T: Zapis struktur do pliku binarnego w C++.

1. **Formatowanie zapisu do pliku w C++** - do formatowanie używamy manipulatorów :

- setprecision
- fixed

2. Zapis do pliku tekstowego

(a) dla zapisu w pliku tekstowym mamy:

```
ofstream file("num.dat");
short x = 1297;
file <<x;
```

(b) zawartość pliku:

```
00000000: 049 050 057 055                                 1297
```

(c) liczba **short int** jest zapisana jako 4 cyfry

'1'	'2'	'9'	'7'	<EOF>

(d) co w kodzie ASCII

49	50	57	55	<EOF>

3. **Zapis do pliku binarnego**:

Liczba **1297** w **pamięci** jest przechowywana w postaci:

(a) binarnie

00000101	00010001

(b) szesnastowo

11	05

(c) zapis do pliku binarnie

```
ofstream file.open("stuff.dat",ios::out|ios::binary);
short x = 1297;
file.write(reinterpret_cast<char *>(&x), sizeof(x));
```

4. Funkcje obiektu strumienia plików **write** i **read** - zapis do pliku binarnego znaku lub tablicy znaków

```
file.write(address, size);
```

address - początkowy adres sekcji pamięci, który będzie zapisany do pliku. Oczekiwanie, że będzie to adres **znaku (char)** lub **wskaźnika na znak**

size - ilość bajtów pamięci do zapisu, argument musi być **całkowity (integer)**

4.1 Przykład :

```
char znak = 'A';
file.write(&znak, sizeof(znak));
```

4.2 Przykład :

```
char data[] = {'A', 'B', 'C', 'D'};
file.write(data, sizeof(data));
```

```
file.read(address, size);
```

address - początkowy adres sekcji pamięci, który będzie odczytany. Oczekiwanie, że będzie to adres **znaku (char)** lub **wskaźnika na znak**

size - ilość bajtów pamięci do odczytu, argument musi być **całkowity (integer)**

4.3 Przykład :

```
char znak = 'A';
file.read(&znak, sizeof(znak));
```

4.4 Przykład :

```
char data[4];
file.read(data, sizeof(data));
```

5. Funkcje obiektu strumienia plików **write** i **read** - zapis do pliku binarnego **innych typów danych niż znak** lub tablica znaków.

Aby to uczynić, musimy rzutować wskaźnik jednego typu na drugi

```
reinterpret_cast<dataType>(value);
```

dataType - typ, który rzutujemy

value - wartość którą konwertujemy

5.1 Przykład :

```
int x = 27;
file.write(reinterpret_cast<char *>(&x), sizeof(x));
```

5.2 Przykład :

```
const int SIZE = 10;
int numbers[SIZE] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
file.write(reinterpret_cast<char *>(numbers),
sizeof(numbers));
```

5.3 Przykład :

```
const int SIZE = 10;
int numbers[SIZE];
file.read(reinterpret_cast<char *>(numbers),
sizeof(numbers));
```

6. Zapis struktury do pliku binarnego :

```
file.write(reinterpret_cast<char *>(&person),
sizeof(person));
```

WAŻNE !!!

Struktury zawierające wskaźniki nie mogą być zapisane na dysku przy pomocy funkcji **write** i **read.** Ponieważ jeśli struktura jest wczytywana do pamięci podczas wykonywania programu, nie można zagwarantować iż wszystkie zmienne będą w tej samej lokalizacji pamięci. Obiekty klasy **string** zawierają same w sobie już wskaźniki, dlatego nie mogą być zapisane jako część struktury w ten sposób.