

Warsztat programisty – Zadanie projektowe.

Etap 2 – format i weryfikacja danych wejściowych

Łukasz Kuszner, Radosław Ziemann

4 grudnia 2024

Ten etap polega głównie na wczytaniu danych wejściowych i sprawdzeniu, czy odpowiadają one specyfikacji podanej w opisie poniżej. Proszę przyjąć, że dane są w postaci tekstu (ang. plain text) kodowanego w ASCII.

Format danych wejściowych

W pierwszej linii jedna liczba całkowita n z przedziału $[1..100]$ oznaczająca liczbę konfiguracji. W kolejnych n liniach znajduje się opis konfiguracji figur na szachownicy.

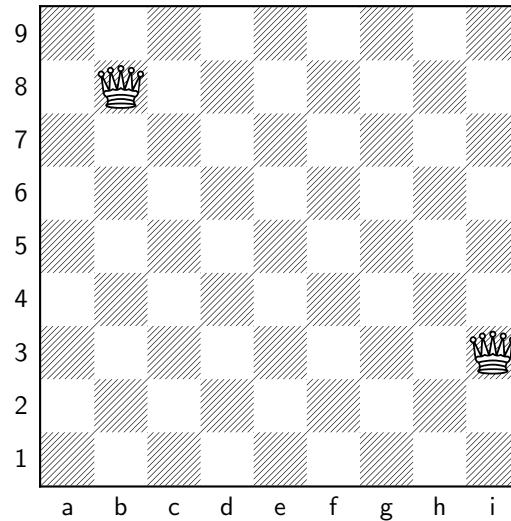
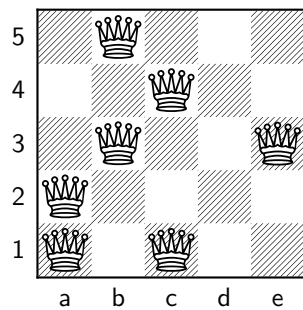
Na opis każdej z konfiguracji składają się w kolejności: liczba całkowita k z przedziału $[1..9]$ będąca rozmiarem szachownicy, liczba figur m z przedziału $[0..k^2]$, i położenie każdej z nich.

Położenie każdej figury to litera Q oznaczająca hetmana (ang. queen) oraz oznaczenie kolumny i wiersza gdzie figura jest postawiona zgodnie z notacją szachową (ang. algebraic notation system) najpierw litera a później cyfra. Na przykład Qb5 oznacza hetmana na polu szachownicy w drugiej kolumnie (licząc od 1) i piątym wierszu.

Przykład

```
2
5 7 Qa1 Qc4 Qb3 Qe3 Qb5 Qc1 Qa2
9 2 Qi3 Qb8
```

W powyższym przykładzie mamy dwie konfiguracje. W pierwszej jest 7 figur na szachownicy o rozmiarze 5x5 a w drugiej konfiguracji są dwie figury na szachownicy o rozmiarze 9x9.



Obie opisane konfiguracje pokazane są powyżej z wykorzystaniem następujących komend:

```
\chessboard[
    maxfield=e5,
    setwhite={Qa1, Qc4, Qb3, Qe3, Qb5, Qc1, Qa2},
    showmover=false,
]
```

oraz:

```
\chessboard[
    maxfield=i9,
    setwhite={Qi3, Qb8},
    showmover=false,
]
```

Sprawdzenie poprawności danych

Należy sprawdzić, czy dane podane do programu są zgodne z opisem. W szczególności:

- Czy liczby n , k , m są istotnie liczbami z podanego zakresu.
- Czy położenie każdej figury jest poprawne: składa się z cyfry oraz małej litery i mieści się na szachownicy.

- Czy liczba konfiguracji odpowiada deklarowanej.
- Czy liczba figur odpowiada deklarowanej.
- Czy nie występuje sytuacja, gdy dwie figury znajdują się na tym samym polu

Przykładowe dane nieprawidłowe.

Przykład 1

```
1
5 7 Qa1 Qc4 Qb3 Qe3 Qb5 Qc1 Qa2
9 2 Qi3 Qb8
```

Liczba konfiguracji nie odpowiada deklarowanej.

Przykład 2

```
2
5 7 Qa1 Qc4 Qb3 Qe3 Qb5 Qc1 Qa2
9 1 Qi3 Qb8
```

Liczba figur w konfiguracji numer 2 nie odpowiada deklarowanej. Miała być jedna a są dwie.

Przykład 3

```
2
5 7 Qa1 Qc4 qb3 Qe3 Qb5 Qc1 Qa2
9 2 Qi3 Qb8
```

W konfiguracji numer 1 jest błędne oznaczenie trzeciej figury, Miała być duża litera Q a jest mała.

Przykład 4

```
0
```

Nieprawidłowa liczba konfiguracji – miała być co najmniej jedna.

Spodziewane efekty

- Elementy z etapu 1: konfiguracja przy pomocy `cmake`, historia zmian w repozytorium `git`, uruchomienie skryptem w Bashu.
- Skrypt można uruchomić podając plik z danymi wejściowymi jako parametr wywołania skryptu.
- W przypadku podania pliku z poprawnymi danymi powstaje dokument w formacie PDF z wszystkimi podanymi konfiguracjami. Skrypt kończy się kodem 0 (wykonanie bezbłędne).
- W przypadku podania pliku z niepoprawnymi danymi nie powstaje dokument PDF, skrypt kończy się zwracając kod różny od zera a na standardowy strumień błędów (`stderr`) wyświetlana jest informacja o błędzie (im dokładniejsza tym lepiej).
- Może się zdarzyć, że plik nie powstanie pomimo podania poprawnych danych. Wtedy skrypt również musi się zakończyć zwracając kod różny od 0. Przykładowe scenariusze, które należy brać pod uwagę:
 - nie udało się otworzyć/odczytać pliku z danymi wejściowymi,
 - nie udało się zapisać pliku wyjściowego
 - wystąpił inny problem w trakcie wykonywanego skryptu, np: `pdflatex` zwrócił błędy, program napisany w C uległ awarii itp.

Dokumentacja

Elementy dokumentacji które mogą się przydać:

- LaTeX: pakiet `chessboard`, pakiet `xskak`
- Wczytywanie danych w C: funkcje: `getline`, `scanf`