

PROGRAMARE PROCEDURALĂ – LABORATOR NR. 7–

1. Să se afișeze toate pozițiile pe care apare un caracter dat într-un cuvânt sau un mesaj corespunzător dacă respectivul caracter nu apare deloc în cuvânt.

Varianta 1 (fără funcții din `string.h`):

```
#include <stdio.h>

int main()
{
    char str[21];
    char ch;

    printf("Introduceti un cuvânt: ");
    scanf("%s", str);
    scanf("%*c");

    printf("Introduceti un caracter: ");
    scanf("%c", &ch);

    int cnt = 0;

    printf("\nPozitiile pe care apare '%c' in \"%s\" sunt: ", ch, str);
    for(int i = 0; str[i] != '\0'; i++)
        if(ch == str[i])
        {
            printf("%d ", i);
            cnt++;
        }

    if(cnt == 0)
        printf("-");

    printf("\n");

    return 0;
}
```

Varianta 2 (cu funcții din string.h):

```
#include <stdio.h>
#include <string.h>

int main()
{
    char *pch, str[21], ch;

    printf("Introduceti un cuvant: ");
    scanf("%s", str);
    scanf("%c", &ch);

    printf("Introduceti un caracter: ");
    scanf("%c", &ch);

    int cnt = 0;
    printf("\nPozitiile pe care apare '%c' in \"%s\" sunt: ", ch, str);

    pch = strchr(str, ch);
    while (pch != NULL)
    {
        printf("%d ", pch-str);
        pch = strchr(pch+1, ch);
        cnt++;
    }

    if(cnt == 0)
        printf("-");

    printf("\n");

    return 0;
}
```

2. Scrieți un program care citește de la tastatură un cuvânt format doar litere mici și afișează poziția ultimei consoane din el. Dacă aceasta nu există, se va afișa mesajul "Nu există nicio consoană!".

Varianta 1 (calculăm maximul dintre pozițiile maxime pe care apare fiecare consoană):

```
#include <stdio.h>
#include <string.h>

int main() {
    char string[50];
    char consoane[] = "bcd fghjklmnpqrstvwxyz";
```

```

int pozMax = -1;

printf("Introduceti un cuvant: ");
scanf("%s", string);

for(int i = 0; i < strlen(consoane); i++) {
    char* ptr = strchr(string, consoane[i]);
    if (ptr != NULL)
        if (pozMax < (ptr - string))
            pozMax = ptr - string;
}

if (pozMax == -1)
    printf("Nu exista nicio consoana!\n");
else
    printf("Ultima consoana din sirul \"%s\" este '%c'!\n",
           string, string[pozMax]);

return 0;
}

```

Varianta 2 (determinăm prima poziție de la dreapta spre stânga pe care o consoană):

```

#include <stdio.h>
#include <string.h>

int main()
{
    int i;
    char string[50];
    char consoane[] = "bcd fghjklmnpqrstvwxyz";

    printf("Introduceti un cuvant: ");
    scanf("%s", string);

    for(i = strlen(string)-1; i >= 0; i--)
        if (strchr(consoane, string[i]) != NULL) {
            printf("Ultima consoana din sirul \"%s\" este '%c'!\n",
                   string, string[i]);
            break;
        }

    if (i == -1)
        printf("Nu exista nicio consoana!\n");

    return 0;
}

```

3. Scrieți un program care șterge toate vocalele aflate într-un șir de caractere.

Rezolvare:

```
#include <stdio.h>
#include <string.h>

int main()
{
    char sir[101];
    int i;

    printf("Introduceti un sir: ");
    fgets(sir, 101, stdin);

    if(sir[strlen(sir) - 1] == '\n')
        sir[strlen(sir) - 1] = '\0';

    i = 0;
    while(sir[i] != '\0')
        if(strchr("aeiouAEIOU", sir[i]) != NULL)
            strcpy(sir+i, sir+i+1);
        else
            i++;

    printf("\nSirul fara vocale: %s\n", sir);

    return 0;
}
```

4. Se citesc, pe rând, mai multe șiruri de caractere. Să se afișeze șirurile de lungime maximă. De exemplu, dacă se vor citi șirurile "mere", "tei" și "pere" se va afișa șirul "mere pere".