

# Programowanie Aplikacji Webowych

## laboratorium 6

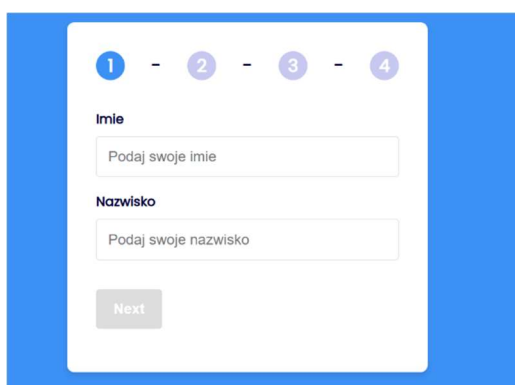
### Cel zajęć:

Celem laboratorium jest przećwiczenie bardziej zaawansowanych zagadnień związanych z programowaniem z wykorzystaniem JS.

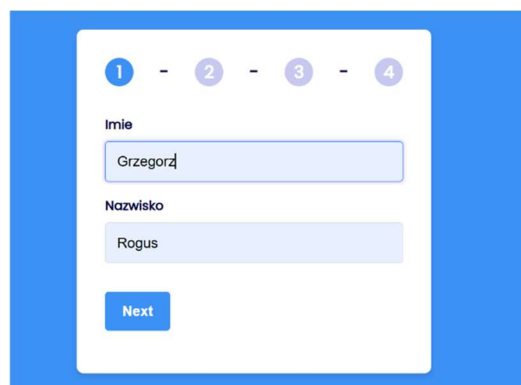
**Zadanie 1.** Stwórz formularz typu multi-step.

**(2pkt)**

Poszczególne ekrany pokazane na poniższych zdjęciach.



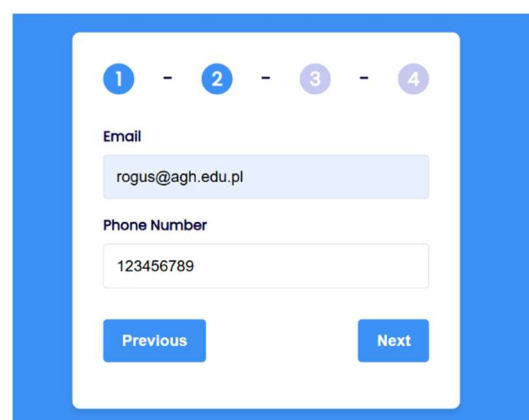
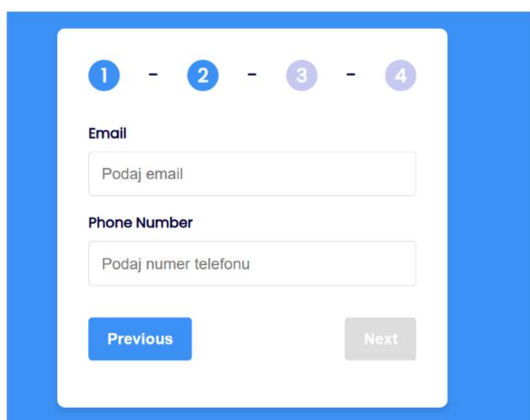
Ekran startowy



Ekran poprawnie wypełniony

Niech numer poprawnie wypełnionych ekranów (wraz z aktualnie wypełnianym) będzie zaznaczony na niebiesko.

Przejsięcie do kolejnego ekranu jest możliwe tylko gdy poszczególne pola są poprawnie wypełnione (nie mogą być puste a dane wprowadzone są w poprawnym formacie – wymaganym przez dany typ pola)



1 - 2 - 3 - 4

Hasło

Podaj hasło

Data urodzenia

dd.mm.rrrr

Previous

Next

1 - 2 - 3 - 4

Adres

Podaj swój adres

PESEL

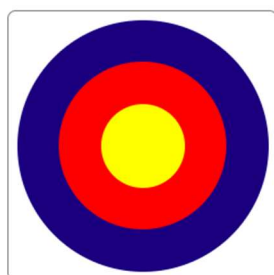
Podaj swój PESEL

Previous

Submit

Po wprowadzeniu danych w ostatnim ekranie i naciśnięciu przycisku Submit niech wyświetli się ekran zawierający podsumowanie wysłanych danych.

**Zadanie 2.** Stwórz app typu tarcza strzelnicza jak na rysunku poniżej. ( 2 pkt )



16

RESETUJ

PROPAGACJA AKTYWNA

- ☒ Wywołanie 1-2-5
- ☐ Wywołanie 1-5-2
- ☐ Wywołanie 2-5-1
- ☐ Wywołanie 5-2-1

#### LOGI

Naciśnięto żółty o wartości 5  
Naciśnięto czerwony o wartości 2  
Naciśnięto niebieski o wartości 1  
Naciśnięto żółty o wartości 5  
Naciśnięto czerwony o wartości 2  
Naciśnięto niebieski o wartości 1

Niech strzał w dowolne miejsce na tarczy (naciśnięcie któregośkolwiek z nich) dodaje odpowiednią liczbę punktów do aktualnego licznika punktów oraz wyświetla odpowiedni komunikat w dedykowanej sekcji na stronie.

Wartość punktowa naciśniętych tarcz to:

Żółtej 5 pkt – komunikat „naciśnąłeś żółty o wartości 5”

Czerwonego 2 pkt komunikat – „naciśnąłeś czerwony o wartości 2”

Niebieski 1 pkt komunikat – „naciśnąłeś niebieski o wartości 1”

Jeśli suma naciśniętych wartości przekroczy 20 należy wyłączyć możliwość obsługi zdarzenia click przez obiekt 3 (żółty). Jeśli 30 usuwamy dodatkowo obsługę zdarzenia przez obiekt 2 (żółty). Powyżej 50 pkt strzelnica nie obsługuje już zdarzeń strzeleckich.

Niech na stronie będą jeszcze 4 przyciski: Stop/StartPropagation, Reset, Zmiana kolejności wywoływania zdarzeń., ClearLog.

Stop/StartPropagation – włącza i wyłącza propagację (niech tekst na przycisku odpowiada stanowi przycisku)

Reset – wraca do stanu początkowego – reset licznika punktów oraz możliwość obsługi zdarzeń typu click, kolejności wywoływania zdarzeń standardowa

Naciśnięcie zmiany kolejności – zmienia kolejność wywoływania zdarzeń.

Clear LOG – czyszczenie ekranu z logami

### **Zadanie 3. ( 2pkt)**

W katalogu Zadanie 3 znajdziesz plik city.json zawierający kolekcję miast Polski.

Ze względu na zbliżającą się katastrofę demograficzną chcemy zobaczyć jak będzie wyglądała przyszłość w roku 2050. Przyjmujemy następujące założenia:

1. Miasta powyżej 100 tys. z województw małopolskiego, pomorskiego, mazowieckiego, dolnośląskiego oraz wielkopolskiego będą zwiększać populację o 2% rocznie.
2. Pozostałe miasta powyżej 100tys zwiększa swoją populację o 1% rocznie.
3. Miasta o liczbie ludności pomiędzy 10 tys a 100tys będą tracić średnio 650 mieszkańców rocznie.
4. Małe miasta ( do 10 tys.) tzw ściany wschodniej + świętokrzyskie tracić będą około 5% populacji co 7 lat.
5. Populacja pozostałych małych miast w Polsce będzie zmieniać się losowo w przedziale +/- 30 osób rocznie.
6. Miasta o średniej gęstości zaludnienia powyżej 1000 mieszkańców/km2 w 2035 roku zwiększa swoją powierzchnię o 10%.

a). Wyświetl na stronie sumaryczną wartość mieszkańców wszystkich miast Polski w roku 2050.

b). wyświetli 5 najludniejszych (pod kątem mieszkańców miast) województw w 2050 roku z możliwością rozwinięcia każdego z województw tak aby wyświetlić listę i statystykę każdego miasta w województwie.

c). wyświetl wszystkie małe miasta posortowane alfabetycznie które będą mniejsze w 2050 niż są aktualnie.

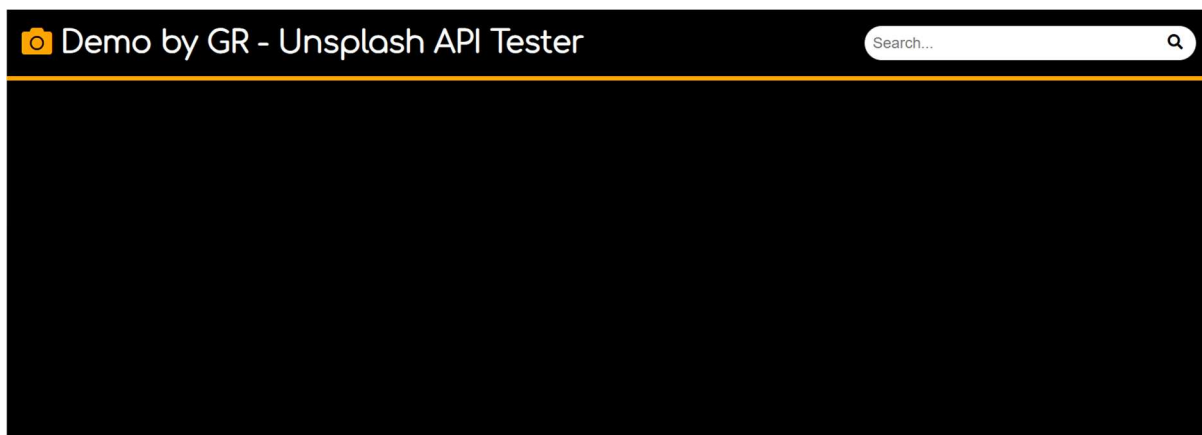
f). jaka będzie średnia gęstość zaludnienia w dużych miastach (powyżej 100 tys) w 2050 roku w porównaniu do aktualnej wartości.

Wyniki wyświetlić na dedykowanych sekcjach na stronie

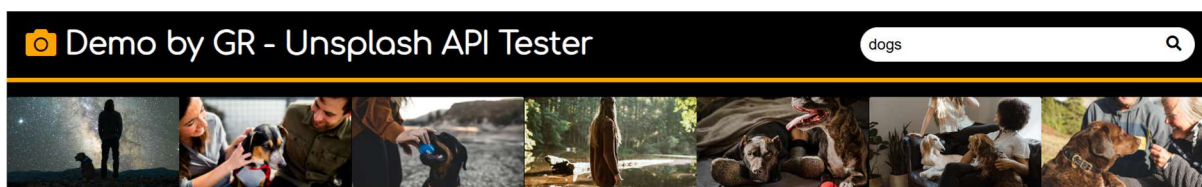
**Zadanie 4. (3pkt )** Pod adresem URL <https://api.unsplash.com> znajduje się backend w chmurze, który zwraca zdjęcia powiązane z podanym tematem kryterialnym. Sposób użycia:

„https://api.unsplash.com/search/photos?query='+input.value+'&per\_page=30&client\_id=xxxxxxxxxx”

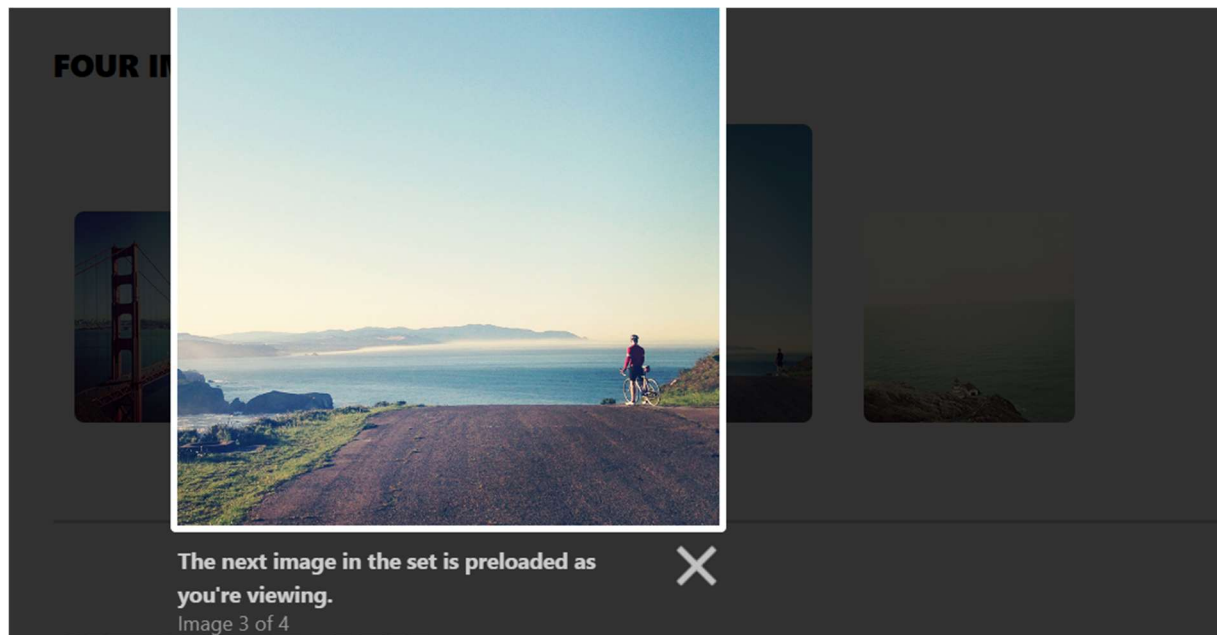
Stwórz prostą wyszukiwarkę, która pozwoli Ci przysyłać zdjęcia o tematyce określonej w polu wejściowym wyszukiwarki. Przykładowa strona jest pokazana poniżej. (Dobór kolorów pozostawiam do Państwa decyzji)



Po podaniu dowolnego słowa kluczowego aplikacja powinna zwrócić listę zdjęć, które wyświetlone mają być na ekranie jako miniaturki w jednym wierszu. Niech wyświetla się np. 8 zdjęć z wierszu przy normalnej rozdzielczości.. Z lewej i prawej strony aplikacji mamy przyciski < i > , których naciśnięcie powinno powodować przesunięcie paska ze zdjęciami w lewo lub w prawo. Oznaczać to będzie wyświetlanie 7 takich samych zdjęć ja poprzednio + 8 nowe ( z lewej lub prawej strony w zależności które przycisk został naciśnięty. W przypadku gdy naciskając > dojdiesz do ostatniego zdjęcia załaduj kolejną porcję zdjęć z serwera.



Gdy naciśniesz dwukrotnie na zdjęcie ma się wyświetlać jako duże zdjęcie na pierwszym planie częściowo przysłaniając pozostałe. Chodzi mi tutaj o efekt znany np. z bibliotek typu LightBox2. ( patrz poniżej). Z tyłu zdjęcia widzimy przyciemnioną zawartość strony. Kliknięcie na przycisk stop na zdjęciu lub na tło zamyka okno i wracamy do domyślnego dokumentu.



**Zadanie 5.** Stwórz aplikację do oceny zgodności dwóch tekstów. Sposób użycie pokazany poniżej ( 2pkt)

## Weryfikacja zgodności tekstów

Wprowadz pierwszy tekst :

Ala ma kotka a kotek Ale

Wprowadz drugi tekst:

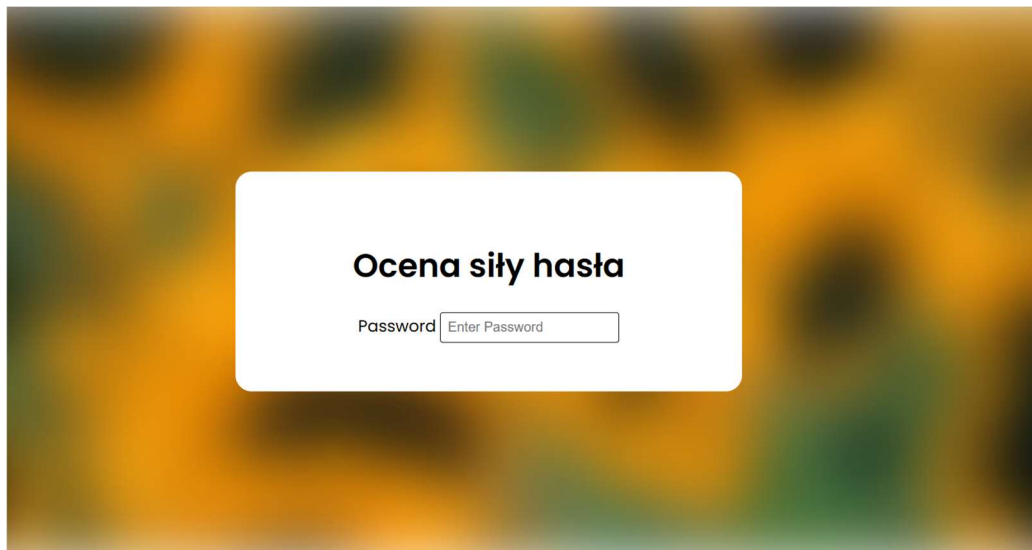
Ma kotka a nie psa

oblicz podobieństwo

Podobieństwo: 22.22%

**Zadanie 6.** Zaimplementuj formularz oceny hasła, który pozwala na ocenę siły hasła. **(2pkt )**  
Formularz wygląda jak poniżej: ( zwróć uwagę na zdjęcie w tle)

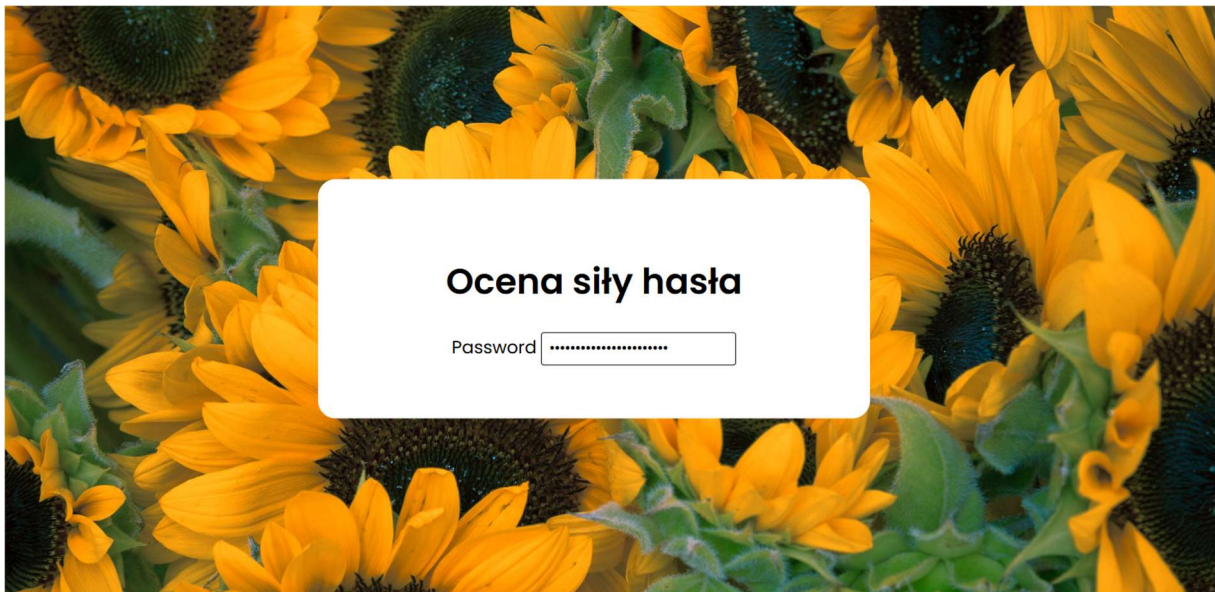
Brak hasła lub hasło bardzo słabe:



Hasło srednie:



Hasło silne:



Poniżej powinna znaleźć się lista wskazówek co do jakości hasła. Spełnienie któregoś z warunków skutkuje zmianą jego statusu (zmiana ikony przy warunku)

Password requirements:

- ✗ at least 8 characters
- ✗ at least 1 digit
- ✗ at least 1 capital letter
- ✗ at least 1 special character

Password requirements:

- ✗ at least 8 characters
- ✓ at least 1 digit
- ✗ at least 1 capital letter
- ✓ at least 1 special character

Niech lista warunków zostanie jeszcze rozszerzona o następujące warunki:

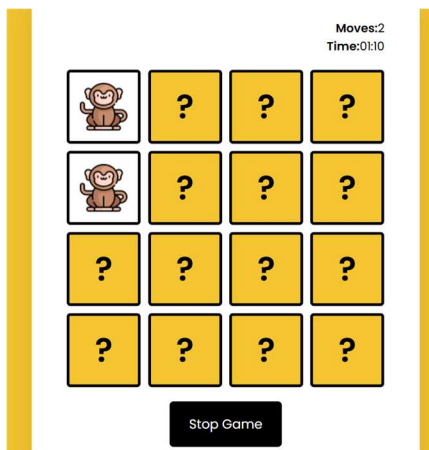
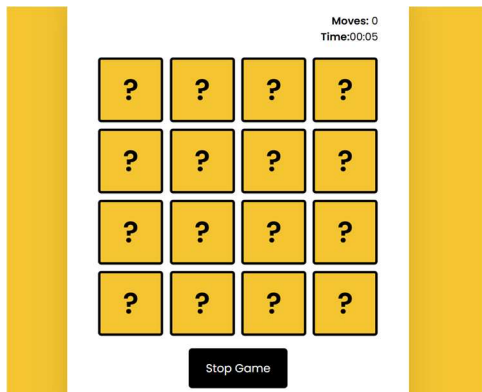
- Kolejne 3 znaki nie mogą być znakami powtarzającymi się – czyli zabronione jest hasło aaa lub 111
- Kolejne znaki nie mogą być znakami występującymi po sobie w alfabecie lub w jako wartości numeryczne - czyli abcd lub 1234 są niepoprawne



Dodaj również możliwość podglądu aktualnie wprowadzanego hasła.

### Zadanie 7. Gra memory ( 3 punktów)

Zaimplementuj grę typu memory. Gra polega na odkrywaniu rewersów dwóch kart. Jeśli odkryte karty są takie same – karty zostają odkryte, jeśli są różne karty wracają do stanu zakrytego. Gra powinna być konfigurowalna to oznacza że wybierasz na początku rozmiar planszy oraz np. typ kart użytych do rozgrywki (niech będą to np. 3 różne typy kart: np. zwierzęta, flagi, samochody itp.). Pod ustaleniami konfiguracyjnych zaczynasz grę:



Po zakończeniu gry appp powinna poprosić o Twój nick aby można go było wraz z ilością ruchów oraz czasem gry zapisać do listy rankingowej.

Można do tego celu użyć:

// highscore json blob: [https://jsonblob.com/\\_WSTAW\\_SWOJ\\_HASH\\_JSON](https://jsonblob.com/_WSTAW_SWOJ_HASH_JSON)")

App zapisuje Twój wynik do listy rankingowej. 10 najlepszych wyników jest wyświetlonych na ekranie po zakończeniu gry.

Oceniam również jakość kodu.

### Zad 8. Dynamiczna tablica (4pkt)

Stworzyć tablice zasilaną danymi zewnętrznymi w formacie json. Tablica powinna umożliwić filtrowanie danych oraz jej sortowanie – po każdej wskazanej kolumnie. Możliwe jest jednocześnie filtrowanie kilku kolumn ( 1pkt )



Lista kolumn ustalana dynamicznie na podstawie formatu json. Tablica na jednym ekranie wyświetla tylko ustaloną liczbę pozycji. Wymagana jest więc obsługa paginacji. Tablica ma być responsywna a z możliwością blokowania szerokości wybranych kolumn ( **1pkt** )

Pod adresem <https://restcountries.com> znajdziesz listę wszystkich krajów swata. Wykorzystując znajdujące się tam dane zasil nimi tablice.

Wyświetlane powinny być następujące dane: Name, Capital, population, area oraz 2 dodatkowe atrybuty wybrane przez użytkownika z listy rozwijalnej prezentującej wszystkie dostępne atrybuty.

Niech kraje będą grupowane pod kątem Subregionu do którego należą. Na poziomie subregionu wyświetlane są tylko nazwa subregionu, sumaryczna ilość ludzi zamieszkujący ten subregion oraz całkowita jego powierzchnia. Oczywiście jest możliwe rozwijanie subregionu aby zobaczyć poszczególne kraje należące do niego. Wizualnie to może wyglądać tak jak przykład poniżej ( uwaga! to jest tylko propozycja, Państwo możecie to zaimplementować w zupełnie inny sposób).

( **1pkt** )

Oceniana jest również jakość kodu ( **1pkt** )

1 2 3 4 5				
Name ▼		Capital ▼	Population ▼	
Western Africa			401908376	61
Eastern Asia			1640439098	11
Western Asia			278388391	48
Caribbean			43851771	23
Eastern Africa			421051358	59

1 2 3 4 5				
Name ▼		Capital ▼	Population ▼	
Western Africa			401908376	
Eastern Asia			1640439098	
Hong Kong Special Administrative Region of the People's Republic of China		City of Victoria	7500700	
People's Republic of China		Beijing	1402112000	
Republic of China (Taiwan)		Taipei	23503349	
Democratic People's Republic of Korea		Pyongyang	25778815	
Macao Special Administrative Region of the People's Republic of China		undefined	649342	
Japan		Tokyo	125836021	
Republic of Korea		Seoul	51780579	
Mongolia		Ulan Bator	3278292	
Western Asia			278388391	

