Rozdział 14. Pliki binarne

Poza plikami tekstowymi omówionymi w poprzednim rozdziale w języku C/C++ możemy również tworzyć pliki binarne. Celem niniejszego rozdziału jest właśnie zapoznanie z zasadami tworzenia i odczytu danych z plików binarnych. Najprostsza definicja pliku binarnego to: plik, który po otworzeniu w edytorze tekstu daje tzw. 'krzaki'. Ich zawartość to po prostu reprezentacja surowych danych zapisanych w pamięci programu, który je utworzył, bez jakiegokolwiek przetwarzania na formę odczytywalną przez człowieka. Na przykład liczba 45 przechowywana w zmiennej typu int zostanie zapisana jako ciąg bajtów   
00 00 00 2D, podczas gdy w pliku tekstowym zostałaby przed zapisaniem przetworzona na znaki '4' i '5', co w formacie ASCII odpowiada bitom 34 i 35. Edytor tekstu otwierając pliki dokonuje operacji odwrotnej, zamieniając dane binarne na znaki, które im odpowiadają. Oczywistym jest, że w wypadku plików binarnych taka zamiana nie ma sensu, stąd biorą się 'krzaki' podczas próby odczytania ich w ten sposób   
np. wspomniana liczba 45 została by zamieniona na znak myślnika tj. „ -”.

14.1 Zasady tworzenia plików binarnych

**Plik binarny** możemy zdefiniowaćjako ciąg bajtów zapamiętanych w pamięci zewnętrznej (dyskowej). Zapis i odczyt pliku binarnego jest zrealizowany w postaci porcji bajtów, a następnie porcja ta może być zinterpretowana jako określony typ danych. Elementem jednostkowym pliku binarnego jest jeden bajt.

**Format pliku binarnego:** [bajty pliku] [EOF]  
gdzie EOF oznacza znak końca pliku (*end of file*)

Odczyt i zapis plików w trybie binarnym realizowany jest odpowiednio za pomocą funkcji:

read(char\*, int) – odczyt danych binarnych z pliku;

write(char\*, int) – zapis danych binarnych do pliku.

Pierwszy parametr to wskaźnik do danych w postaci kodów ASCII, a drugi parametr to liczba bajtów do przesłania.

**Zadanie 1**

*Napisz program zapisujący liczbę całkowitą do pliku binarnego*.

1. #include<iostream>
2. #include<fstream>
3. using namespace std;
4. int main()
5. {
6. ofstream plik\_binarny("nazwa.dat", ios::binary);
7. int liczba = 35;
8. plik\_binarny.write((char\*)(&liczba), sizeof(liczba));
9. plik\_binarny.close();
10. system("pause");
11. }

**Listing 14.1.** Kod programu realizującego Zadanie 1.

Zwróćmy uwagę, że zmienną plikową tworzymy analogicznie jak w przypadku plików tekstowych dodając jedynie tryb binarny tworzonego pliku w linii 6 kodu. Z kolei w linii 8 z użyciem funkcji write odwołując się do adresu pamięci, w którym przechowywana jest zmienna liczba wpisujemy do pliku to co znajduje się tam znajduje.

Jeśli otworzymy plik nazwa.dat w edytorze tekstu jako zawartość zobaczymy znak #, a nie liczbę 35, gdyż właśnie taki znak przypisany jest w tabeli kodu ASCII kodowi 35.

**Zadanie 2**

*Napisz program, który wypisze na ekran liczbę zapisaną w pliku binarnym utworzonym w poprzednim przykładzie*.

Poniższy Listing 14.2 prezentuje rozwiązanie Zadania 2. Rozpoczynamy od otwarcia pliku do czytania w trybie binarnym (linia 6). Następnie w linii 8 następuje wczytanie danych z pliku do zmiennej wczytana\_liczba.

1. #include<iostream>
2. #include<fstream>
3. using namespace std;
4. int main()
5. {
6. ifstream z\_plik\_binarny("nazwa.dat",ios::binary);
7. int wczytana\_liczba;
8. z\_plik\_binarny.read((char\*)&wczytana\_liczba, sizeof(wczytana\_liczba));
9. cout << wczytana\_liczba;
10. system("pause");
11. }

**Listing 14.2.** Kod programu realizującego Zadanie 2.

**Zadanie 3**

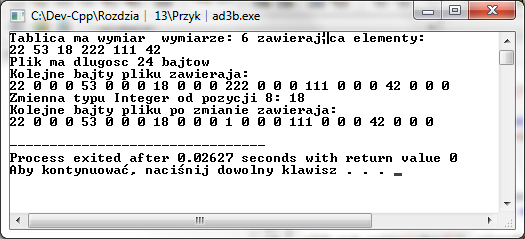
*Napisz program przepisujący liczby z tablicy do pliku binarnego. Wyświetl zawartość kolejnych bajtów pliku, następnie wyświetl zawartość 9-go bajtu pliku, potem zamień zawartość bajtów pliku odpowiadającym trzeciej liczbie z tabeli na liczbę =1.*

Kod programu przedstawia Listing 14.3. W linii 15 otwieramy plik o nazwie dane.dat do pisania w trybie binarnym, a następnie w linii 18 zapisujemy do niego od razu zawartość całej tabeli i zamykamy plik. Następnie zgodnie z kodem w linii 21 otwieramy ponownie ten sam plik tym razem do czytania i pisania. W linii 24 wykorzystując funkcję seekg()pozycjonujemy strumień na końcu pliku i odczytujemy pozycję: powinna ona wskazywać na bajt „pierwszy za ostatnim”, a ostatni ma numer 23. Zatem powinna to być pozycja numer 24, co jest równe ilości bajtów w pliku odczytanej za pomocą funkcji tellg().W linii 28 ponownie pozycjonujemy strumień tym razem na początek pliku i wypisujemy zawartość kolejnych bajtów na ekran (linie 29-33).

Po tej operacji stan strumienia jest bad, ponieważ w pętli próbowaliśmy odczytać bajt po osiągnięciu końca pliku. Każda następna operacja wejścia/wyjścia na tym strumieniu byłaby zignorowana (choć nie zostałby zgłoszony żaden błąd - musimy to zawsze sami sprawdzać!). Dlatego przed dalszym użyciem strumienia musimy go „naprawić” wykorzystując w linii 35 metodę clear(). Pozycjonujemy teraz strumień na bajcie numer 8, czyli przed pierwszym należącym do trzeciej liczby. Zaczynając od tej pozycji czytamy 4 bajty i kopiujemy je do zmiennej k typu int (linie 37-39). Następnie pozycjonujemy lokalizator do pisania na bajcie numer 12=3\*4 (początek czwartej liczby) i zapisujemy tam 4 bajty zmiennej x (linie 37-47). Tak więc czwarta liczba zapisana w pliku powinna zmienić się na 1. Wynik uruchomienia programu przedstawia Rysunek 14.1.

1. #include <iostream>
2. #include <fstream>
3. using namespace std;
4. int main(void) {
5. int T[] = { 22, 53, 18, 222, 111, 42 };
6. int rozmiar = sizeof(T)/sizeof(T[0]);
7. cout << "Tablica ma wymiar wymiarze: " << rozmiar <<   
    " zawierająca elementy: "<<endl;
8. for (int i = 0; i < rozmiar; ++i)
9. cout << T[i] << " ";
10. cout << endl;
12. ofstream plik\_out("dane.dat",ios::out|ios::binary);
13. if (! plik\_out ) cout << "Nie mozna otworzyc pliku" << endl;
15. plik\_out.write((char\*)T, sizeof(T));
16. plik\_out.close();
17. fstream plik("dane.dat",ios::in|ios::out|ios::binary);
18. if (! plik ) cout << "Nie mozna otworzyc pliku" << endl;
20. plik.seekg(0,ios::end);
21. streamsize dlugosc\_pliku = plik.tellg();
22. cout << "Plik ma dlugosc " << dlugosc\_pliku << " bajtow\n";
23. plik.seekg(0);
24. cout << "Kolejne bajty pliku zawieraja:" << endl;
25. int k;
26. while ( (k = plik.get()) != EOF )
27. cout << k << " ";
28. cout << endl;
29. plik.clear();
30. plik.seekg(8);
31. plik.read((char\*)&k,4);
32. cout << "Zmienna typu Integer od pozycji 8: " << k << endl;
34. int x=1;
35. plik.seekp(3\*sizeof(int));
36. plik.write((char\*)&x,4);
37. plik.seekg(0);
38. cout << "Kolejne bajty pliku po zmianie zawieraja:" << endl;
39. while ( (k = plik.get()) != EOF )
40. cout << k << " ";
41. cout << endl;
43. plik.close();
44. }

**Listing 14.3.** Kod programu realizującego Zadanie 3.



**Rysunek 14.1** Przykładowy wynik uruchomienia programu z Listingu 14.3.

Pliki binarne najczęściej wykorzystuje się do zapisu danych przechowywanych w strukturach. W tym celu wykorzystuje się operator reinterpret\_cast<T>(arg). Jest to operator, którego celem jest konwersja przez zamianę typów, które są niepewne lub zależne od implementacji.   
W deklaracji, reinterpret\_cast<T>(arg) , T musi być wskaźnikiem, referencją, typem arytmetycznym, wskaźnikiem na funkcję, lub wskaźnikiem na element.

Załóżmy, że mamy zadeklarowaną strukturę postaci:

struct firma{

char nazwa[30];

float przychod;

};

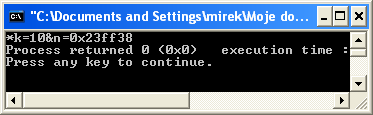
oraz zmienną f typu firma i chcemy zapisać ją do pliku binarnego. W tym celu należy otworzyć plik binarny, a następnie zapisać do niego zmienną f korzystając właśnie z operatora reinterpret\_cast:

ofstream plik("dane\_firmy.dat",ios::binary | ios::app| ios::out);

plik.write(reinterpret\_cast<char\*>(&f),sizeof(f));

Operator rzutowania reintepred\_cast ma postac:

Reinterpred\_cast<nazwa\_typu\_na\_który\_chcemy\_rzutować\_daną>(wyrażenie)

Wyjaśnienie operatora \* i &

#include <iostream>

#include <conio.h>

using namespace std;

int main()

{

int n=10; // deklaracja i definicja zmienej

int \*k; //deklaracja wskaźnika na int

k=&n; //przypisanie zmiennej, która przechowuje adres adresu zmiennej n

cout<<"\*k="<<\*k; //nie powinno dziwić, że pojawi się 10

cout<<"&n="<<&n;// a co to?

getch();

return 0;

}

**Zadanie 4**

*Napisz program, w którym zadeklarujesz strukturę pracownik, a dane pracowników zapiszesz w pliku binarnym.*

Kod realizujący zadanie 4 przedstawia Listing 14.4. W linii 12 tworzona jest zmienna plikowa plik i skojarzona z plikiem dyskowym dane.dat otwartym do zapisywania w trybie binarnym. Ponieważ nie wiemy ilu pracowników zechce wprowadzić użytkownik wprowadzanie danych i zapisywanie ich do pliku realizowane jest w pętli while w zależności od wartości zmiennej wybor, która wczytywana jest ponownie pod koniec pętli, linie 14-20 mogą zostać powtórzone wielokrotnie. W liniach 14-20 dane wczytywane są do zmiennej p typu pracownik, następnie w linii 18 zmienna p zapisywana jest do pliku skojarzonego z zmienną plik. Po skończonym zapisywaniu danych strumień plik zostaje zamknięty- linia 22.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | #include <iostream>  #include <fstream>  using namespace std;  struct pracownik{  char nazwisko[50];  float zarobki;  };  int main()  {  pracownik p, p1;  int wybor=1;  fstream plik("dane.dat",ios::binary | ios::out);  while(wybor){  cout<<"Podaj nazwisko: ";  cin>>p.nazwisko;  cout<<"Podaj zarobki: ";  cin>>p.zarobki;  plik.write(reinterpret\_cast<char\*>(&p),sizeof(p));  cout<<"Czy chcesz wprowadzic nastepnego pracownika (0/1) ? ";  cin>>wybor;  }  plik.close();  } |

**Listing 14.4** Kod programu realizującego Zadanie 4.

**Zadanie 5**

*Napisz program, w którym wczytasz z pliku utworzonego w Zadaniu 4 i wypiszesz na ekran nazwiska i imiona osób w nim zawartych. Następnie zaktualizujesz dane wybranego pracownika i ponownie wypiszesz dane z pliku na ekran.*

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46 | #include <iostream>  #include <fstream>  using namespace std;  struct pracownik{  char nazwisko[50];  float zarobki;  };  main()  {  pracownik p;  int i=1,wybor;  ifstream plik("dane.dat",ios::binary);  if (! plik )  {  cout << "Nie mozna otworzyc pliku" << endl;  return 0;  }  do  {  plik.read(reinterpret\_cast<char\*>(&p),sizeof(p));  if (!plik.eof()) cout<<i<<". "<<p.nazwisko<<" "<<p.zarobki<<endl;  i++;  } while (!plik.eof());  plik.close();  cout<<endl<<"Podaj nr rekordu do edycji: ";  cin>>wybor;  ofstream plik1("dane.dat",ios::binary | ios::in);  plik1.seekp(sizeof(struct pracownik)\*(wybor-1));  cout<<"Podaj nowe dane"<<endl;  cout<<"Podaj nazwisko: ";  cin>>p.nazwisko;  cout<<"Podaj zarobki: ";  cin>>p.zarobki;  plik1.write(reinterpret\_cast<char\*>(&p),sizeof(p));  plik1.close();  ifstream plik2("dane.dat",ios::binary);  i=1;  cout<<"zawartosc pliku po aktualizacji"<<endl;  do  {  plik2.read(reinterpret\_cast<char\*>(&p),sizeof(p));  if (!plik2.eof()) cout<<i<<". "<<p.nazwisko<<" "<<p.zarobki<<endl;  i++;  } while (!plik2.eof());  plik2.close();  } |

**Listing 14.5** Kod programu realizującego Przykład 4

W linii 12 otwieramy plik dane.dat tym razem do czytania w trybie binarnym. Zwróćmy uwagę, że odczytywanie pliku odbywa się w pętli do … while, Samo odczytywanie z pliku zakodowane jest w linii 20, pod zmienna p wczytywane jest tyle bajtów jaki jest rozmiar zmiennej p (sizeof(p)). Na koniec należy pamiętać o zamknięciu pliku- linia 24. Ponownie otwieramy plik dane.dat tym razem do pisania (linia 27). Następnie należy wybrać dane której osoby z kolei chcemy zmienić. Przypuśćmy, że chcemy poprawić rekord o numerze k. Można to zrobić za pomocą funkcji seekp następująco:

plik1.seekp(sizeof(struct pracownik)\*(wybor-1));

Po takim poleceniu kursor do pisania zostanie ustawiony przed rekordem k-tym. W naszym zadaniu jest to zrealizowane w 28 linii kodu. Wczytujemy nowe dane do zmiennej p, a następnie w linii 34 wykorzystując funkcje write zapisujemy tą zmienna do naszego pliku. W tym momencie kolejne bajty pliku zostaną zastąpione nowymi danymi. Po zmianie należy ponownie zamknąć plik. W kolejnych liniach 36-44 plik ponownie zostaje otwarty i wypisana jego zawartość po zmianie.

14.2 Wybrane funkcje operujące na plikach

Przy pracy z plikami binarnymi pomocne są następujące funkcje :

* istream & read (char\* tekst, int bajty) - odczytuje ze strumienia określoną liczbę bajtów,
* ostream & write(const char\* tekst, int bajty) - zapisuje do strumienia określoną liczbę bajtów,
* streampos tellg() - gdzie pokazuje wskaźnik do czytania,
* streampos tellp() - gdzie pokazuje wskaźnik do pisania,
* istream & seekg(pozycja, odniesienie = ios::beg) - pozycjonuje wskaźnik do czytania,
* ostream & seekp(pozycja, odniesienie = ios::beg) - pozycjonuje wskaźnik do pisania,
  + pierwszy argument pozycja mówi, na który bajt będzie wskazywał wskaźnik,
  + drugi argument to punkt odniesienia dla pozycji:
    - beg - początek pliku,
    - cur - obecnie wskazywane miejsce przez wskaźnik,
    - end - koniec pliku.
* int eof() - zwraca niezerową wartość gdy został napotkany koniec pliku,
* int good() - funkcja zwraca wartość niezerową jeśli wszystko jest w porządku,
* int bad() - zwraca wartość niezerową np. gdy strumień będzie wczytywał z pliku, który nie istnieje,
* int fail() - funkcja zwraca wartość niezerową np. gdy oczekujemy z klawiatury wartości liczbowej a otrzymamy tekst,
* void clear (iostate state = goodbit) - funkcja pozwala na ustawianie flag stanu strumienia, domyślnie czyści wszystkie flagi błędów,
* int gcount() - zwraca ilość wczytanych znaków ze strumienia,
* int peek() - funkcja zwraca bajt, który czeka aktualnie na wczytanie, (pozwala podejrzeć co czeka na wczytanie).

14.3 Zadania

1. Mając dany następujący plik wejściowy rozdzielany przecinkami:

*Imie, Nazwisko, Wzrost*

*Jan, Nowak, 1.85*

*Zofia, Kowalska, 2.30*

*Tomasz, Kozak, 1.60*

*Klara, Rzepka, 1.75*

Napisz program wczytujący z niego dane korzystając uprzednio zadeklarowanej struktury. Następnie posortuj te dane rosnąco (dowolnym algorytmem sortowania) według nazwiska i zapisz je do pliku binarnego.

1. Mając dany następujący plik wejściowy rozdzielany przecinkami:

*Nazwisko autora, Tytuł, Cena, ilość egzemplarzy*

*Sienkiewicz, Krzyżacy, 50, 100*

*Prus, Lalka, 78, 23*

*Gombrowicz, Ferdydurke, 66, 55*

Napisz program wczytujący z niego dane korzystając uprzednio zadeklarowanej struktury. Następnie posortuj te dane rosnąco (dowolnym algorytmem sortowania) według ilości egzemplarzy i zapisz je do pliku binarnego.

1. Mając dany następujący plik wejściowy rozdzielany średnikami:

*Nazwisko; imię; Miasto; numer telefonu*

*Grochowska; Marta; Radom; 786987123*

*Zielony; Piotr; Warszawa; 567987123*

*Stoch; Adam; Białystok; 345398634*

Napisz program wczytujący z niego dane korzystając uprzednio zadeklarowanej struktury. Następnie posortuj te dane rosnąco (dowolnym algorytmem sortowania) według miasta i zapisz je do pliku binarnego.

1. Wypełnij dynamiczną tablicę dwuwymiarową liczbami rzeczywistymi, a następnie liczby podzielne przez k podane przez użytkownika przepisz do pliku binarnego. Zamknij plik. Następnie wczytaj z niego liczby do nowej tablicy dynamicznej.
2. Wypełnij dynamiczna tablicę dwuwymiarową losowymi liczbami całkowitymi z przedziału [a,b] określonego przez użytkownika, a następnie liczby będące dzielnikami liczby k podanej przez użytkownika przepisz do pliku binarnego. Zamknij plik. Następnie wczytaj z niego liczby i zapisz je w pliku tekstowym.
3. Wypełnij dynamiczną tablicę dwuwymiarową znakami, a następnie przepisz ją do pliku binarnego. Zamknij plik. Następnie wczytaj z niego znaki i te będące wielkimi literami lub cyframi przepisz do nowej tablicy dynamicznej.
4. W pliku binarnym zostały zapisane dane o strukturze: nazwisko, adres, stanowisko, brutto (dla celów testowych należy wprowadzić dane). Napisz program, który na podstawie tych danych utworzy raport i zapisz go do pliku tekstowego. Ma on zawierać informacje: tytuł raportu, data raportu, kto sporządził, dane z pliku binarnego, podsumowanie i średnia dla wybranego stanowiska.
5. W pliku binarnym zostały zapisane dane o strukturze: towar, dostawca, ilość, cena jednostkowa (dla celów testowych należy wprowadzić dane). Napisz program, który na podstawie tych danych utworzy raport. Raport ma być zapisany w pliku tekstowym i zawierać informacje: tytuł raportu, data raportu, kto sporządził, dane z pliku binarnego, podsumowanie wartości i średnia z wartości dla wybranego dostawcy.
6. W pliku binarnym zostały zapisane dane o strukturze: właściciel, pow. działki w m2, cena 1 m2, (dla celów testowych należy wprowadzić dane). Napisz program, który na podstawie tych danych utworzy raport. Raport ma być zapisany w pliku tekstowym i zawierać informacje: tytuł raportu, data raportu, kto sporządził, ?
7. Utwórz bazę danych ”Książka telefoniczna” zawierającą imiona i nazwiska osób i ich numery telefonu stacjonarnego oraz komórkowego. Każdy element ma ten sam rozmiar w pamięci. Baza danych ma być umieszczona w pliku binarnym, wypełnij ją przykładowymi danymi.
8. Utwórz bazę danych ”Biblioteka” zawierającą autorów, tytuły, ilość egzemplarzy książki w bibliotece. Każdy element ma ten sam rozmiar w pamięci. Baza danych ma być umieszczona w pliku binarnym, wypełnij ją przykładowymi danymi.
9. Utwórz bazę danych ”Studenci” zawierającą imiona i nazwiska osób i ich numery Pesel, rok przyjęcia na studia oraz średnia ocen za ostatni rok. Każdy element ma ten sam rozmiar w pamięci. Baza danych ma być umieszczona w pliku binarnym, wypełnij ją przykładowymi danymi.
10. Napisz program, w którym wczytasz dane z pliku utworzonego w zadaniu 3, a następnie zaktualizujesz numery telefonów wskazanej przez użytkownika osoby. Wypisz również zawartość pliku po modyfikacji na ekran oraz zapisz ją do pliku tekstowego.
11. Napisz program, w którym wczytasz dane z pliku utworzonego w zadaniu 2, a następnie zaktualizujesz ilość egzemplarzy wskazanej przez użytkownika książki. Wypisz również zawartość pliku po modyfikacji na ekran oraz zapisz ją do pliku tekstowego..
12. Napisz program, w którym wczytasz dane z pliku utworzonego w zadaniu 1, a następnie zaktualizujesz dane osoby wskazanej przez użytkownika. Wypisz również zawartość pliku po modyfikacji na ekran oraz zapisz ją do pliku tekstowego.
13. Utwórz program, w którym kolejne n liczb ciągu Fibonacciego zapiszesz do pliku binarnego o nazwie fibonacci.dat.
14. Napisz program tworzący plik binarny losowe.dat wypełniony całkowitymi liczbami losowymi z zadanego przedziału [a, b].
15. Napisz funkcję wyznaczającą pozycję liczby o maksymalnej wartości w ciągu liczb całkowitych zapisanych na dysku w pliku binarnym losowe.dat utworzonym w poprzednim zadaniu. Następnie napisz program, w którym uruchomisz tą funkcję.
16. Napisz funkcję, która dostaje jako argumenty ścieżki dostępu do dwóch plików binarnych i dopisuje zawartość pierwszego pliku na koniec drugiego pliku. Wykorzystaj pliki utworzone w poprzednich dwóch zadaniach tj. fibonacci.dat oraz losowe.dat. Następnie napisz program, w którym uruchomisz tą funkcję.
17. Napisz program symulujący n rzutów kostką. Wyniki kolejnych rzutów zapisz do pliku binarnego rzuty.dat.
18. Napisz funkcję, która dostaje jako argumenty nazwę pliku, dwuwymiarową tablicę o elementach typu int oraz wymiary tablicy i zapisuje binarnie zawartość tablicy oraz jej wymiary do podanego pliku. Następnie napisz program, w którym uruchomisz tą funkcję.
19. Napisz funkcję, która dostaje jako argumenty nazwę pliku, wczytuje zawartość pliku do nowo utworzonej dwuwymiarowej tablicy. Wymiary tablicy powinny być podane w pliku. Napisz funkcję tak, aby była „kompatybilna” z funkcją z poprzedniego zadania. Następnie napisz program, w którym uruchomisz tą funkcję.
20. Dana jest baza struktur Student zawierająca: Nazwisko, Imię, pesel, adres (będący strukturą), średnią z każdego semestru studiów (max 7 semestrów). Z użyciem funkcji napisz program, który z użyciem funkcji będzie:
21. Dodawał studenta do bazy i zapisywał bazę do pliku binarnego (bez użycia tablic);
22. Wyszukiwał studentów o średniej za dany semestr większej od zadanej, jako parametr, i wypisywał ich nazwiska oraz średnie do pliku tekstowego.
23. Zaprojektuj strukturę składająca się z dwóch pól: liczba oraz parzystość. Następnie stwórz w pliku binarnym bazę takich struktur wypełniając ją liczbami całkowitymi z zadanego zakresu [a, b] określając, czy wartość w polu liczba jest parzysta bądź nie i wpisując tę informację do pola parzystość.
24. Utwórz w pliku binarnym dane.dat bazę przechowującą struktury postaci Imię, Nazwisko, Płeć, Rok\_urodzenia. Następnie przepisz do pliku tekstowego dane tych osób, którym pozostało do emerytury mniej niż zadane przez użytkownika k lat. Użyj funkcji.
25. Napisz program, umożliwiający modyfikację danych w bazie danych utworzonej w poprzednim zadaniu. Osobę do modyfikacji należy wyszukać po nazwisku.
26. Utwórz strukturę Rezerwacja zawierającą pola: numer referencyjny, imię i nazwisko klienta, data rezerwacji, data przyjazdu, rodzaj pokoju, liczba nocy. Następnie stwórz bazę danych rezerwacji, zapisz ją w pliku binarnym rezerwacje.dat i wypełnij przykładowymi danymi. Napisz program umożliwiający modyfikację rezerwacji danego klienta.
27. Utwórz strukturę Rezerwacja zawierającą pola: numer referencyjny, imię i nazwisko klienta, data rezerwacji, data przyjazdu, rodzaj pokoju, liczba nocy. Następnie stwórz bazę danych rezerwacji, zapisz ją w pliku binarnym rezerwacje.dat i wypełnij przykładowymi danymi. Napisz program umożliwiający usunięcie z bazy rezerwacji danego klienta.
28. Utwórz strukturę Pożyczkobiorca zawierającą pola imię, nazwisko, wiek, PESEL klienta, wysokość pożyczki, kwoto do spłaty, ilość rat, wysokość raty. Następnie stwórz bazę danych pożyczkobiorców, zapisz ją w pliku binarnym pozyczkobiorcy.dat i wypełnij przykładowymi danymi. Napisz program, wpisujący do pliku tekstowego dane postaci: imię, nazwisko, ile pozostało do spłaty.
29. Utwórz strukturę Pożyczkobiorca zawierającą pola imię, nazwisko, wiek, PESEL klienta, wysokość pożyczki, kwoto do spłaty, ilość rat, wysokość raty. Następnie stwórz bazę danych pożyczkobiorców, zapisz ją w pliku binarnym pozyczkobiorcy.dat i wypełnij przykładowymi danymi. Napisz program umożliwiający modyfikację danych pożyczki (kwota do spłaty, ilość rat, wysokość raty) danego klienta.
30. Napisz program przepisujący liczby z tablicy liczb rzeczywistych podanej przez użytkownika do pliku binarnego. Wyświetl zawartość kolejnych bajtów pliku, następnie zamień zawartość bajtów pliku odpowiadającym k-tej liczbie z tabeli, K- podaje użytkownik.
31. Napisz program przepisujący liczby z tablicy liczb rzeczywistych o podwójnej precyzji podanej przez użytkownika do pliku binarnego. Wyświetl zawartość kolejnych bajtów pliku, następnie zamień w pliku liczbę znajdującą się na k-tym miejscu od końca tabeli licząc, K- podaje użytkownik.
32. Napisz program przepisujący liczby z tablicy znaków wypełnionej przez użytkownika do pliku binarnego. Wyświetl zawartość kolejnych bajtów pliku, następnie zamień w pliku znak znajdujący się na k-tym miejscu, K- podaje użytkownik.