i = 1, \dots, n$. Folosind un tip de date structura pentru memorarea partilor reale si imaginare ale numerelor complexe, sa se determine norma vectorului data de relatia: $d = \max(|z_1|, |z_2|, \dots, |z_n|)$.

2. Folodind un tip de date structura pentru reprezentarea numerelor complexe, sa se scrie un program pentru calculul valorii complexe a unui polinom de variabila complexa, cu coeficienti numere complexe.

3. Se considera ca o societate comerciala isi tine evidenta produselor din stoc folosind un vector cu inregistrari. Fiecare inregistrare este aferenta unui produs si contine urmatoarele informatii: nume produs, cantitate de produse de acelasi tip si pret produs. Sa se realizeze un program care citeste de la tastatura informatiile despre produsele din stocul unei societati comerciale si efectueaza operatiile:

- a) creaza vectorul aferent produselor;
- b) determina valoarea totala a produselor din stoc;
- c) determina numarul de produse din stoc cu o valoare mai mare decat o valoare data;
- d) determina produsele din stoc cu o cantitate mai mica decat un numar dat;
- e) adaugarea in stoc a unui nou produs;
- f) adaugarea in stoc a unei noi cantitati de produse dintr-un produs existent;
- g) stergerea din stoc a unei cantitati dintr-un anumit produs.

4. Sa se scrie un program, `calc.c`, care citeste de la tastatura numele unei functii matematice si argumentele sale pe o linie de comanda. Se va folosi urmatoarea structura pentru a crea un tablou de functii matematice asociat functiilor din biblioteca standard:

```
struct fmat (  
    char *nume;          /* numele functiei matematice */  
    double (*pfm) ();   /* pointer spre functia din biblioteca standard */  
    int nargs;         /* numarul de argumente al functiei */  
);
```

Folositi doar functii matematice cu parametri de tip `double`, care returneaza un rezultat de tip `double`. Exemple de folosire a programului `calc.c`:

```
> calc sqrt 25  
> 5  
> calc pow 2.5 2  
> 6.25  
> calc exp .5  
> 1.64872
```

5. Se considera o data calendaristica specificata printr-o structura, cu campurile *an*, *luna*, *zi*. Sa se scrie un program care citeste si creeaza un tablou de date calendaristice, apoi le ordoneaza crescator si le afiseaza. Se va folosi o functie care compara doua date calendaristice.

6. Sa se defineasca o structura *Punct* pentru un punct din plan (de coordonate reale), o structura *Poligon* pentru un poligon definit prin numarul de varfuri si un vector de puncte ce reprezinta varfurile sale, precum si doua functii: una pentru calculul distantei dintre doua puncte, iar a doua pentru calculul perimetrului unui poligon. Sa se scrie un program pentru calculul perimetrului unui triunghi si al unui patrulater.

7. Sa se defineasca o structura *Pereche* pentru o pereche *(sir de caractere, numar intreg)* si o structura *TablouAsociativ* pentru un vector perechi. Sa se scrie un program care permite realizarea urmatoarelor operatii:

- Citirea de la tastatura a unui sir de cuvinte cuvinte (separate prin spatiu) si crearea tabloului asociativ, in care, pentru fiecare cuvint distinct, se specifica numarul de aparitii al acestuia,
- Citirea de la tastatura a unui cuvint, cautarea acestuia in tabelul asociativ si afisarea numarului de aparitii ale acestuia in sirul initial de cuvinte,

- Citirea de la tastatura a unui cuvânt și actualizarea tabelului asociativ adăugarea unei perechi <cuvânt, număr> la dicționar (în cazul în care cuvântul nu există în tabelul asociativ) sau incrementarea numărului de apariții ale cuvântului (în cazul în care cuvântul este găsit),
- Afișarea tabelului asociativ.

Vectorul de perechi se poate realiza cu alocare dinamică, sau cu alocare statică (în acest caz trebuie specificat numărul curent de elemente din vector).

8. Să se crie un program care realizează următoarele operații:

- Citește de la tastatură un șir de numere reale și creează cu acestea o listă liniară simplu legată, conținând numerele în aceeași ordine ca la introducerea,
- Elimină din listă toate numerele nule,
- Afișează elementele rămase în listă.