

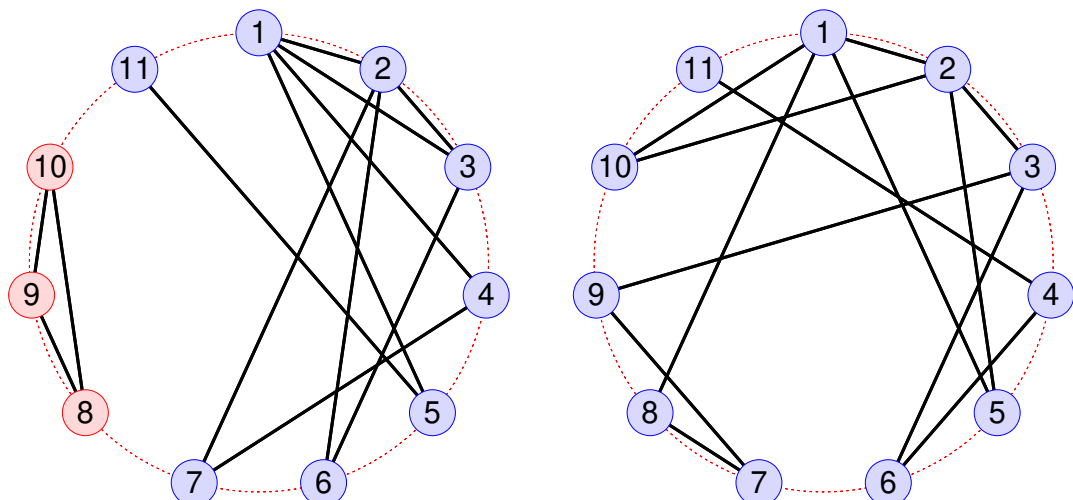
## PRZYKŁADOWE WEJŚCIE – ZESTAW 2.

*Uwaga: jeśli program będzie generował wizualizację grafu, to taka wizualizacja może być dowolna – nie ma już przymusu reprezentacji na okręgu.*

### Ad. 1: Ciąg graficzny

**Dane wejściowe:** sekwencja liczb naturalnych.

**Wyjście programu:** odpowiedź, czy podany ciąg jest ciągiem graficznym. Jeśli tak, to wynikiem powinien być również graf zbudowany według zadanego ciągu graficznego, przedstawiony w dowolnej postaci (np. jedna z reprezentacji: lista sąsiedztwa/macierz sąsiedztwa/macierz incydencji lub obrazek z wizualizacją, jak np. na rys. 1(a)).



(a) Efekt dla ciągu 4 2 2 3 2 1 4 2 2 2 2, który jest graficzny (zad. 1)      (b) Efekt po dziesięciu randomizacjach (zad. 2)

Rysunek 1: Dane **wyjściowe**: (a) wizualizacja grafu dla ciągu graficznego 4 2 2 3 2 1 4 2 2 2 2; (b) graf z rys. (a) po dziesięciokrotnej randomizacji krawędzi

Przykład ciągu, który nie jest graficzny: 4 4 3 1 2.

### Ad. 2: Randomizacja krawędzi

**Dane wejściowe:** liczba żądanych randomizacji oraz graf w dowolnej postaci (np. ciągu graficznego).

**Wyjście programu:** wynikowy graf przedstawiony w dowolnej postaci, np. jak na rys. 1(b)).

*Zadanie 2. można połączyć z 1. zadaniem.*

### Ad. 3: Oznaczanie spójnych składowych/największa spójna składowa

**Dane wejściowe:** graf w dowolnej postaci (np. ciągu graficznego).

**Wyjście programu:** dowolny sposób, np. na rys. 1(a), 1(b) różne kolory wierzchołków oznaczają różne spójne składowe, natomiast listing 1. wypisuje wierzchołki należące do poszczególnych składowych.

```

1 1) 1 2 3 4 5 6 7 11
2 2) 8 9 10
3 Największa składowa ma numer 1.

```

Listing 1: Dane **wyjściowe** w formie: “numer\_składowej) numery\_wierzchołków” wraz ze wskazaniem, która składowa jest największa; wejściem przykładu był graf z rys. 1(a)

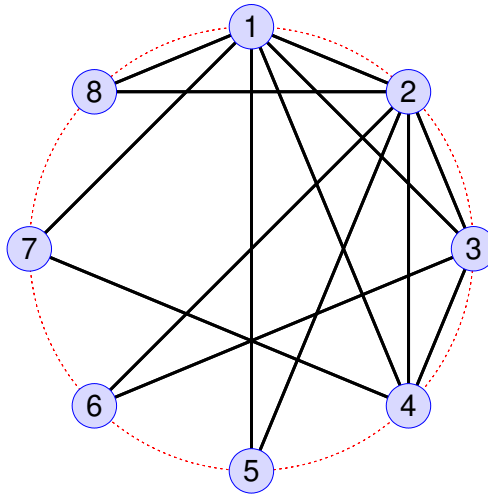
*Zadanie 3. można połączyć z 1. zadaniem.*

#### Ad. 4: Graf eulerowski, poszukiwanie cyklu Eulera

**Dane wejściowe:** liczba wierzchołków  $n$  lub nic (jeśli  $n$  będzie również losowane).

**Wyjście programu:** cykl Eulera w dowolnej formie (np. wypisanie poszczególnych wierzchołków).

**Przykład:** dla  $n = 8 \Rightarrow$  wylosowano stopnie 4 2 6 2 6 2 4 2, dla których wygenerowano losowy graf wejściowy z rys. 2. Znaleziony cykl Eulera: listing 2.



Rysunek 2: Wygenerowany przykładowy graf eulerowski (po stu randomizacjach)

```

1 [1 - 2 - 3 - 1 - 4 - 2 - 5 - 1 - 7 - 4 - 3 - 6 - 2 - 8 - 1]

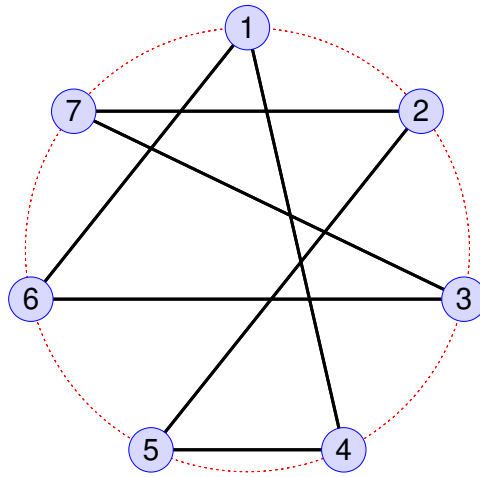
```

Listing 2: Dane **wyjściowe** – wypisano kolejne wierzchołki cyklu Eulera

#### Ad. 5: Grafy $k$ -regularne

**Dane wejściowe:** liczba wierzchołków  $n$  oraz stopień wierzchołków  $k$ .

**Wyjście programu:** wynikowy graf w dowolnej reprezentacji, np. wizualizacja (jak w przykładzie z rys. 3.).



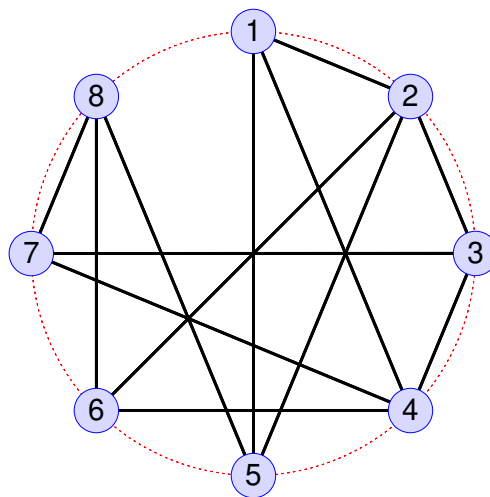
Rysunek 3: Wygenerowany losowy graf  $k$ -regularny dla parametrów wejściowych:  $n = 7$ ,  $k = 2$  (po stu randomizacjach)

#### Ad. 6: Graf hamiltonowski, poszukiwanie cyklu Hamiltona

**Dane wejściowe:** dowolne (graf może być zadany przy pomocy wejściowej reprezentacji lub losowany).

**Wyjście programu:** informacja, czy graf jest hamiltonowski; jeśli tak: cykl Hamiltona w dowolnej formie (np. wypisanie poszczególnych wierzchołków).

**Przykład:** dla grafu wejściowego z rys. 4. znaleziono cykl Hamiltona z listingu 3.



Rysunek 4: Przykładowy wejściowy graf

```
1 [1 - 2 - 3 - 7 - 4 - 6 - 8 - 5 - 1]
```

Listing 3: Dane **wyjściowe** – wypisano kolejne wierzchołki cyklu Hamiltona