

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

/* deklaracje funkcji przetwarzania tekstu */
char * str_chsort(char *str); // sortuje alfabetycznie znaki w łańcuchu znaków
int str_chmin(char *str); // zwraca pozycję najmniejszej litery

/* dwie tablice znaków - miejsce na łańcuchy znaków */
char cbuf[1024];
char rbuf[1024];

int main()
{
    printf( "\nTEST funkcji przetwarzania tekstu");

    printf( "\npodaj tekst źródłowy: "); // aby użytkownik mógł wprowadzić całe zdanie
    fgets(cbuf, sizeof(cbuf), stdin); // czyta do znaku nowej linii włącznie z nim, ale nie więcej
    znaków niż podano // inaczej: scanf("%1023[^\n]s", cbuf); czy działa w gcc?

    printf( "\nPodano: %s", cbuf);

    strcpy(rbuf,cbuf); // kopiuj do drugiego bufora
    str_chsort(rbuf); // sortuj litery

    printf( "\nPosortowane litery: %s", rbuf);

    printf( "\nNacisnij dowolny klawisz!");
    _getch();

    return 0;
}

/* funkcja sortuje znaki w łańcuchu znaków
funkcja realizuje algorytm sortowania przez selekcję
funkcja sortuje "w miejscu" tzn. zmienia łańcuch znaków, do którego adres dostała
najprostszy algorytm (nie jest stabilny)
*/
char* str_chsort(char *str)
{
    int k=0, // * indeks znaku */
        i=0; // * indeks znaku najmniejszego - minimum */
    char tmpch=' '; // * bufor na przerzucenie znaku */

    for(k=0; str[k]!=0; k++)
    {
        /* znajdz minimum w nieposortowanym podciągu (znak z minimalnym kodem ASCII) */
        i = str_chmin(str+k);
        i = k + i;

        /* zamien znaki miejscami */
        tmpch = str[i];
        str[i] = str[k];
        str[k] = tmpch;
    }

    return str; // * funkcja zwraca adres łańcucha znaków, na którym pracowała */
}

/* znajdz najmniejszy alfabetycznie znak (najmniejszy kod ASCII) w lancuchu znaków
funkcja zwraca indeks litery o minimalnym kodzie
-1 w przypadku ciągu pustego
funkcja realizuje wyszukiwanie liniowe wg sieci działań z wykładu
z tym, że nie zna liczby znaków i warunkiem końca pętli jest znalezienie kodu znaku = 0
*/
int str_chmin(char *str)
```

```
{
    int k,          /* indeks kolejnego znaku */
        i;         /* indeks znaku najmniejszego - minimum */
    char min;      /* minimum (najmniejszy kod ASCII znaku) */

    if (str[0]==0) return -1;

    for(i=0, k=1, min=str[0]; /* przyjmij, że najmniejszym znakiem jest pierwszy element */
        str[k]!=0;          /* wykonuj dotąd, aż nie napotkasz znacznika końca (kod ASCII = 0) */
        k++)
    {
        /* porównaj wielkość kodu ASCII kolejnej litery z przyjętym minimum */
        if (str[k]<min)
        { /* nowe minimum */
            i = k;
            min = str[k];
        }
    }

    return i; /* pozycja minimum */
}

/* można znaleźć liczbę znaków w ciągu za pomocą funkcji (własnej lub bibliotecznej)

    len = strlen(str); // długość łańcucha

    i zrealizować pętlę z licznikiem liczby elementów w łańcuchu znaków

*/
```