# PPJ

ćwiczenia 8
zaoczne

20 Listopada 2016

I. Utwórz dwuwymiarową tablicę o rozmiarach $8 \times 8$ i wypełnij ją losowymi wartościami z przedziału od 0 do 10. Następnie sprawdz czy na przekątnych tej tablicy (łącznie) występują przynajmniej 3 takie same liczby.

II. Dane są dwie trzyelementowe tablice zmiennych typu `double` wypełnione losowymi liczbami z przedziału 0.0 do 5.0. Napisz program, w wyniku którego powstanie trzecia trzyelementowa tablica wyliczana zgodnie z zasadą, że pierwszy element wynikowej tablicy jest wynikiem sumy pierwszego elementu z pierwszej tablicy i ostatniego elementu z drugiej tablicy, drugi element tablicy wynikowej jest wynikiem sumy drugiego elementu z pierwszej tablicy i drugiego od końca elementu drugiej tablicy itd itd.

III. Dana jest następująca tablica znaków:

```
1  char [][] tab = {
2    {'S','a','m','s','u','n','g'},
3    {'N','o','k','i','a'},
4    {'A','p','p','l','e'},
5    {'B','l','a','c','k','B','e','r','r','y'},
6    {'A','l','c','a','t','e','l'},
7    {'S','o','n','y'},
8    {'J','o','l','l','a'}
9  };
```

Utwórz program sprawdzający, które z nazw umieszczonych w powyższej tablicy:

- zawierają przynajmniej dwie takie same litery,
- suma wszystkich znaków w słowie jest większa niż 255,
- zawiera przynajmniej jeden znak 'i',
- zawiera przynajmniej dwa takie same znaki.

Następnie wypisz nazwy spełniające przynajmniej 3 z powyższych warunków.

IV. Utwórz tablicę o losowej długości i wypełnij ją losowymi elementami. Następnie utwórz program, który utworzy tablicę o rozmiarze dwa razy większym i wypełni jej pierwszą połowę wartościami z pierwszej tablicy posortowanymi w porządku rosnącym, a następnie drugą połowę wypełni tymi samymi wartościami uporządkowanymi malejąco.

V. Utwórz dwuwymiarową tablicę zmiennych typu `int` o rozmiarach 10 na 10 i wypełnij ją losowymi liczbami. Następnie posortuj rosnąco każdy wiersz tej tablicy.

VI. Dana jest poniższy fragment kodu:

```
1  char [][] tab = {
2  {o----------},
3  {xxxxxxxxx-},
4  {---------x-},
5  {-xxxxxx-x-},
6  {-xe-----x-},
7  {-xxxxxxxx-},
8  {----------}
9  };
10
11 int pozycjaX = 0, pozycjaY = 0;
```

Zmienne `pozycjaX` i `pozycjaY` określają położenie `mrówki`, która w drodze do znaku `'e'` może poruszać się tylko po znakach `'-'`.

Napisz program, który przeprowadzi `mrówkę` zapewniając że każda pozycja `'-'` zostanie odwiedzona dokładnie jeden raz.

VII. Przygtuj bezrezultatową metodę `myMethod` z argumentem typu `int`, której zadaniem będzie zwiększenie argumentu i wypisanie rezultatu. Następnie przygtuj bezrezultatową metodę `myMethod` z argumentem typu `double`, której zadaniem będzie zmniejszenie argumentu i wypisanie rezultatu.

Przedstaw przykład programu wywołującego obie metody, gdy do dyspozycji mamy tylko jedną zainicjowaną zmienną typu `char`.

VIII. Zaimplementuj metodę sprawdzającą czy dostarczona jako argument tablica znaków typu `char` jest palindromem. Rezultat operacji zwróć jako wartość typu logicznego `boolean`. Poprawność działania przetestuj na przykładach.

IX. Utwórz rekurencyjną metodę obliczającą ciąg Fibonacciego, zdefiniowany dla elementu *fibonacci(n)* jako sumę *fibonacci(n-1) + fibonacci(n-2)* przy założeniu, że *fibonacci(1)* i *fibonacci(2)* mają odpowiednio wartości 1 i 2

X. Utwórz program tworzący wszystkie możliwe permutacje znaków z dostarczonej tablicy znaków.