Temat: NCDC House of Talents - ostatni etap rekrutacji
Nadawca: NCDC House of Talents <talents@ncdc.pl>
Data: 14.05.2019, 16:03
Adresat: tomek.prof@wp.pl

Cześć *Tomasz*!

Przed Tobą ostatni etap rekrutacji. Mamy dla Ciebie zadania - tym razem praktyczne. Napisz program(y) w **Javie!**

Ważne dla nas jest to, jak piszesz, więc nawet jeśli nie uzyskasz oczekiwanego rezultatu prześlij do nas swój kod.

Programów jest pięć. Można za nie otrzymać odpowiednio do 100, 200, 75, 75 i 150 punktów. Samodzielnie zdecyduj, które/ile programów chcesz napisać. Jeśli wybierzesz mniej niż jeden, nie będziemy mogli zaproponować Ci udziału w HoT#4 ;-)

Wracając do zadań:

* We wszystkich zadaniach limit czasu wykonywania programu wynosi 30 sekund.
* Nie ma znaczenia czy program wykona się szybciej, czy wolniej; byle nie dłużej niż 30 sekund.
* Każdy program musi się zakończyć kodem 0.
* W rozwiązaniu program ma wypisać dokładnie i wyłącznie to, co wskazane.
* Warto jednak wiedzieć, że występujące na końcach wierszy znaki spacji, a po ostatnim wierszu rozwiązania dodatkowo znaki nowego wiersza bezduszny robot testujący Fryderyk nonszalancko ignoruje.
* Wielkość liter w danych wejściowych i wynikach jest istotna.
* Każdy program powinien być odporny na błędne dane wejściowe i - o ile nie zaznaczono inaczej - w przypadku błędnych danych wejściowych, program powinien wypisać słowo "klops" (bez cudzysłowu).
* Bezduszny robot testujący Fryderyk będzie używał Javy 11.
* STDIN pojawia się w opisach zadań nieprzypadkowo
* Komendy, którymi programy będą kompilowane i uruchamiane pojawiają się w opisach zadań również nieprzypadkowo - bezduszny robot testujący Fryderyk strasznie denerwuje się za każdym razem, gdy programy nie kompilują się lub nie uruchamiają.

Zadanie 1. (maksymalnie 100 punktów)

Doktorant, królik Zbigniew, w roztargnieniu pomylił kserokopiarkę z niszczarką, a na kartce były dane do wyznaczania trajektorii lotu. Bez tego kosmolot nie doleci na Księżyc.
Zorientował się dopiero wtedy, kiedy zmełł już całą kartkę i nie mógł znaleźć tacki, na której powinna pojawić się kopia (no cóż, taka gapa).
Kartka z danymi przeszła przez niszczarkę. Długa kartka. Niszczarka pocięła kartkę na paski, a każdy z pasków na odcinki. Mimo wszystko królik Zbigniew musi przed świtem podsumować słupki, bo inaczej zła czarownica uwięzi wszystkich studentów z jego uczelni i nakaże im sumowanie wszystkiego ręcznie, na umowę o dzieło. Całe szczęście, że każdy ze słupków miał na kartce inny kolor tła - dzięki temu udało się złożyć odcinki w całe słupki (choć kolejność odcinków jest losowa).

Pomóż królikowi Zbigniewowi, uratuj pozostałych studentów i tym samym pomóż pokonać trudności w misji Kosmolotu.

Dane wejściowe:
plik CSV - nie wiadomo ile ma kolumn, ale może ich być dużo. Układ kolumn nie jest znany. Niech będzie, że wierszy może być 5000 i tyle samo kolumn.

Poszczególne komórki mogą zawierać nazwy kolumn (nazwą jest łańcuch znaków ujęty w nawiasach klamrowych, np. {nazwa_kolumny}), dane (liczby całkowite większe od -10000 i mniejsze od 10000) lub jakieś śmieci. "Mogą zawierać" oznacza, że nazw kolumn, danych lub śmieci może nie być wcale, mogą występować dokładnie raz lub mogą występować wiele razy. Sumujemy tylko dane (śmieci nie mają być uwzględniane w sumie, ale program nie powinien się na nich wysypać).

Plik z programem powinien nazywać się Krolik.java

Program zostanie skompilowany następującą komendą:
javac Krolik.java

Program będzie uruchamiany komendą:
java Krolik nazwa_kolumny < plik.csv

Oczekiwany wynik:
Program zwraca sumę z kolumny, w której występuje nazwa wskazana jako parametr. Brak możliwości jednoznacznego wskazania kolumny, z której suma ma zostać zwrócona oznacza błąd danych wejściowych.

przykładowy stdin (w formie pliku csv - każdy wiersz kończy się znakiem końca linii):
aaa,1,{aaa},5,4
2,{bbb},2,{5},{ccc}
1,1,2,3,{bbb}
{{}},1,1,1,{ddd}

Przykładowe wyniki działania programu:
java Krolik aaa < plik.csv
5

java Krolik 3 < plik.csv
klops

java Krolik '{}' < plik.csv
3

java Krolik bbb < plik.csv
klops

java Krolik ccc < plik.csv
4

java Krolik ddd < plik.csv

4

Zadanie 2. (maksymalnie 200 punktów)

Ponieważ słupki udało się zsumować, zwierzątka zaplanowały start tuż po podwieczorku. Jakież było ich rozczarowanie, gdy przed drugim śniadaniem zorientowały się, że zły wilk Błażej, wypił całe paliwo rakietowe ze zbiorników zatankowanego kosmolotu. No bez sensu, nie? Nie pomyślał o innych zwierzątkach, które chciały lecieć w podróż na Księżyc./p>

Sarenka Kasia i gąska Karolina głowią się jak by tu ponownie napełnić zbiorniki.
"Najlepiej z kranu!" - krzyknęła sarenka Kasia.
"Najlepiej z dwóch!" - krzyknęła gąska Karolina.
"Ta, z miliona!" - burknął zły wilk Błażej, co nie spotkało się z entuzjazmem.
"Jeszcze słowo i będziesz robił za Łajkę, draniu!" - krzyknęła sarenka Kasia, z charakterystycznym dla siebie tupnięciem kopytkiem, które wykonuje, kiedy werbalizuje swoje zdenerwowanie.

Dopiero po chwili dotarło do wszystkich, że to nie taki głupi pomysł, bo wtedy zbiorniki napełnią się bardzo, bardzo szybko i być może wcale nie będzie potrzeby odroczenia startu.

"Pamiętaj Kasiu, że musimy wiedzieć, jaka jest temperatura paliwa!" - przypomniała gąska Karolina - "Poprzednim razem tego nie sprawdziłyśmy i borsuk Wasyl naprzemiennie wykrzykiwał jakieś brzydkie słowa i chuchał na zbiornik z zamarzniętym paliwem..."

Dane wejściowe:
ilość paliwa potrzebnego do lotu, przepływność każdego kranu, temperatura paliwa w każdym kranie.
Nie trzeba się przejmować stratami ciepła przy napełnianiu zbiorników (ponieważ jest to szczęśliwa, magiczna kraina, gdzie woda nie stygnie, ani nie paruje).
Umówmy się, że jednocześnie paliwa nie będzie lać więcej niż milion kranów (a jeśli w danych wejściowych będzie więcej, to należy je uznać za niepoprawne), a ilość wymaganego paliwa nie będzie większa niż 100000000 litrów.
Woda (bo tak naprawdę to ta rakieta lata na wodę) może mieć nie więcej niż 90 stopni i nie mniej niż 1 (tak, wiemy co to punkt potrójny :)).
Dane wejściowe mogą mieć dokładność do pięciu miejsc po przecinku. Wyniki należy podawać z dokładnością do piątego miejsca po przecinku. Separatorem dziesiętnym jest kropka.

Plik z programem powinien nazywać się Wilk.java

Twój program zostanie skompilowany następującą komendą:
javac Wilk.java

Program będzie uruchamiany komendą:
java Wilk < plik.csv

Oczekiwany wynik:
Trzeba obliczyć czas napełniania i końcową temperaturę wody (paliwa rakietowego).

Przykładowy stdin dla dwóch kranów (w formie pliku csv - każdy wiersz kończy się znakiem końca linii):
10.12345

100.12345 20.12345
100.12345 20.12345

pierwszy wiersz to ilość paliwa potrzebnego do lotu - w tym przypadku 10.12345 m^3
W pozostałych wierszach mamy odpowiednio przepływność kranu (w drugim wierszu 100.12345 l/min, w trzecim 100.12345 l/min) i temperaturę wody płynącej z kranu (w drugim wierszu 20.12345 stopni Celsjusza, w trzecim 20.12345 stopni Celsjusza)
Trzeba pamiętać, że poszczególne krany mogą mieć różne przepływności i temperatury wody, a dane mogą być niepoprawne na różne sposoby (przygotowujący je osioł Czesław jest bardzo twórczy).

Wynik (stdout)
3033.2904
20.12345

gdzie pierwsza linijka oznacza liczbę sekund potrzebnych do napełnienia zbiorników
druga linijka to temperatura paliwa w zbiornikach (w stopniach Celsjusza).

Zadanie 3. (maksymalnie 75 punktów)

Kotka Paprotka, zrzuciła donicę, z nomen omen paprotką. Popatrzyła z dumą na swoje dzieło, fuknęła i wróciła do swoich zajęć, tj. wygrzewania się na parapecie w siedzibie NASA.

Heh - pomyślał mądry pies Sebastian - rozłożyła ją na czynniki pierwsze. Pani Jolanta, inżynier lotu, bardzo lubiła tę paprotkę. Może udałoby się ją poskładać? Tyle, że czasem brakuje pewnych elementów.

Trzeba stworzyć model matematyczny doniczki i odwrócić proces.

Dane są (albo i nie):
* W pierwszej linii - doniczka - liczba (może brakować tej liczby, ale wtedy mamy pewność, że mamy wszystkie czynniki pierwsze)
* W drugiej linii oddzielone przecinkami czynniki pierwsze (ich liczba może być równa liczbie wszystkich czynników pierwszych, ale nie musi. Nadmiar czynników oznacza, że dane wejściowe są błędne)
Wszystkie liczby są liczbami naturalnymi, większymi od jeden.

Plik z programem powinien nazywać się Paprotka.java

Twój program zostanie skompilowany następującą komendą:
javac Paprotka.java

Program będzie uruchamiany komendą:
java Paprotka < plik.csv

Przykładowy stdin (w formie pliku csv - każdy wiersz kończy się znakiem końca linii):
48
2,2,2,2,3

Odpowiedzią jest:

* w pierwszej linii - liczba (doniczka)
* w drugiej linii - oddzielone przecinkami - BRAKUJĄCE czynniki pierwsze.

czyli dla przykładu powyżej odpowiedź to:
48

(w pierwszej linii 48, w drugiej linii po prostu znak końca linii, bo nie brakowało żadnych czynników pierwszych).

Gdyby w pliku wejściowym było:
48
2,2,2,2
odpowiedzią byłoby:
48
3

Gdyby w pliku wejściowym było:
48

wówczas odpowiedzią byłoby:
48
2,2,2,2,3

Gdyby w pliku wejściowym było:

2,2,2,2,3
wówczas odpowiedzią byłoby:
48

itd.

Zadanie 4. (maksymalnie 75 punktów)

Trajektoria kosmolotu.
Borsuk Wasyl nie dał wiary zapewnieniom sarenki Kasi, że tym razem paliwo mu w locie nie zamarznie i stanowczo odmówił pilotowania kosmolotu. Naturalnym kandydatem do roli pilota wydał się zły wilk Błażej...
Zwierzątka szczęśliwie wystartowały na Księżyc.
Żółw Wiktor niespecjalnie ogarnął co się wydarzyło, ponieważ wszystko działo się zbyt szybko. Poza tym, kiedy silniki odpaliły, schował się w skorupie (taki trochę strusi odruch).
Prosi Ciebie o pomoc w narysowaniu trajektorii lotu, żeby mógł sobie wszystko obejrzeć na spokojnie.

Zły wilk Błażej, z nieznanych względów, prowadził kosmolot nie po linii prostej, lecz po kursie łamanym... Wyglądało jakby tropił węża - trajektoria lotu przypominała przebieg trójkątny. Narysuj trajektorię, dla podanej na wejściu amplitudy i długość lotu, wg przykładów.

Maksymalna długość ścieżki lotu (zwizualizowana liczbą gwiazdek), to 50000
maksymalna amplituda ścieżki lotu, to 50000. Obie wartości są większe od zera.

Plik z programem powinien nazywać się Trajektoria.java

Twój program zostanie skompilowany następującą komendą:
javac Trajektoria.java

Program będzie uruchamiany komendą:
java Trajektoria <amplituda> <dlugosc_lotu>

Przykładowe wyniki działania programu znajdziesz w pliku Trajektoria.png. W przykładach spacje są przedstawione jako kolorowe kropki.

Zadanie 5 (maksymalnie 150 punktów)

Galaktyka spiralna
Jeżyk Mirek bardzo lubi oglądać obiekty głębokiego nieba w swoim małym refraktorze. Galaktyki składają się z gwiazdek i pustej przestrzeni (teleskop jeżyka Mirka jest za mały, by dostrzec więcej). Czasami jednak jeżyk Mirek myśli jak by to było, gdyby miał teleskop z dużym lustrem. Zamiast dawać jeżykowi Mirkowi jabłko (jeże nie jedzą jabłek!), narysuj jak wyglądałaby jego ulubiona galaktyka w teleskopie.

podana jest wielkość teleskopu (1,2,3, ... 10000)
ścieżka w galaktyce ma zawsze szerokość 1 spacji

jako parametr podajemy także orientację (N,E,S,W)

Po narysowaniu obrazu galaktyki, należy podać długość spirali, w latach świetlnych, przy czym jedna spacja, reprezentuje jeden rok świetlny

Plik z programem powinien nazywać się Galaktyka.java

Twój program zostanie skompilowany następującą komendą:
javac Galaktyka.java

Program będzie uruchamiany komendą:
java Galaktyka <wielkosc_teleskopu><orientacja>

Przykładowe wyniki działania programu znajdziesz w plikach Galaktyka1.png i Galaktyka2.png. W przykładach spacje są przedstawione jako kolorowe kropki.

Kod(y) programu/ów odeślij nam odpowiadając na tę wiadomość.

Powodzenia!

Pozdrawiamy

Zespół NCDC

Nordic Consulting & Development Company

NCDC Polska S.A.

phone: [+48 91 881 96 00]
mail: talents@ncdc.pl

Galaktyka1.png

```
Galaktyka - przykłady:

dla parametru: 1W

***
..*
*.*
***
3


dla parametru: 2W

****
...*
**.*
*..*
****
6


dla parametru: 3W

*****
....*
***.*
*.*.*
*...*
*****
10


dla parametru: 4W

******
.....*
****.*
*..*.*
*.**.*
*....*
******
15


dla parametru: 5W

*******
......*
*****.*
*...*.*
*.*.*.*
*.***.*
*.....*
*******
21
```

Galaktyka2.png

```
Galaktyka - przykłady c.d.:

dla parametru: 1W

***
..*
*.*
***
3


dla parametru: 1N

**.*
*..*
****
3


dla parametru: 2N

***.*
*.*.*
*...*
*****
6


dla parametru: 2E

****
*..*
*.**
*...
****
6


dla parametru: 2S

*****
*...*
*.*.*
*.***
6
```

Trajektoria.png

```
Trajektoria - przykłady:

dla parametrów: 1 10

**********


dla parametrów: 2 10

*.*.*.*.*
.*.*.*.*.*


dla parametrów: 2 5

*.*.*
.*.*


dla parametrów: 3 17

*...*...*...*...*
.*.*.*.*.*.*.*.*
..*...*...*...*


dla parametrów: 3 15

*...*...*...*
.*.*.*.*.*.*.*
..*...*...*...*


dla parametrów: 7 16

*..........*
.*........*.*
..*......*...*
...*....*.....*
....*...*
.....*.*
......*


dla parametrów: 9 4

*
.*
..*
...*

itd.
```

Załączniki:

Galaktyka1.png	10,0 KB
Galaktyka2.png	7,6 KB
Trajektoria.png	11,6 KB