

**ANS w Elblągu**  
**Instytut Informatyki Stosowanej im. Krzysztofa**  
**Brzeskiego**

**Podstawy programowania II SN –**  
**laboratorium**

**Studium niestacjonarne, sem. 2, 2023/2024**

**Projekt minibazy danych,**  
**niedziela, gr.1, 12:15-13:22**

**Data oddania: 16.06.2024**

**Nazwisko i imię: Sparzak Julia**

**Nr albumu: 21198**

**Nazwa pliku : pr1Kod21198Sparzak2024.pdf**

```

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

int bazaOtwarta = 0;
char nazwaOtwartejBazy[12];

typedef char* string;

struct probka
{
    char nazwa[10];
    char typ[10];
    char dataPoboru[11];
    char region[31];
    char wataha[30];
    double x;
    double y;
    char kto[20];
};

void menu(void);
struct probka wczytStruct(void);
void drukStruct(struct probka st);
void sortujBaze(void);
void przegladBazy(void);
void modStruct(int pos);
void usunStruct(int pos);
void utworzBaze(void);
void otworzBaze(void);
int sprawdzNazweBazy(char *nazwa);
void dopiszStruct(void);
void przegladMenu(int pos);
void wyswietlStrukture(int pos);
void home(void);
void end(void);
void next(int *pos);
void previous(int *pos);
int liczbaStruktur(void);
void gotoxy(short x, short y, int columns, int rows);
void drukujWszystkieStruktury(void);

int main(void)
{
    menu();
    return 0;
}

void menu(void)
{
    char ch;
    int pos = 1;

```

```

FILE *fp;

do
{

    system("cls");

    CONSOLE_SCREEN_BUFFER_INFO csbi;
    int columns, rows;
    GetConsoleScreenBufferInfo(GetStdHandle(STD_OUTPUT_HANDLE), &csbi);
    columns = csbi.srWindow.Right - csbi.srWindow.Left + 1;
    rows = csbi.srWindow.Bottom - csbi.srWindow.Top + 1;

    gotoxy(0, 0, columns, rows); printf("=====\n");
    gotoxy(0, 1, columns, rows); printf(" 1. Otworz baze danych\n");
    gotoxy(0, 2, columns, rows); printf(" 2. Utworz nowa baze\n");
    gotoxy(0, 3, columns, rows); printf(" 3. Przegląd bazy\n");
    gotoxy(0, 4, columns, rows); printf(" 4. Sortowanie bazy\n");
    gotoxy(0, 5, columns, rows); printf(" 5. Modyfikacja struktury\n");
    gotoxy(0, 6, columns, rows); printf(" 6. Usun plik bazy danych\n");
    gotoxy(0, 7, columns, rows); printf(" 7. Zakoncz program\n");
    gotoxy(0, 8, columns, rows); printf("=====\n");
    gotoxy(0, 9, columns, rows); printf(" Wybierz opcje :");
    ch = getch();

    switch(ch)
    {
    case '1':
        otworzBaze();
        przegladMenu(pos);
        break;
    case '2':
        utworzBaze();
        przegladMenu(pos);
        break;
    case '3':
        przegladBazy();
        break;
    case '4':
        sortujBaze();
        break;
    case '5':
    {
        int pos;
        printf("\n Podaj pozycje struktury do modyfikacji: ");
        scanf("%d", &pos);
        modStruct(pos);
        break;
    }
    case '6':
    {
        char odpowiedz;

```

```

        printf("\n Czy na pewno chcesz usunac baze danych '%s'? (T/N): ",
nazwaOtwartejBazy);
        odpowiedz = getch();
        if (odpowiedz == 'T' || odpowiedz == 't')
        {
            remove(nazwaOtwartejBazy);
            printf("\n Baza danych '%s' zostala usunieta.", nazwaOtwartejBazy);
            bazaOtwarta = 0;
        }
        else
        {
            printf("\n Operacja usuniecia bazy danych anulowana.");
        }
        break;
    }
    case '7':
    {
        char odpowiedz;
        printf("\n Czy chcesz zapisac aktualny stan bazy danych przed wyjsciem? (T/N): ");
        odpowiedz = getch();
        if (odpowiedz == 'T' || odpowiedz == 't')
        {
            if (bazaOtwarta)
            {
                printf("\n Zapisywanie aktualnego stanu bazy danych...");
                fclose(fp);
                printf("\n Aktualny stan bazy danych zostal zapisany.");
            }
            else
            {
                printf("\n Nie ma otwartej bazy danych do zapisania.");
            }
        }
        else
        {
            printf("\n Stan bazy danych nie zostanie zapisany.");
        }
        printf("\n Zamykanie programu...");
        exit(0);
    }
}
while (ch != '7');
}

```

```

void gotoxy(short x, short y, int columns, int rows)
{
    // Calculate the starting position to center the menu
    int start_x = (columns - 30) / 2; // Adjust width as necessary
    int start_y = (rows - 10) / 2 + y; // Adjust height as necessary

    COORD pos = {start_x + x, start_y};
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), pos);
}

```

```

struct probka wczytStruct(void)
{
    struct probka st;
    printf("\nNazwa:");
    scanf("%9s", st.nazwa);
    printf("\nTyp:");
    scanf("%9s", st.typ);
    printf("\nData poboru:");
    scanf("%10s", st.dataPoboru);
    printf("\nRegion:");
    scanf("%30s", st.region);
    printf("\nWataha:");
    scanf("%29s", st.wataha);
    printf("\nKoordynaty x:");
    scanf("%lf", &st.x);
    printf("\nKoordynaty y:");
    scanf("%lf", &st.y);
    printf("\nOsoba, ktora pobrala probke:");
    scanf("%19s", st.kto);
    return st;
}

void drukStruct(struct probka st)
{
    printf("\n Nazwa: %s", st.nazwa);
    printf("\n Typ: %s", st.typ);
    printf("\n Data poboru: %s", st.dataPoboru);
    printf("\n Region: %s", st.region);
    printf("\n Wataha: %s", st.wataha);
    printf("\n Koordynaty x: %lf", st.x);
    printf("\n Koordynaty y: %lf", st.y);
    printf("\n Osoba, ktora pobrala probke: %s", st.kto);
}

```

```

int sprawdzNazweBazy(char *nazwa)
{
    char lan1[4], lan2[3], lan3[5];
    int wynik1, wynik2, wynik3;

    strncpy(lan1, nazwa, 4);
    lan1[4] = '\0';
    strncpy(lan2, nazwa + 4, 2);
    lan2[2] = '\0';
    strcpy(lan3, nazwa + 6);

    wynik1 = strcmp(lan1, "baza");
    wynik2 = isdigit(lan2[0]) && isdigit(lan2[1]);
    wynik3 = strcmp(lan3, ".dat");
}

```

```

    if (!wynik1 && wynik2 && !wynik3)
    {
        printf("\n Nazwa bazy jest zgodna z wymogami");
        return 1;
    }
    else
    {
        printf("\n Nazwa bazy jest niezgodna z wymogami");
        return 0;
    }
}

void otworzBaze(void)
{
    FILE *fp;
    char nazwa[12];
    char odpowiedz;

    if (bazaOtwarta)
    {
        printf("\n Plik minibazy %s jest otwarty. Czy zamknac plik T/N?", nazwaOtwartejBazy);
        odpowiedz = getch();
        if (odpowiedz == 'T' || odpowiedz == 't')
        {
            bazaOtwarta = 0;
        }
        else
        {
            printf("\n Powrot do menu glownego.");
            return;
        }
    }

    printf("\n Podaj nazwe minibazy do otwarcia: ");
    scanf("%11s", nazwa);

    if (!sprawdzNazweBazy(nazwa))
    {
        printf("\n Nazwa bazy niezgodna z wymogami.");
        return;
    }

    fp = fopen(nazwa, "rb");
    if (fp == NULL)
    {
        printf("\n Plik %s nie zostal otwarty", nazwa);
        return;
    }
    else
    {
        printf("\n Plik %s zostal otwarty", nazwa);
        strcpy(nazwaOtwartejBazy, nazwa);
        fclose(fp);
        bazaOtwarta = 1;
    }
}

```

```
}
```

```
void sortujBaze(void)
```

```
{
```

```
    FILE *fp;
```

```
    struct probka bufs1, bufs2;
```

```
    int n = liczbaStruktur(), zam, i;
```

```
    int wybor;
```

```
    if (n <= 1)
```

```
    {
```

```
        printf("\n Brak wystarczajacej liczby struktur do sortowania");
```

```
        return;
```

```
    }
```

```
    fp = fopen(nazwaOtwartejBazy, "r+b");
```

```
    if (fp == NULL)
```

```
    {
```

```
        printf("\n Plik %s nie zostal otwarty", nazwaOtwartejBazy);
```

```
        exit(1);
```

```
    }
```

```
    printf("\n Wybierz pole, wedlug ktorego chcesz posortowac struktury:");
```

```
    printf("\n 1. Nazwa");
```

```
    printf("\n 2. Typ");
```

```
    printf("\n 3. Data poboru");
```

```
    printf("\n 4. Region");
```

```
    printf("\n 5. Wataha");
```

```
    printf("\n 6. Koordynaty x");
```

```
    printf("\n 7. Koordynaty y");
```

```
    printf("\n 8. Kto\n");
```

```
    printf(" Wybierz opcje :");
```

```
    scanf("%d", &wybor);
```

```
    do
```

```
    {
```

```
        zam = 0;
```

```
        for (i = 0; i < n - 1; i++)
```

```
        {
```

```
            fseek(fp, i * sizeof(struct probka), SEEK_SET);
```

```
            fread(&bufs1, sizeof(struct probka), 1, fp);
```

```
            fread(&bufs2, sizeof(struct probka), 1, fp);
```

```
            switch(wybor) {
```

```
                case 1:
```

```
                    if (strcmp(bufs1.nazwa, bufs2.nazwa) > 0)
```

```
                        zam = 1;
```

```
                    break;
```

```
                case 2:
```

```
                    if (strcmp(bufs1.typ, bufs2.typ) > 0)
```

```
                        zam = 1;
```

```
                    break;
```

```
                case 3:
```

```

        if (strcmp(bufs1.dataPoboru, bufs2.dataPoboru) > 0)
            zam = 1;
        break;
    case 4:
        if (strcmp(bufs1.region, bufs2.region) > 0)
            zam = 1;
        break;
    case 5:
        if (strcmp(bufs1.wataha, bufs2.wataha) > 0)
            zam = 1;
        break;
    case 6:
        if (bufs1.x > bufs2.x)
            zam = 1;
        break;
    case 7:
        if (bufs1.y > bufs2.y)
            zam = 1;
        break;
    case 8:
        if (strcmp(bufs1.kto, bufs2.kto) > 0)
            zam = 1;
        break;
    default:
        printf("\n Niepoprawny wybor");
        fclose(fp);
        return;
    }

    if (zam) {
        fseek(fp, i * sizeof(struct probka), SEEK_SET);
        fwrite(&bufs2, sizeof(struct probka), 1, fp);
        fwrite(&bufs1, sizeof(struct probka), 1, fp);
    }
}
n--;
}
while (zam);

fclose(fp);
printf("\n Plik %s zostal posortowany", nazwaOtwartejBazy);
}

int liczbaStruktur(void)
{
    FILE *fp;
    int count = 0;
    struct probka bufs;

    fp = fopen(nazwaOtwartejBazy, "rb");
    if (fp == NULL)
    {
        printf("\n Plik %s nie zostal otwarty", nazwaOtwartejBazy);
        exit(1);
    }
}

```



```

while (fread(&bufs, sizeof(struct probka), 1, fp))
{
    count++;
}

fclose(fp);
return count;
}

void przegladBazy(void)
{
    przegladMenu(1);
}

void przegladMenu(int pos)
{
    char ch;
    do
    {
        printf(" \n\n=====");
        printf("\nB(lub strzalka w dol)- przesuniecie o jedna strukture w dol\n");
        printf("Y(lub strzalka w gore)- przesuniecie o jedna strukture w gore\n");
        printf("G (lub HOME)- przesuniecie na poczatek\n");
        printf("H (lub END)- przesuniecie na koniec\n");
        printf("D- dopisz nowa strukture\n");
        printf("U- usun biezaca strukture\n");
        printf("M- modyfikuj strukture\n");
        printf("S- sortuj baze danych\n");
        printf("A- drukuj wszystkie struktury\n");
        printf("esc- powrot do menu\n");
        printf("=====\\n");

        ch = getch();
        switch(ch)
        {
            case 80:
            case 'b':
                next(&pos);
                break;
            case 72:
            case 'y':
                previous(&pos);
                break;
            case 71:
            case 'g':
                home();
                pos = 1;
                break;
            case 79:
            case 'h':
                end();
                pos = liczbaStruktur();
                break;
            case 'D':

```

```

        case 'd':
            dopiszStruct();
            break;
        case 'U':
        case 'u':
            usunStruct(pos);
            break;
        case 'M':
        case 'm':
            modStruct(pos);
            break;
        case 'S':
        case 's':
            sortujBaze();
            break;
        case 'A':
        case 'a':
            drukujWszystkieStruktury();
            break;
        case 27:
            menu();
            return;
    }
    wyswietlStrukture(pos);
}
while (ch != 27);
}

void wyswietlStrukture(int pos)
{
    FILE *fp;
    struct probka buf;

    fp = fopen(nazwaOtwartejBazy, "rb");
    if (fp == NULL)
    {
        printf("\n Plik %s nie zostal otwarty", nazwaOtwartejBazy);
        exit(1);
    }

    fseek(fp, (pos - 1) * sizeof(struct probka), SEEK_SET);
    if (fread(&buf, sizeof(struct probka), 1, fp) == 1)
    {
        system("cls");
        printf("\n Struktura %d", pos);
        drukStruct(buf);
    }
    else
    {
        printf("\n Brak struktury na pozycji %d", pos);
    }

    fclose(fp);
}

```

```

void drukujWszystkieStruktury(void)
{
    FILE *fp;
    struct probka buf;
    int pos = 1;

    fp = fopen(nazwaOtwartejBazy, "rb");
    if (fp == NULL)
    {
        printf("\n Plik %s nie zostal otwarty", nazwaOtwartejBazy);
        exit(1);
    }

    while (fread(&buf, sizeof(struct probka), 1, fp))
    {
        printf("\n\n Struktura %d", pos++);
        drukStruct(buf);
    }

    fclose(fp);

    printf("\nNacisnij dowolny klawisz, aby wrocic do menu...");
    getch();

    menu();
}

void home(void)
{
    wyswietlStruktura(1);
}

void end(void)
{
    wyswietlStruktura(liczbaStruktur());
}

void next(int *pos)
{
    if (*pos < liczbaStruktur())
    {
        (*pos)++;
        wyswietlStruktura(*pos);
    }
}

void previous(int *pos)
{
    if (*pos > 1)
    {
        (*pos)--;
        wyswietlStruktura(*pos);
    }
}

```

```

void dopiszStruct(void)
{
    FILE *fp;
    struct probka nowa = wczytStruct();

    fp = fopen(nazwaOtwartejBazy, "ab");
    if (fp == NULL)
    {
        printf("\n Plik %s nie zostal otwarty", nazwaOtwartejBazy);
        exit(1);
    }

    fwrite(&nowa, sizeof(struct probka), 1, fp);
    fclose(fp);
    printf("\n Nowa struktura zostala dodana");
}

void modStruct(int pos)
{
    FILE *fp;
    struct probka nowa;

    fp = fopen(nazwaOtwartejBazy, "r+b");
    if (fp == NULL)
    {
        printf("\n Plik %s nie zostal otwarty", nazwaOtwartejBazy);
        exit(1);
    }

    fseek(fp, (pos - 1) * sizeof(struct probka), SEEK_SET);
    if (fread(&nowa, sizeof(struct probka), 1, fp) != 1)
    {
        printf("\n Brak struktury na pozycji %d", pos);
        fclose(fp);
        return;
    }

    printf("\n Biezaca struktura na pozycji %d:", pos);
    drukStruct(nowa);

    printf("\n\n Podaj nowe wartosci dla struktury na pozycji %d:\n", pos);
    nowa = wczytStruct();

    fseek(fp, (pos - 1) * sizeof(struct probka), SEEK_SET);
    fwrite(&nowa, sizeof(struct probka), 1, fp);
    fclose(fp);

    printf("\n Struktura na pozycji %d zostala zmodyfikowana", pos);
}

void usunStruct(int pos)
{
    FILE *fp, *temp;
    struct probka bufs;
    int count = 0;

```

```

fp = fopen(nazwaOtwartejBazy, "rb");
if (fp == NULL)
{
    printf("\n Plik %s nie zostal otwarty", nazwaOtwartejBazy);
    exit(1);
}

temp = fopen("temp.bin", "wb");
if (temp == NULL)
{
    printf("\n Tymczasowy plik nie zostal utworzony");
    fclose(fp);
    exit(1);
}

while (fread(&bufs, sizeof(struct probka), 1, fp))
{
    count++;
    if (count != pos)
    {
        fwrite(&bufs, sizeof(struct probka), 1, temp);
    }
}

fclose(fp);
fclose(temp);

remove(nazwaOtwartejBazy);
rename("temp.bin", nazwaOtwartejBazy);

printf("\n Struktura na pozycji %d zostala usunieta", pos);
}

void utworzBaze(void)
{
    FILE *fp;
    char nazwa[12];
    char odpowiedz;

    if (bazaOtwarta)
    {
        printf("\n Plik minibazy %s jest otwarty. Czy zamknac plik T/N?", nazwaOtwartejBazy);
        odpowiedz = getch();
        if (odpowiedz == 'T' || odpowiedz == 't')
        {
            bazaOtwarta = 0;
        }
        else
        {
            printf("\n Powrot do menu glownego.");
            return;
        }
    }
}

```

```

printf("\n Wymagania co do nazwy: ");
printf("\n 1. Max 11 znakow dlugosci ");
printf("\n 2. format 'bazaXX.dat', gdzie 'XX' to dwie cyfry, a '.dat' to rozszerzenie pliku ");
printf("\n 3. Pierwsze cztery znaki musza byc 'baza' ");
printf("\n 4. Nastepne dwa znaki musza byc cyframi. ");
printf("\n 5. Ostatnie cztery znaki musza byc '.dat'\n");
printf("\n Podaj nazwe nowej minibazy: ");
scanf("%11s", nazwa);

if (!sprawdzNazweBazy(nazwa))
{
    printf("\n Niepoprawna nazwa bazy.");
    return;
}

strcpy(nazwaOtwartejBazy, nazwa);

fp = fopen(nazwaOtwartejBazy, "rb");
if (fp != NULL)
{
    fclose(fp);
    printf("\n Plik %s juz istnieje. Wybierz inna nazwe.", nazwaOtwartejBazy);
    return;
}

fp = fopen(nazwaOtwartejBazy, "w+b");
if (fp == NULL)
{
    printf("\n Plik %s nie zostal utworzony", nazwaOtwartejBazy);
    exit(1);
}

struct probka tablicaStruktur[] =
{
    {"BD234", "tkanka", "2014.02.10", "Bory Dolnoslaskie", "Zagan", 15.2579, 51.5585,
    "J.Karwanski"},
    {"LNR028", "odchody", "2016.02.18", "Lasy Napiwodzko-Ramuckie", "Nidzica/Jedwabno",
    19.82024, 53.74159, "M. Szewczyk"},
    {"PK008", "mocz", "2016.01.10", "Puszcza Kampinoska", "Kampinoska", 20.51367,
    52.36366, "M. Szewczyk"},
    {"WE27", "odchody", "2023.01.10", "Wysoczyzna Elblaska", "Kadyny", 20.48580,
    54.07302, "J.Sparzak"}
};

fwrite(tablicaStruktur, sizeof(struct probka), sizeof(tablicaStruktur) / sizeof(struct probka), fp);

fclose(fp);
printf("\n Nowy plik %s zostal utworzony", nazwaOtwartejBazy);
bazaOtwarta = 1;
}

```

Tworzenie pliku bazy danych:

```
=====
1. Otworz baze danych
2. Utworz nowa baze
3. Przegląd bazy
4. Sortowanie bazy
5. Modyfikacja struktury
6. Usun plik bazy danych
7. Zakoncz program
=====
Wybierz opcje :
Wymagania co do nazwy:
1. Max 11 znakow dlugosci
2. format 'bazaXX.dat', gdzie 'XX' to dwie cyfry, a '.dat' to rozszerzenie pliku
3. Pierwsze cztery znaki musza byc 'baza'
4. Nastepne dwa znaki musza byc cyframi.
5. Ostatnie cztery znaki musza byc '.dat'

Podaj nazwe nowej minibazy: baza01.dat

Nazwa bazy jest zgodna z wymogami
Nowy plik baza01.dat zostal utworzony
```

Wybranie opcji tworzenia nowej bazy w przypadku otwartego pliku:

```
=====
1. Otworz baze danych
2. Utworz nowa baze
3. Przegląd bazy
4. Sortowanie bazy
5. Modyfikacja struktury
6. Usun plik bazy danych
7. Zakoncz program
=====
Wybierz opcje :
Plik minibazy baza01.dat jest otwarty. Czy zamknac plik T/N?
```

Przegląd bazy:

```
Struktura 1
Nazwa: BD234
Typ: tkanka
Data poboru: 2014.02.10
Region: Bory Dolnoslaskie
Wataha: Zagan
Koordynaty x: 15.257900
Koordynaty y: 51.558500
Osoba, ktora pobrala probke: J.Karwanski

=====
B(lub strzałka w dol)- przesuniecie o jedna strukture w dol
Y(lub strzałka w gore)- przesuniecie o jedna strukture w gore
G (lub HOME)- przesuniecie na poczatek
H (lub END)- przesuniecie na koniec
D- dopisz nowa strukture
U- usun biezaca strukture
M- modyfikuj strukture
S- sortuj baze danych
A- drukuj wszystkie struktury
esc- powrot do menu
=====
```

Kliknij dwukrot

```
Struktura 2
Nazwa: LNR028
Typ: odchody
Data poboru: 2016.02.18
Region: Lasy Napiwodzko-Ramuckie
Wataha: Nidzica/Jedwabno
Koordynaty x: 19.820240
Koordynaty y: 53.741590
Osoba, która pobrała probke: M. Szewczyk
```

```
=====
B(lub strzałka w dol)- przesuniecie o jedna strukture w dol
Y(lub strzałka w gore)- przesuniecie o jedna strukture w gore
G (lub HOME)- przesuniecie na poczatek
H (lub END)- przesuniecie na koniec
D- dopisz nowa strukture
U- usun biezaca strukture
M- modyfikuj strukture
S- sortuj baze danych
A- drukuj wszystkie struktury
esc- powrot do menu
=====
```

Dodana opcja wydruku wszystkich struktur bazy:

```
C:\Users\sirre\OneDrive\Pulpi  X  +  v

Struktura 1
Nazwa: BD234
Typ: tkanka
Data poboru: 2014.02.10
Region: Bory Dolnoslaskie
Wataha: Zagan
Koordynaty x: 15.257900
Koordynaty y: 51.558500
Osoba, która pobrała probke: J.Karwanski

Struktura 2
Nazwa: LNR028
Typ: odchody
Data poboru: 2016.02.18
Region: Lasy Napiwodzko-Ramuckie
Wataha: Nidzica/Jedwabno
Koordynaty x: 19.820240
Koordynaty y: 53.741590
Osoba, która pobrała probke: M. Szewczyk

Struktura 3
Nazwa: PK008
Typ: mocz
Data poboru: 2016.01.10
Region: Puszcza Kampinoska
Wataha: Kampinoska
Koordynaty x: 20.513670
Koordynaty y: 52.363660
Osoba, która pobrała probke: M. Szewczyk

Struktura 4
Nazwa: WE27
Typ: odchody
Data poboru: 2023.01.10
Region: Wysoczyzna Elbl-ńska
Wataha: Kadyne
Koordynaty x: 20.485800
Koordynaty y: 54.073020
Osoba, która pobrała probke: J.Sparzak
Nacisnij dowolny klawisz, aby wrocic do menu...|
```



Dodawanie nowej struktury:

```
Struktura 5
Nazwa: AB169
Typ: siersc
Data poboru: 02.06.202
Region: 4
Wataha: Elblag
Koordynaty x: 20.123450
Koordynaty y: 50.123450
Osoba, ktora pobrala probke: J.Sparzak

=====
B(lub strzałka w dol)- przesuniecie o jedna strukture w dol
Y(lub strzałka w gore)- przesuniecie o jedna strukture w gore
G (lub HOME)- przesuniecie na poczatek
H (lub END)- przesuniecie na koniec
D- dopisz nowa strukture
U- usun biezaca strukture
M- modyfikuj strukture
S- sortuj baze danych
A- drukuj wszystkie struktury
esc- powrot do menu
=====
```

Modyfikacja struktury:

```
Struktura 5
Nazwa: HV2137
Typ: siersc
Data poboru: 02.02.2024
Region: Wysocyzna
Wataha: Elblaska
Koordynaty x: 20.123400
Koordynaty y: 50.098600
Osoba, ktora pobrala probke: J.Sparzak

=====
B(lub strzałka w dol)- przesuniecie o jedna strukture w dol
Y(lub strzałka w gore)- przesuniecie o jedna strukture w gore
G (lub HOME)- przesuniecie na poczatek
H (lub END)- przesuniecie na koniec
D- dopisz nowa strukture
U- usun biezaca strukture
M- modyfikuj strukture
S- sortuj baze danych
A- drukuj wszystkie struktury
esc- powrot do menu
=====
```

Potwierdzenie wprowadzenia zmian przez druk wszystkich struktur:

```
C:\Users\sirre\OneDrive\Pulpi  X + v
Osoba, która pobrała probkę: J.Karwanski

Struktura 2
Nazwa: LNR028
Typ: odchody
Data poboru: 2016.02.18
Region: Lasy Napiwodzko-Ramuckie
Wataha: Nidzica/Jedwabno
Koordynaty x: 19.820240
Koordynaty y: 53.741590
Osoba, która pobrała probkę: M. Szewczyk

Struktura 3
Nazwa: PK008
Typ: mocz
Data poboru: 2016.01.10
Region: Puszcza Kampinowska
Wataha: Kampinowska
Koordynaty x: 20.513670
Koordynaty y: 52.363660
Osoba, która pobrała probkę: M. Szewczyk

Struktura 4
Nazwa: WE27
Typ: odchody
Data poboru: 2023.01.10
Region: Wysoczyzna Elbl-ńska
Wataha: Kadyny
Koordynaty x: 20.485800
Koordynaty y: 54.073020
Osoba, która pobrała probkę: J.Sparzak

Struktura 5
Nazwa: HV2137
Typ: siersc
Data poboru: 02.02.2024
Region: Wysoczyzna
Wataha: Elblaska
Koordynaty x: 20.123400
Koordynaty y: 50.098600
Osoba, która pobrała probkę: J.Sparzak
```

Usuwanie struktury:

```
C:\Users\sirre\OneDrive\Pulpi X + v

Struktura 5
Nazwa: HV2137
Typ: sierśc
Data poboru: 02.02.2024
Region: Wysoczyzna
Wataha: Elblaska
Koordynaty x: 20.123400
Koordynaty y: 50.098600
Osoba, która pobrała probke: J.Sparzak

=====
B(lub strzałka w dol)- przesuniecie o jedna strukture w dol
Y(lub strzałka w gore)- przesuniecie o jedna strukture w gore
G (lub HOME)- przesuniecie na poczatek
H (lub END)- przesuniecie na koniec
D- dopisz nowa strukture
U- usun biezaca strukture
M- modyfikuj strukture
S- sortuj baze danych
A- drukuj wszystkie struktury
esc- powrot do menu
=====

Struktura na pozycji 5 zostala usunieta
Brak struktury na pozycji 5

=====
B(lub strzałka w dol)- przesuniecie o jedna strukture w dol
Y(lub strzałka w gore)- przesuniecie o jedna strukture w gore
G (lub HOME)- przesuniecie na poczatek
H (lub END)- przesuniecie na koniec
D- dopisz nowa strukture
U- usun biezaca strukture
M- modyfikuj strukture
S- sortuj baze danych
A- drukuj wszystkie struktury
esc- powrot do menu
=====
```

Ponowny druk wszystkich struktur:

```
C:\Users\sirre\OneDrive\Pulpi X + v

Struktura 1
Nazwa: BD234
Typ: tkanka
Data poboru: 2014.02.10
Region: Bory Dolnoslaskie
Wataha: Zagan
Koordynaty x: 15.257900
Koordynaty y: 51.558500
Osoba, która pobrała probke: J.Karwanski

Struktura 2
Nazwa: LNR028
Typ: odchody
Data poboru: 2016.02.18
Region: Lasy Napiwodzko-Ramuckie
Wataha: Nidzica/Jedwabno
Koordynaty x: 19.820240
Koordynaty y: 53.741590
Osoba, która pobrała probke: M. Szewczyk

Struktura 3
Nazwa: PK008
Typ: mocz
Data poboru: 2016.01.10
Region: Puszcza Kampinoska
Wataha: Kampinoska
Koordynaty x: 20.513670
Koordynaty y: 52.363660
Osoba, która pobrała probke: M. Szewczyk

Struktura 4
Nazwa: WE27
Typ: odchody
Data poboru: 2023.01.10
Region: Wysoczyzna Elbl-ńska
Wataha: Kadyny
Koordynaty x: 20.485800
Koordynaty y: 54.073020
Osoba, która pobrała probke: J.Sparzak
Nacisnij dowolny klawisz, aby wrocic do menu...|
```

W celach prezentacyjnych dodano struktury wypełnione rekordami 1, 2 i 3, aby ukazać działanie sortowania.

## Sortowanie:

```
C:\Users\sirre\OneDrive\Pulpi  x + v

Struktura 1
Nazwa: BD234
Typ: tkanka
Data poboru: 2014.02.10
Region: Bory Dolnoslaskie
Wataha: Zagan
Koordynaty x: 15.257900
Koordynaty y: 51.558500
Osoba, ktora pobrala probke: J.Karwanski

=====
B(lub strzałka w dol)- przesuniecie o jedna strukture w dol
Y(lub strzałka w gore)- przesuniecie o jedna strukture w gore
G (lub HOME)- przesuniecie na poczatek
H (lub END)- przesuniecie na koniec
D- dopisz nowa strukture
U- usun biezaca strukture
M- modyfikuj strukture
S- sortuj baze danych
A- drukuj wszystkie struktury
esc- powrot do menu
=====

Wybierz pole, wedlug ktorego chcesz posortowac struktury:
1. Nazwa
2. Typ
3. Data poboru
4. Region
5. Wataha
6. Koordynaty x
7. Koordynaty y
8. Kto
Wybierz opcje :|
```

Wybór opcji 5 oraz wydruk wszystkich rekordów (opcja A):

```
Struktura 1
Nazwa: 1
Typ: 1
Data poboru: 1
Region: 1
Wataha: 1
Koordynaty x: 1.000000
Koordynaty y: 1.000000
Osoba, ktora pobrala probke: 1

Struktura 2
Nazwa: 2
Typ: 2
Data poboru: 2
Region: 2
Wataha: 2
Koordynaty x: 2.000000
Koordynaty y: 2.000000
Osoba, ktora pobrala probke: 2

Struktura 3
Nazwa: 3
Typ: 3
Data poboru: 3
Region: 3
Wataha: 3
Koordynaty x: 3.000000
Koordynaty y: 3.000000
Osoba, ktora pobrala probke: 3

Struktura 4
Nazwa: WE27
Typ: odchody
Data poboru: 2023.01.10
Region: Wysocyzna Elbl-Źska
Wataha: Kadyny
Koordynaty x: 20.485800
Koordynaty y: 54.073020
Osoba, ktora pobrala probke: J.Sparzak

Struktura 5
Nazwa: PK008
Typ: mocz
Data poboru: 2016.01.10
Region: Puszcza Kampynoska
Wataha: Kampynoska
Koordynaty x: 20.513670
Koordynaty y: 52.363660
Osoba, ktora pobrala probke: M. Szewczyk

Struktura 6
Nazwa: LNR028
Typ: odchody
Data poboru: 2016.02.18
Region: Lasy Napiwodzko-Ramuckie
Wataha: Midzica/Jedwabno
Koordynaty x: 19.820240
Koordynaty y: 53.741590
Osoba, ktora pobrala probke: M. Szewczyk

Struktura 7
Nazwa: BD234
Typ: tkanka
Data poboru: 2014.02.10
Region: Bory Dolnoslaskie
Wataha: Zagan
Koordynaty x: 15.257900
Koordynaty y: 51.558500
Osoba, ktora pobrala probke: J.Karwanski
Nacisnij dowolny klawisz, aby wrocic do menu...|
```

Usuwanie pliku bazy danych:

```
C:\Users\sirre\OneDrive\Pulpi  ×  +  v

=====
1. Otworz baze danych
2. Utworz nowa baze
3. Przegląd bazy
4. Sortowanie bazy
5. Modyfikacja struktury
6. Usun plik bazy danych
7. Zakoncz program
=====
Wybierz opcje :
Czy na pewno chcesz usunac baze danych 'baza01.dat'? (T/N): |
```