

Tablice wielowymiarowe, odczyt z pliku

Dzisiaj PDF krótki, bo już dużo wiemy, więc przydałoby się na tej wiedzy popracować.

Na pierwszy ogień, tablice

Tablica wielowymiarowa, to tak naprawdę tablica jednowymiarowa zawierająca tablice (lub tablicę).

```
int arr[5][5], counter=0;
  for (int i=0; i<5; i++){
    for (int j=0; j<5; j++){
      arr[i][j] = counter++;
    }
  }
  cout<< "{ ";
  for (int i=0; i<5; i++){
    cout<< "{ ";
    for (int j=0; j<5; j++){
      cout<<arr[i][j]<<" ";
    }
    cout<<" },"<<endl;
  }
```

To w zasadzie cała filozofia, tablica wielowymiarowa to po prostu tablica tablic. Na nasze potrzeby wystarczą w zupełności tablice dwuwymiarowe, które możemy sobie porównać do układu, gdzie każdy element ma swoje określone współrzędne (wiersz i kolumnę). W prezentowanym powyżej przykładzie mamy tablicę 5x5 (5 wierszy – pierwsza wartość i 5 elementów w każdym wierszu – druga wartość). Plansza do szachów, plansza do Scrabble, do gry w statki, układ współrzędnych, macierz – you name it, przykładów jest mnóstwo.

Odczyt danych z pliku

Temat odrobinę bardziej rozległy niż powyższy, ale bardzo ważny. Zapis i odczyt z pliku otwierają morze możliwości przed programistą. W zasadzie wszystkie programy, które do tej pory napisaliśmy operowały bezpośrednio w pamięci ulotnej, co znaczyło, że wszelkie wprowadzone dane bezpowrotnie ginęły po zakończeniu pracy programu. Teraz zajmiemy się jedynie odczytem z pliku, na następnym spotkaniu omówimy zapis. Korzystać będziemy z biblioteki *fstream* – od **File Stream**. Zacząć musimy zatem od załączenia odpowiedniej biblioteki:

```
#include <fstream>
```

Następnym krokiem będzie utworzenie tzw. „uchwyty”, czyli obiektu klasy *ifstream*.

```
ifstream file;
```

Przy użyciu obiektu nazwanego przeze mnie *file* możemy korzystać z każdego znajdującego się na dysku pliku, ale musimy podać do niego ścieżkę:

```
file.open("plik.txt");
```

Powyższy przykład zakłada, że pliczek „plik.txt” znajduje się w tym samym katalogu co program. Jeśli z dowolnego powodu tak nie jest, to należy podać dokładną ścieżkę do tego pliku (względną lub bezwzględną).

Następnie, korzystając z metody `good()` sprawdzimy, czy plik został prawidłowo wczytany:

```
if (file.good()){
    cout<<"plik otwarty, przechodzę do czytania!";
}
else cout<<"plik niedostępny";
```

Aby odczytywać plik wiersz po wierszu możemy korzystać z funkcji `getline()`. Przyjmuje ona dwa argumenty – strumień wejściowy i zmienną, do której zapisywane będą odczytywane wartości

```
while (getline(file, line)){
    cout<<line<<endl;
}
```

Warto pamiętać, że każdy otwarty plik należy też zamknąć:

```
file.close();
```

Wczytywać możemy też pojedyncze wyrazy, do tego skorzystamy z operatora `>>`

```
while (!file.fail()){
    file>>word;
    cout<<word<<endl;
}
```

Zadania

1. Transponuj zadaną macierz:

$$\begin{matrix} -1 & 2 & 8 \\ 3 & -6 & 3 \\ 0 & 9 & 1 \end{matrix}$$

2. Napisz program, który wczyta z pliku `sudoku.txt` 9 wierszy i sprawdzi, czy każdy z nich zawiera unikatowe cyfry od 1 do 9.

TO BĘDZIE ŁATWIEJSZE PO NASTĘPNYCH ZAJĘCIACH.