Ruchy ramienia robota RRR

Sterowanie ramienia pewnego robota polega na wydawaniu na podłączonej konsoli kolejnych komend `L` lub `P` w zależności od tego, czy chcemy aby jego ramię wykonało odpowiednio przesunięcie w lewo bądź w prawo na jednakową odległość. Prawidłowa sekwencja ruchów rozpoczyna się zawsze od ruchu w lewo oraz kończy powrotem do miejsca startu. Niestety konsola się popsuła i pozostaje sterowanie niskopoziomowe za pomocą specjalnego kodu. Zapis ruchu ramienia w tym kodzie jest ciągiem liczb odpowiadających kolejnym ruchom w prawo. Wartością odpowiedniej liczby dla danego ruchu jest liczba wszystkich ruchów w prawo (włącznie z danym ruchem) wykonana od czasu, gdy ramię ostatnio znajdowało się w tym samym miejscu, co przed wykonaniem tego ruchu. Cały proces produkcji został jednak kiedyś zapisany w starszej wersji kodu. Ten z kolei wykorzystywał ciąg liczb będących ilością wszystkich ruchów w lewo poprzedzających dany ruch w prawo. Zadanie polega na przetłumaczeniu zapisu ze starego kodu na nowy.

Wejście
W pierwszej i jedynej linii znajduje się ciąg oddzielonych pojedynczą spacją liczb całkowitych stanowiących zapis ruchu w starej wersji kodu. Długość ciągu nie przekracza `100` liczb.

Wyjście
Jedna linia zawierająca ciąg oddzielonych pojedynczą spacją liczb całkowitych stanowiących zapis ruchu w nowej wersji kodu.

Przykład
Przyjmijmy, że ciąg komend wpisywanych na działającej konsoli wygląda w następujący sposób:

```
L L P L L L P P P P L P
```

Wejście

```
2 5 5 5 5 6
```

Wyjście

```
1 1 2 3 5 1
```

Dobra Kadrowa

W pewnej dość dużej firmie pewna dość młoda Kadrowa otrzymała od swojej szefowej zadanie zebrania od wszystkich pracowników firmy informacji o odpowiadających im terminach urlopów. Zadanie proste, wręcz banalne. Kadrowa poprosiła wszystkich pracowników o niezbędne informacje. Jednak chcąc „przypodobać się" pracownikom poprosiła tylko o datę rozpoczęcia urlopu lub datę zakończenia urlopu (znała bowiem liczbę dni urlopu przypadającą pracownikom). Sama podjęła się zadania ustalenia brakującej daty zakończenia urlopu lub brakującej daty rozpoczęcia urlopu. Zadanie nadal nie jest trudne, ale trochę czasochłonne i narażające na pomyłkę. Dlatego owa Kadrowa prosi Ciebie o pomoc i napisanie programu wyznaczającego brakujące daty, uwzględniając czas urlopu przypadający pracownikom na różnych stanowiskach, wliczając w okres urlopu tylko dni pracujące (wszystkie soboty i wszystkie niedziele są wolne, a ponadto wszystkie świąteczne dni powszednie ustawowo wolne od pracy nie są wliczane do urlopu).
Program ma być uniwersalny w takim sensie, że ma służyć do wyliczenia urlopów nie tylko w roku 2018, ale też innych. Może się zdarzyć, że jakiś pracownik, któremu przysługuje np. 28 dni urlopu zażyczy rozpoczęcie swojego urlopu np. 15 grudnia 2019 roku.
Uwaga: w przypadku, gdy pracownik zażyczy rozpoczęcie urlopu w dniu wolnym od pracy jako pierwszy dzień urlopu należy policzyć pierwszy pracujący dzień następujący po dacie podanej przez pracownika. W przypadku, gdy pracownik zażyczy zakończenie urlopu w dniu wolnym od pracy jako ostatni dzień urlopu należy policzyć ostatni pracujący dzień występujący przed datą podaną przez pracownika.

Dane wejściowe
Jako dane wejściowe przyjmuje się trzy pliki tekstowe w formacie XML (załączone w przykładzie poniżej):

1) `stanowiska.xml` z listą stanowisk i przysługującą liczbą dni urlopu na danym stanowisku
2) `pracownicy.xml` z listą **1<=N<=100** pracowników i ich preferencjami (dla każdego pracownika musi wystąpić tylko jedna data)
3) `swieta.xml` z listą wszystkich świąt w roku 2018

Dla uproszczenia można przyjąć, że lista świąt z ich datami w roku 2018 obowiązuje także w innych latach.

Dane wyjściowe
Dla każdego pracownika z listy pracowników należy wypisać brakującą datę rozpoczęcia lub brakującą datę zakończenia urlopu w formacie DD/MM/RRRR. Zatem na wyjściu ma pojawić się **N** linii z odpowiednimi datami. Kolejność dat ma być zgodna z kolejnością pracowników w pliku `pracownicy.xml`.

Przykład

Dane wejściowe

Załączone poniżej treści plików `stanowiska.xml`, `pracownicy.xml` i `swieta.xml`.

Dane wyjściowe

```
08/04/2020
25/02/2019
```

Plik stanowiska.xml:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<stanowiska>
    <stanowisko nazwa="dyrektor">
        <dniUrlopu>40</dniUrlopu>
    </stanowisko>
    <stanowisko nazwa="kierownik">
        <dniUrlopu>36</dniUrlopu>
    </stanowisko>
    <stanowisko nazwa="magazynier">
        <dniUrlopu>28</dniUrlopu>
    </stanowisko>
</stanowiska>
```

Plik pracownicy.xml:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pracownicy>
    <pracownik>
        <nazwisko>Kowalski</nazwisko>
        <stanowisko>kierownik</stanowisko>
        <staz>21</staz>
        <urlop_od>
            <dzien>15</dzien>
            <miesiac>2</miesiac>
            <rok>2020</rok>
        </urlop_od>
    </pracownik>
    <pracownik>
        <nazwisko>Nowak</nazwisko>
        <stanowisko>magazynier</stanowisko>
        <staz>19</staz>
        <urlop_do>
            <dzien>7</dzien>
            <miesiac>4</miesiac>
            <rok>2019</rok>
        </urlop_do>
    </pracownik>
</pracownicy>
```

Plik swieta.xml:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<swieta>
    <swieto>
        <data>
            <dzien>1</dzien>
            <miesiac>1</miesiac>
```

```
        <rok>2018</rok>
    </data>
    <nazwa>Nowy Rok</nazwa>
</swieto>
<swieto>
    <data>
        <dzien>6</dzien>
        <miesiac>1</miesiac>
        <rok>2018</rok>
    </data>
    <nazwa>Trzech Kroli</nazwa>
</swieto>
<swieto>
    <data>
        <dzien>1</dzien>
        <miesiac>4</miesiac>
        <rok>2018</rok>
    </data>
    <nazwa>Wielkanoc</nazwa>
</swieto>
<swieto>
    <data>
        <dzien>2</dzien>
        <miesiac>4</miesiac>
        <rok>2018</rok>
    </data>
    <nazwa>Wielkanoc</nazwa>
</swieto>
<swieto>
    <data>
        <dzien>1</dzien>
        <miesiac>5</miesiac>
        <rok>2018</rok>
    </data>
    <nazwa>Swieto Pracy</nazwa>
</swieto>
<swieto>
    <data>
        <dzien>3</dzien>
        <miesiac>5</miesiac>
        <rok>2018</rok>
    </data>
    <nazwa>Swieto Konstytucji 3 Maja</nazwa>
</swieto>
<swieto>
    <data>
        <dzien>20</dzien>
        <miesiac>5</miesiac>
        <rok>2018</rok>
    </data>
    <nazwa>Zielone Swiatki</nazwa>
</swieto>
<swieto>
```

```
        <data>
            <dzien>31</dzien>
            <miesiac>5</miesiac>
            <rok>2018</rok>
        </data>
        <nazwa>Boze Cialo</nazwa>
    </swieto>
    <swieto>
        <data>
            <dzien>15</dzien>
            <miesiac>8</miesiac>
            <rok>2018</rok>
        </data>
        <nazwa>Wniebowziecie Najswietszej Maryi Panny</nazwa>
    </swieto>
    <swieto>
        <data>
            <dzien>1</dzien>
            <miesiac>11</miesiac>
            <rok>2018</rok>
        </data>
        <nazwa>Wszystkich Swietych</nazwa>
    </swieto>
    <swieto>
        <data>
            <dzien>11</dzien>
            <miesiac>11</miesiac>
            <rok>2018</rok>
        </data>
        <nazwa>Swieto Niepodleglosci</nazwa>
    </swieto>
    <swieto>
        <data>
            <dzien>25</dzien>
            <miesiac>12</miesiac>
            <rok>2018</rok>
        </data>
        <nazwa>Pierwszy dzien Bozego Narodzenia</nazwa>
    </swieto>
    <swieto>
        <data>
            <dzien>26</dzien>
            <miesiac>12</miesiac>
            <rok>2018</rok>
        </data>
        <nazwa>Drugi dzien Bozego Narodzenia</nazwa>
    </swieto>
</swieta>
```

Antarktyda pod Palmami

Janusz, zapalony turysta i jednocześnie mniej zapalony informatyk w biurze podróży Antarktyda pod Palmami sprzedającym wycieczki do wszystkich krajów świata, a nawet dalej, ma bardzo nieufną szefową, panią Karinę. Szefowa wszędzie widzi osoby próbujące ją oszukać, nie ufa nawet swojemu informatykowi, jednocześnie małżonkowi Januszowi, a co dopiero pozostałym pracownikom. Pani Karina zleciła Januszowi napisanie skryptu, który będzie przetwarzał pliki ze zdjęciami z kamer w firmie. Zdjęcia są ukryte w podkatalogu `ISeeYou` katalogu domowego komputera firmowego, ale pani Karina nie chce przechowywać ich tam zbyt długo. Pliki mają nazwy `sweetfocia-yyyy-mm-dd-HH-MM-SS.jpg` (yyyy-mm-dd-HH-MM-SS – dokładny czas wykonania zdjęcia). Pani Karina zażyczyła sobie, by pliki zdjęć były łączone ze sobą w bardzo prosty sposób – powinien być tworzony nowy plik w tym samym katalogu o nazwie `fotos-yyyy-mm-dd-HH-MM-SS.foto` (yyyy-mm-dd-HH-MM-SS – dokładny czas zapisania nowego pliku) jako konkatenacja wszystkich plików zdjęć dostępnych aktualnie w katalogu. Uwaga – połączone mają być tylko pliki zdjęć o nazwach i rozszerzeniach zgodnych z powyższym formatem. Inne pliki powinny być pomijane. Następnie rozmiar każdego pliku oryginalnego wyrażony w bajtach należy dopisać do pliku `kontrolne.txt` umieszczonym w katalogu domowym komputera (jeżeli plik nie istnieje – należy go utworzyć) w takiej samej kolejności, w jakiej pliki były dołączane do pliku wynikowego (każdy rozmiar w nowej linii pliku). Oryginalne pliki zdjęć należy usunąć i ostatecznie obliczyć sumę kontrolną sha1 na nowo utworzonym pliku i jej wartość dopisać na końcu pliku `kontrolne.txt`. Taki skrypt powinien być automatycznie wykonywany co 5 minut.

UWAGA:
1. Maszyna wirtualna do zadania do pobrania ze strony RKI (login: rki, hasło: rki2018)
2. Proszę używać tylko poleceń, które są osiągalne na maszynie – nie doinstalowywać niczego
3. Polecenia wykonywać z uprawnieniami zwykłego użytkownika
4. Jeżeli w katalogu pomiędzy wywołaniami skryptu nie pojawią się żadne nowe pliki zdjęć – skrypt nie powinien uruchamiać się

Do testów przygotowany jest podkatalog `ISeeYou`, w którym znajdują się przykładowe pliki.

Rozwiązanie, które należy przesłać: plik skryptu, w którym należy dodać w postaci komentarza informację, w jaki sposób automatycznie wywoływać skrypt co 5 minut.

Szkolenie szkoły

Nowoczesne technologie dawno już dotarły pod strzechy, jednak pod betonowe sufity szkół trafiają powoli, i jakby z oporem. Niestety, informatyk w szkole w Łąkołodach nie należy do osób lubiących nadążać za nowościami. A dyrekcja szkoły wprowadziła właśnie nowy system bazodanowy do obsługi sekretariatu. Informatyk dostał proste zadanie – przygotować zapytania do bazy danych, które pozwolą na odczytanie z niej niezbędnych informacji. Męczył się człowiek niemiłosiernie, ale nawet wujek google nie chciał mu ostatecznie pomóc (a może informatyk nie wiedział tak naprawdę o co pytać?). Pomóż biedakowi, bo do końca straci wiarę w nowoczesne technologie.
Czego dotyczą zapytania? Niezbyt długa lista poniżej. Pamiętaj, że wszystkie (oprócz 4, 8 i 10) mają być wyświetlone w czytelny sposób (metoda pretty()).
1. Wyświetl wszystkie informacje o uczennicach, które mają na imię Anna. Wynik posortuj rosnąco według nazwiska.
2. Wyświetl wszystkie informacje o uczniach, którzy mają na imię Jan i nazwisko Kowalski.
3. Wyświetl wszystkie informacje o uczniach, których nazwisko to Kowalski lub Nowak. Wynik posortuj malejąco według imienia.
4. Wyświetl liczbę uczennic mających na imię Anna.
5. Wyświetl wszystkie informacje o uczniach, którzy mieszkają na ulicy Lipowej.
6. Wyświetl wszystkie informacje o uczniach, którzy urodzili się po 15 stycznia 2001 roku.
7. Wyświetl same nazwiska uczennic mających na imię Anna.
8. Wyświetl średnią wzrostu wszystkich uczniów. Zaprezentuj wynik w kluczu srednia (np { "srednia" : 167.5 }).
9. Wyświetl uczniów, którzy interesują się warcabami.
10. Wyświetl same imiona i nazwiska uczniów, którzy w adresie nie mają numeru mieszkania.

UWAGA:
1. Maszyna wirtualna do zadania do pobrania ze strony RKI (login: rki, hasło: rki2018)
2.Baza danych MongoDB jest zainstalowana, załączone są też przykładowe dane do testów (kolekcja uczniowie)
3. Model pojęciowy bazy:

Rozwiązanie, które należy przesłać: plik tekstowy, w którym znajdą się w kolejnych liniach utworzone zapytania.

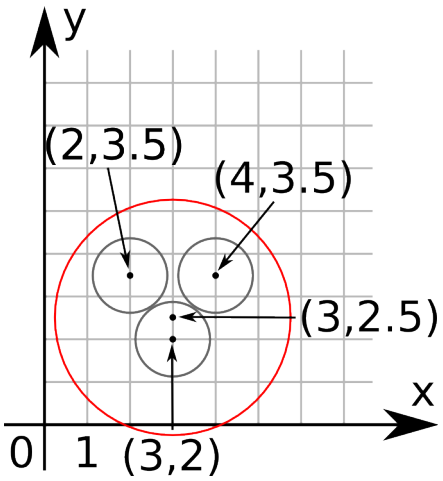
Ale jaja!

Pewien znamienity naukowiec Eggar Basket układając pisanki w koszyczku postanowił zbadać, jaka jest największa wielkość jajek, które mogłyby się w tym koszyku zmieścić. Oczywiście nie byłby prawdziwym naukowcem, gdyby nie spróbował rozwiązać tego problemu w ogólnym przypadku n-wymiarowej przestrzeni. Dla ułatwienia przyjął jednak, że jajka i koszyk mają kształt sferyczny, wszystkie jajka są jednakowe a proces ich translokacji odbywa się w próżni. Twoim zadaniem jest stworzenie programu, który pomoże mu zweryfikować rozwiązania uzyskane w przypadku dwuwymiarowym.

Wejście
W pierwszej linii znajdują się trzy liczby rzeczywiste oddzielone spacją oznaczające odpowiednio współrzędną **X**, współrzędną **Y** oraz promień **R** „koszyka" na płaszczyźnie. Kolejne linie zawierają oddzielone spacją pary współrzędnych **X** i **Y** środków „jajek" znajdujących się wewnątrz „koszyka".

Wyjście
Jedna linia określająca wartość największego możliwego promienia „jajek" mieszczących się w „koszyku" z dokładnością do 4 miejsc po przecinku.

Przykład



Wejście
```
3.0 2.5 2.75
2.0 3.5
3.0 2.0
4.0 3.5
```

Wyjście
```
0.9014
```