

5. POINTERI SI SIRURI DE CARACTERE

A. Probleme rezolvate

1. Urmatorul program incrementeaza fiecare element al unui tablou de numere intregi.

```
#include <stdio.h>
int main() {
    static int t [2] [3] = {
                                1, 2, 3,
                                4, 5, 6
                            };

    int *P;
    for (p = (int *)t; p < (int *)t + 6;)
        ++*P++;
    for (p = (int *)t; p < (int *)t + 6;)
        printf("%2d", *P++);
    putchar('\n');
    return 0;
}
```

2. Sa se scrie un program (asemanator cu functia *strncmp*), care compara lexicografic cel mult *n* caractere ale sirurilor *s1* si *s2*.

```
#include <stdio.h>
int main() {
    char *s1, *s2;
    int n;
    printf("\nprimul sir = ");
    scanf("%s", &s1);
    printf("\nal doilea sir = ");
    scanf("%s", &s2);
    printf("\nnumarul de caractere:\n");
    scanf("%d", &n);
    while (n>0 && *s1 && *s2 && *s1 == *s2 ) {
        n--;
        ++s1;
        ++s2;
    }
    if (n==0 || (*s1==0 && *s2==0))
        printf("\s1 = s2");
    else if (*s1 < *s2)
        printf("\ns1 < s2");
    else
        printf("\ns1 > s2");
    return 0;
}
```

B. Probleme propuse

1. Sa se scrie un program care codifica un sir de caractere citit de la terminalul standard de intrare si il afiseaza apoi la terminalul standard de iesire. Operatie de codificare presupune urmatoorii pasi, pentru fiecare caracter din sirul de

intrare:

- se convertește caracterul în codul său ASCII echivalent;
- la valoarea respectivă se adună un număr întreg generat aleator (același număr se adaugă la toate caracterele șirului);
- notând cu n_1 valoarea ASCII cea mai mică a unui caracter afișabil, iar cu n_2 valoarea cea mai mare, dacă valoarea rezultată după adunare depășește pe n_2 , atunci din numărul respectiv se scade cel mai mare multiplu posibil de n_2 din acest număr (care nu depășește numărul inițial) și se adaugă n_1 la numărul rezultat; în acest fel valoarea rezultată este între n_1 și n_2 ;
- se afișează caracterul ASCII corespunzător valorii rezultate.

2. Să se scrie un program care realizează decodificarea unui șir de caractere codificat conform problemei anterioare (se va utiliza același număr aleator ca la codificare).

3. Să se scrie un program care citește de la terminalul standard de intrare un șir de maxim 80 de caractere, reprezentând cuvinte separate prin spații, iar apoi afișează la terminalul standard de ieșire un șir de aceeași lungime, în care fiecare cuvânt are primul și ultimul caracter transpuse. De exemplu, cuvântul „program” va fi afișat „mrograp”. Se presupune să textul conțină doar caractere alfabete și spații.

4. Să se scrie un program care citește n linii de șiruri de caractere de la terminalul standard de intrare, le sortează în ordine lexicografică și le afișează apoi la terminalul standard de ieșire.

5. Se consideră un șir de caractere s , precum și alte două șiruri, $s1$ și $s2$. Să se scrie un program care înlocuiește toate aparițiile (în șirul s) ale șirului $s1$ prin șirul $s2$.

6. Să se scrie un program care citește două mulțimi de numere întregi, determină și afișează reuniunea acestora. Se vor folosi date alocate dinamic pentru mulțimi.

7. Să se scrie un program care citește două mulțimi de numere întregi, determină și afișează intersecția acestora. Se vor folosi date alocate dinamic pentru mulțimi.

8. Să se scrie un program care citește două polinoame cu coeficienți reali, calculează și afișează suma acestora. Se vor folosi date alocate dinamic pentru vectorul coeficienților (se vor alocă doar valorile nenule ale coeficienților).

9. Să se scrie un program care citește două polinoame cu coeficienți reali, calculează și afișează diferența acestora. Se vor folosi date alocate dinamic pentru vectorul coeficienților (se vor alocă doar valorile nenule ale coeficienților).