

Metody projektowania algorytmów

Zadanie 1

Napisz program, który znajdzie takie ułożenie n królowych na szachownicy o wymiarach $n \times n$, że żadne dwie królowe nie będą się wzajemnie atakowały. Wypisz takie ułożenie na ekranie w formie graficznej, gdzie znakiem `.` oznaczysz puste pole szachownicy, a znakiem `Q` miejsce zajmowane przez królową.

Wejście

- n - liczba naturalna, $3 \leq n \leq 15$

Przykład

Wejście

```
4
```

Wyjście

```
..Q.  
Q...  
...Q  
.Q..
```

Zadanie 2

Dla zadanego zbioru liczb całkowitych, znajdź **najmniejszy** możliwy iloczyn jaki można uzyskać przemnażając przez siebie pewien niepusty podzbiór elementów tego zbioru.

Wejście

- n - liczba naturalna, $n > 0$
- n liczb całkowitych, po jednej w każdej linii

Przykład

Wejście

```
7  
-1  
-1  
-2  
-1  
4  
0  
3
```

Wyjście

-24

Wyjaśnienie: minimalny iloczyn uzyskamy przemnażając przez siebie $(-2) * (-1) * (-1) * 4 * 3$

Zadanie 3

Na stole znajduje się n monet. Dwóch graczy wykonuje swoje ruchy naprzemiennie. W każdym ruchu gracz może zabrać ze stołu 1, 2 lub 5 monet. Zwycięża gracz, który jako ostatni zabierze monety ze stołu.

Napisz program, który dla zadanej początkowej liczby monet n określi, czy pierwszy gracz może **zawsze** wygrać dane ułożenie, bez względu na to, jakie ruchy będzie wykonywać drugi gracz.

Wejście

- n - liczba naturalna, $n > 0$

Przykład 1

Wejście

4

Wyjście

TAK

Przykład 2

Wejście

3

Wyjście

NIE

Zadanie 4

Napisz program, który dla podanej w złotych polskich kwoty poda minimalną liczbę monet i/lub banknotów potrzebną do uzyskania tej kwoty, a także wypisze takie zestawienie na ekranie. Zakładamy, że każdego nominału możemy użyć dowolną liczbę razy.

Wejście

- n - liczba naturalna, $n > 0$

Przykład

Wejście

```
48
```

Wyjście

```
5  
20 20 5 2 1
```

Zadanie 5

Napisz program, który dla podanej kwoty oraz zestawu nominałów pod minimalną liczbę monet/banknotów potrzebną do uzyskania tej kwoty przy wykorzystaniu dostępnych nominałów, a także wypisze zestawienie na ekranie. Zakładamy, że każdego nominału możemy użyć dowolną liczbę razy.

Jeżeli podanej kwoty nie da się wydać na pomocą podanego zestawu nominałów, program powinien wypisać stosowny komunikat.

Wejście

- k - kwota do wydania, liczba naturalna, $k > 0$,
- n - liczba nominałów, liczba naturalna, $n > 0$,
- n liczb naturalnych większych od zera, wartości nominałów

Przykład

Wejście

```
14  
3  
1 7 10
```

Wyjście

```
2  
7 7
```