

```

// PRZYKŁAD p_30.cpp
// Przykładowa implementacja "listy studentów"
// z porządkowaniem według różnych kryteriów
// zaimplementowanym poprzez "fizyczne" sortowanie danych
// oraz poprzez sortowanie "logiczne" za pomocą tablic indeksowych
//
// Implementacja w/w zadania bez wprowadzenia abstrakcyjnego interfejsu
// do wykorzystywanych modeli danych: studenta, listy studentów

```

```

struct Student
{
    char nazwisko[30];
    int wiek;
    float ocena;
};

#define MAX_ROZMIAR 3
Student lista_studentow[MAX_ROZMIAR];

int indeksy_wg_wieku[MAX_ROZMIAR];
int indeksy_wg_ocen[MAX_ROZMIAR];

#include <iostream.h>
#include <string.h> //strcmp()
#include <conio.h> //clrscr(), getch()

void main()
{
    int i,j;
    clrscr();
    //wczytanie danych do listy_studentow
    for(i=0; i<MAX_ROZMIAR; i++)
    {
        cout<<"\n\nWczytywanie danych studenta nr: "<<i;
        cout<<"\nPodaj nazwisko: "; cin>>lista_studentow[i].nazwisko;
        cout<<"\nPodaj wiek: "; cin>>lista_studentow[i].wiek;
        cout<<"\nPodaj ocene: "; cin>>lista_studentow[i].ocena;
    }

    //prymitywne 'babelkowe' sortowanie listy w/g nazwisk
    for(j=1; j<MAX_ROZMIAR; j++)
        for(i=1; i<MAX_ROZMIAR; i++)
            if( strcmp(lista_studentow[i-1].nazwisko,
                lista_studentow[i].nazwisko) > 0 )
            {
                Student bufor;
                bufor = lista_studentow[i];
                lista_studentow[i] = lista_studentow[i-1];
                lista_studentow[i-1] = bufor;
            }
}

```

```

//inicjacja tablicy indeksów w/g wieku i sortowanie jej zawartosci
for(i=0; i<MAX_ROZMIAR; i++)
    indeksy_wg_wieku[i]=i;

for(j=1; j<MAX_ROZMIAR; j++)
    for(i=1; i<MAX_ROZMIAR; i++)
    {
        int pierwszy = indeksy_wg_wieku[i-1];
        int drugi    = indeksy_wg_wieku[i];

        if( lista_studentow[pierwszy].wiek > lista_studentow[drugi].wiek )
        {
            int bufor;
            bufor = indeksy_wg_wieku[i];
            indeksy_wg_wieku[i] = indeksy_wg_wieku[i-1];
            indeksy_wg_wieku[i-1] = bufor;
        }
    }

//inicjacja tablicy indeksów w/g ocen i sortowanie jej zawartosci
for(i=0; i<MAX_ROZMIAR; i++)
    indeksy_wg_ocen[i]=i;

for(j=1; j<MAX_ROZMIAR; j++)
    for(i=1; i<MAX_ROZMIAR; i++)
    {
        int pierwszy = indeksy_wg_ocen[i-1];
        int drugi    = indeksy_wg_ocen[i];

        if( lista_studentow[pierwszy].ocena > lista_studentow[drugi].ocena )
        {
            int bufor;
            bufor = indeksy_wg_ocen[i];
            indeksy_wg_ocen[i] = indeksy_wg_ocen[i-1];
            indeksy_wg_ocen[i-1] = bufor;
        }
    }

// dalszy ciag tej funkcji jest na nastepnej stronie

```

```

//wyswietlanie listy studentów posortowanej w/g dowolnego kryterium
int klawisz;
do
{
    cout<<"\n\nMENU:";
    cout<<"\n n - Wyswietlanie w/g nazwisk";
    cout<<"\n w - Wyswietlanie w/g wieku";
    cout<<"\n o - Wyswietlanie w/g ocen";
    cout<<"\n k - Koniec programu";
    cout<<"\n\nWybierz opcje: ";
    klawisz=getch();
    switch(klawisz)
    {
    case 'n': //Wyswietlanie w/g porzadku "fizycznego" tzn. w/g nazwisk
        cout<<"\n\nLista studentow w/g nazwisk:";
        for(i=0; i<MAX_ROZMIAR; i++)
        {
            cout<<endl<<lista_studentow[i].nazwisko<<" , ";
            cout<<lista_studentow[i].wiek<<" , ";
            cout<<lista_studentow[i].ocena;
        }
        break;

    case 'w': //Wyswietlanie w/g pierwszej tablicy indeksow
        cout<<"\n\nLista studentow w/g wieku:";
        for(i=0; i<MAX_ROZMIAR; i++)
        {
            j = indeksy_wg_wieku[i];

            cout<<endl<<lista_studentow[j].nazwisko<<" , ";
            cout<<lista_studentow[j].wiek<<" , ";
            cout<<lista_studentow[j].ocena;
        }
        break;

    case 'o': //Wyswietlanie w/g pierwszej drugiej indeksow
        cout<<"\n\nLista studentow w/g ocen:";
        for(i=0; i<MAX_ROZMIAR; i++)
        {
            j = indeksy_wg_ocen[i];

            cout<<endl<<lista_studentow[j].nazwisko<<" , ";
            cout<<lista_studentow[j].wiek<<" , ";
            cout<<lista_studentow[j].ocena;
        }
        break;
    }
}
while(klawisz!='k');
}

```

```

// PRZYKŁAD p_31.cpp
// Bardziej zaawansowana implementacja "listy studentów"
// z porzadkowaniem według różnych kryteriów
// zaimplementowanym poprzez "fizyczne" sortowanie danych
// oraz poprzez sortowanie "logiczne" za pomocą tablic indeksowych
//
// Ta implementacja wykorzystuje "obiektywne" podejście
// do wykorzystywanych modeli danych: studenta, listy studentów
// z wyróżnieniem abstrakcyjnego interfejsu

```

```

#include <iostream.h>
#include <string.h>      //strcmp()
#include <conio.h>       //clrscr(), getch()

// Definicja modelu danej typu "Student"
// przechowującej informacje o studencie: nazwisko, wiek, ocena
// z operacjami WCZYTAJ i WYSWIETL

struct Student
{
    // dane studenta
    char  nazwisko[30];
    int   wiek;
    float ocena;

    // podstawowe operacje na danej typu Student

    void  WCZYTAJ()
    {
        cout<<"\nPodaj nazwisko: "; cin>>nazwisko;
        cout<<"\nPodaj wiek: ";      cin>>wiek;
        cout<<"\nPodaj ocene: ";    cin>>ocena;
    }

    void  WYSWIETL()
    {
        cout<<endl<<nazwisko<<" , ";
        cout<<wiek<<" , ";
        cout<<ocena;
    }
}; //-----struct Student

```

```

// Definicja modelu danej typu "Lista_Studentow"
// przechowujacej informacje o liscie danych typu "Student"
// z podstawowymi operacjami:
//     INICJUIJ,
//     WCZYTAJ, WYSWIETL
//

// operacjami sterujacymi porzadkiem wyswietlania:
//     SORTUJ, USTAW_INDEKS
//
// i pomocniczymi operacjami sortowania:
//     ZAMIEN(student1,student2)
//     ZAMIEN(indeks1,indeks2)

struct Lista_Studentow
{
    //reprezentacja danych listy studentów
    #define MAX_ROZMIAR 3
    Student lista_studentow[MAX_ROZMIAR];
    int indeksy_wg_wieku[MAX_ROZMIAR];
    int indeksy_wg_ocen[MAX_ROZMIAR];
    int *aktywny_indeks;

    //podstawowe operacje
    void INICJUIJ(void)
        { aktywny_indeks=NULL; }

    void USTAW_INDEKS(int *indeks)
        { aktywny_indeks = indeks; }

    void WCZYTAJ(void)
        {
            cout<<"\n\nWczytywanie listy studentow";
            for(int i=0; i<MAX_ROZMIAR; i++)
                {
                    cout<<"\n\nWczytywanie danych studenta nr: "<<i;
                    lista_studentow[i].WCZYTAJ();
                }
        }

    void WYSWIETL(void)
        {
            if( aktywny_indeks==NULL )
                {
                    for(int i=0; i<MAX_ROZMIAR; i++)
                        lista_studentow[i].WYSWIETL();
                }
            else
                {
                    for(int j,i=0; i<MAX_ROZMIAR; i++)
                        {
                            j = aktywny_indeks[i];
                            lista_studentow[j].WYSWIETL();
                        }
                }
        }
}

```

```

void ZAMIEN( Student &s1, Student &s2)
{
    Student bufor;
    bufor = s1;
    s1 = s2;
    s2 = bufor;
}

void ZAMIEN( int &i1, int &i2)
{
    int bufor;
    bufor = i1;
    i1 = i2;
    i2 = bufor;
}

void SORTUJ( void )
{
    int i,j;
    // poczatkowa inicjacja tablic indeksów
    for(i=0;i<MAX_ROZMIAR; i++)
    {
        indeksy_wg_wieku[i]=i;
        indeksy_wg_ocen[i]=i;
    }

    // "fizyczne" sortowanie w/g nazwisk
    for(j=1;j<MAX_ROZMIAR; j++)
        for(i=1;i<MAX_ROZMIAR; i++)
        {
            if( strcmp(lista_studentow[i-1].nazwisko,
                       lista_studentow[i].nazwisko) > 0 )
                ZAMIEN( lista_studentow[i-1], lista_studentow[i] );
        }

    // "indeksowe" sortowanie w/g wieku i ocen
    for(j=1;j<MAX_ROZMIAR; j++)
        for(i=1;i<MAX_ROZMIAR; i++)
        {
            int pierwszy = indeksy_wg_wieku[i-1];
            int drugi    = indeksy_wg_wieku[i];
            if( lista_studentow[pierwszy].wiek >
                lista_studentow[drugi].wiek )
                ZAMIEN( indeksy_wg_wieku[i-1], indeksy_wg_wieku[i]);

            pierwszy = indeksy_wg_ocen[i-1];
            drugi    = indeksy_wg_ocen[i];
            if( lista_studentow[pierwszy].ocena >
                lista_studentow[drugi].ocena )
                ZAMIEN( indeksy_wg_ocen[i-1], indeksy_wg_ocen[i]);
        }
    }
};//-----struct Lista_Studentow

```

```

void main()
{
Lista_Studentow lista_inf;    // Lista_Studentow  inf_2, eit;

clrscr();
lista_inf.INICJUIJ();
lista_inf.WCZYTAJ();
lista_inf.SORTUIJ();

//wyswietlanie listy studentów posortowanej w/g dowolnego kryterium
int klawisz;
do
{
    cout<<"\n\nMENU:";
    cout<<"\n n - Wyswietlanie w/g nazwisk";
    cout<<"\n w - Wyswietlanie w/g wieku";
    cout<<"\n o - Wyswietlanie w/g ocen";
    cout<<"\n k - Koniec programu";
    cout<<"\n\nWybierz opcje: ";
    klawisz=getch();
    switch(klawisz)
    {
        case 'n': //Wyswietlanie w/g porzadku "fizycznego" tzn. w/g nazwisk
            cout<<"\n\nLista studentow w/g nazwisk:";

            lista_inf.USTAW_INDEKS(NULL);
            lista_inf.WYSWIETL();

            break;

        case 'w': //Wyswietlanie w/g pierwszej tablicy indeksow
            cout<<"\n\nLista studentow w/g wieku:";

            lista_inf.USTAW_INDEKS(lista_inf.indeksy_wg_wieku);
            lista_inf.WYSWIETL();

            break;

        case 'o': //Wyswietlanie w/g pierwszej drugiej indeksow
            cout<<"\n\nLista studentow w/g ocen:";

            lista_inf.USTAW_INDEKS(lista_inf.indeksy_wg_ocen);
            lista_inf.WYSWIETL();

            break;
    }
}
while(klawisz!='k');
}

```