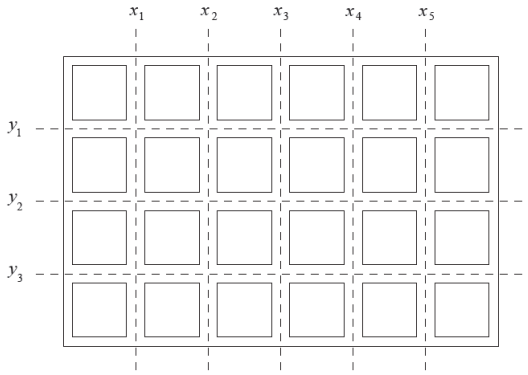
Praca domowa 01 – plate

Termin zwrotu : 20 października godz. 23.00
Zadanie uznaje się za zaliczone, gdy praca oceniona zostanie na co najmniej 6 pkt.

Dana jest tafla niejednorodnego materiału. Taflę należy pociąć na $n * m$ pojedynczych elementów ułożonych w n wierszy i m kolumn. Cięcia możemy dokonywać wzdłuż pionowych i poziomych linii (zaznaczonych na rysunku liniami przerywanymi). Jedno przecięcie tafli wzdłuż wybranej pionowej lub poziomej linii dzieli ten kawałek na dwa mniejsze. Każde cięcie tafli jest obarczone pewnym kosztem wyrażającym się dodatnimi liczbami całkowitymi. Koszt ten nie zależy od długości cięcia, a jedynie od linii wzdłuż której tniemy. Oznaczmy koszty cięcia kolejnych pionowych linii przez $x_1, x_2, \ldots, x_{m-1}$, a wzdłuż poziomych linii przez $y_1, y_2, \ldots, y_{n-1}$. Koszt pocięcia całej tafli na pojedyncze elementy to suma kosztów kolejnych cięć. Należy obliczyć minimalny koszt pocięcia całej tafli na pojedyncze elementy.



Przykładowo, jeżeli potniemy taflę przedstawioną na rysunku, najpierw wzdłuż linii poziomych, a następnie każdy z otrzymanych kawałków wzdłuż linii pionowych, to koszt takiego pocięcia wyniesie y1 +y2 +y3 + 4 * (x1 +x2 +x3 +x4 +x5).

Należy napisać program, który wczyta dane z pliku tekstowego, obliczy minimalny koszt pocięcia całej tafli na pojedyncze elementy oraz wypisze wynik. Plik wejściowy (wskazywany przez parametr `<input_file>` programu) zawiera w kolejnych liniach ciąg nieujemnych liczb całkowitych $n, m, x_1, x_2, \ldots, x_{m-1}, y_1, y_2, \ldots, y_{n-1}$. Długość pojedynczej linii pliku wejściowego ani ilość liczb w linii nie jest w jakikolwiek sposób określona (format swobodny).

Program ma być zapisany wyłącznie w dwóch plikach `Plate.java` zawierającym implementację algorytmu podziału, oraz `Main.java` zawierającym implementację programu głównego. Program nie może korzystać z jakichkolwiek bibliotek zewnętrznych.

Proces kompilacji programu musi być możliwy z użyciem komendy

```
javac -Xlint Plate.java Main.java
```

Uruchomienie programu winno być możliwe z użyciem komendy

```
java Main <input_file>
```

Przykładowy wynik końcowy (w strumieniu wyjściowym nie powinny pojawiać się jakiekolwiek inne elementy – np. wydruki kontrolne) :

```
Koszt cięcia : 15
```

Wymagania :

- Klasa implementująca elementy programu głównego winna zostać zdefiniowana w pliku `Main.java`
- Klasa implementująca mechanizm podziału winna być zdefiniowana w pliku `Plate.java`
- W pliku README.pdf winien być zawarty opis organizacji struktur danych, szczegółowy opis algorytmu oraz analiza/dyskusja złożoności obliczeniowej zaproponowanego rozwiązania.

Sposób oceny **:**

- 1 pkt – **Kompilacja** : każdy z plików winien być kompilowany bez jakichkolwiek błędów lub ostrzeżeń (w sposób omówiony wyżej)
- 1 pkt – **Wykonanie** : program powinien wykonywać się bez jakichkolwiek błędów i ostrzeżeń (dla pliku danych wejściowych zgodnych z wyżej zamieszczoną specyfikacją) z wykorzystaniem omówionych wyżej parametrów linii komend
- 2 pkt – **README** : plik README.pdf dokumentuje w sposób kompletny i właściwy zarówno struktury danych, strategię poszukiwania rozwiązania (algorytm) oraz szacowaną złożoność obliczeniową przyjętego rozwiązania.
- 1 pkt – **Komentarze wewnętrzne** : czy program jest skomentowany w sposób zapewniający zrozumienie jego działania, oraz wyjaśniający warunki, które muszą zachodzić przed i po wykonaniu każdej z funkcji.
- 1 pkt – **Styl kodowania** : Styl oceniany jest w oparciu o wskaźniki NCSS oraz CCN (patrz dokument *'Zasady oznaczania i przesyłania prac'*) oraz dodatkowo z uwzględnieniem następujących elementów : Czy funkcje i zmienne posiadają samo-wyjaśniające nazwy ? Czy podział na funkcje ułatwia czytelność i zrozumiałość kodu ? Czy funkcje eliminują (redukują) powtarzające się bloki kodu ? Czy wcięcia, odstępy, wykorzystanie nawiasów itp. (formatowanie kodu) są spójne i sensowne ?
- 4 pkt – **Poprawność algorytmu** : czy rozwiązanie problemu zostało zaprojektowane i zaimplementowane poprawnie, przy czym za merytorycznie poprawny algorytm można uzyskać dwa punkty, a dwa kolejne za te wyróżniające elementy/cechy rozwiązania (trafność doboru struktury danych, efektywność algorytmu itp.).