

Programowanie 1

Zadanie 12

Piotr Błaszyński

9 stycznia 2023

Obsługa struktur danych i wskaźników (do struktur i wewnątrz nich). Zadaniem jest, przy pomocy odpowiednich struktur danych, wczytać z pliku i zaprezentować prostą mapę. Drugim fragmentem zadania jest prezentacja odległości poszczególnych punktów na mapie od punktu startowego znajdującego się w lewym górnym rogu.

Mapa składa się z pojedynczych elementów o wielkości jednego znaku na ekranie, element może być podłogą (znak '.'), albo ścianą (znak '#'). Format pliku jest następujący: - pierwsza linia: liczba wierszy mapy (r), - druga linia: liczba kolumn mapy (c), - kolejne r linii: poszczególne wiersze mapy, każdy o długości c.

Poszczególne komórki należy wczytać do struktury (jedna komórka - jedna struktura). Komórki przechowują wskaźnik do wszystkich 4 swoich sąsiadów. Ponadto wszystkie komórki są przechowywane w dwuwymiarowej tablicy (r * c) alokowanej dynamicznie.

Przykładowa struktura (można a nawet TRZEBA ją trochę zmodyfikować):

```
struct cell
{
    struct cell *left_cell;
    struct cell *up_cell;
    struct cell *right_cell;
    struct cell *down_cell;
    int left_visited;
    int up_visited;
    int right_visited;
    int down_visited;
    char value;
};
```

Przykładowe pliki są w katalogu test. W ostatecznym rozwiązaniu należy wczytać wszystkie pliki i zaprezentować wszystkie wyniki. Czyli dla każdego pliku: - prezentacja mapy, - prezentacja mapy z odlegościami poszczególnych punktów podłogi od startu. Poruszać się można oczywiście tylko po podłodze i tylko pionowo i poziomo. Jeżeli do jakiegoś elementu nie można dojść to należy wypisać 0. W punkcie startu zawsze jest podłoga. Jeśli odległość jest większa od 9 to wyświetlamy tylko ostatnią cyfrę.

Dla przykładowego pliku:

```
3
4
....
.#.#
....
```

Należy wypisać:

```
Mapa :
....
.#.#
....
```

Odległości:

```
0123
1#3#
2345
```

PS. Proszę się zastanowić nad tym czy odległości będą wyznaczane zawsze tak samo, czy będą zależne od kolejności przechodzenia elementów (czy najpierw idziemy w prawo, lewo czy może w dół lub górę).

Dla zainteresowanych: na stronie <https://github.com/piotrbla/pngMap> znajduje się prosty projekt generujący obraz w formacie png (z przykładem generowania), potrzebne pliki:

- lodepng.cpp
- lodepng.h
- main.cpp (przykład użycia)

Przy pomocy tego kodu można przedstawić znacznie większe labirynty i ich rozwiązania.