

Zadanie 1 - Napisz program, który przygotuje i zwizualizuje drzewo decyzyjne dla problemu n hetmanów rozwiązywanego metodą nawrotów (jak na WDI). Program powinien kolejno:

przeczytać liczbę hetmanów z jedyne go, obowiązkowego argumentu wywołania;
uruchomić funkcję, która znajdzie wszystkie rozwiązania problemu, i której efektem działania będzie wyłącznie stworzenie w pamięci drzewa decyzyjnego;
uruchomić funkcję, która wydrukuje to drzewo, przekazane jako argument (szczegóły poniżej);

jawnie zwolnić całą zarezerwowaną pamięć i zakończyć.

Wydruk w wersji uproszczonej (do 12 pkt.) dla $n = 3$ powinien wyglądać tak:

```
Kolumna 0 wiersz 0 nie jest atakowany, stawiamy hetmana
  Kolumna 1 wiersz 0 jest atakowany przez hetmana z kolumny 0
  Kolumna 1 wiersz 1 jest atakowany przez hetmana z kolumny 0
  Kolumna 1 wiersz 2 nie jest atakowany, stawiamy hetmana
    Kolumna 2 wiersz 0 jest atakowany przez hetmana z kolumny 0
    Kolumna 2 wiersz 1 jest atakowany przez hetmana z kolumny 1
    Kolumna 2 wiersz 2 jest atakowany przez hetmana z kolumny 0
Kolumna 0 wiersz 1 nie jest atakowany, stawiamy hetmana
  Kolumna 1 wiersz 0 jest atakowany przez hetmana z kolumny 0
  Kolumna 1 wiersz 1 jest atakowany przez hetmana z kolumny 0
  Kolumna 1 wiersz 2 jest atakowany przez hetmana z kolumny 0
Kolumna 0 wiersz 2 nie jest atakowany, stawiamy hetmana
  Kolumna 1 wiersz 0 nie jest atakowany, stawiamy hetmana
    Kolumna 2 wiersz 0 jest atakowany przez hetmana z kolumny 0
    Kolumna 2 wiersz 1 jest atakowany przez hetmana z kolumny 1
    Kolumna 2 wiersz 2 jest atakowany przez hetmana z kolumny 0
  Kolumna 1 wiersz 1 jest atakowany przez hetmana z kolumny 0
  Kolumna 1 wiersz 2 jest atakowany przez hetmana z kolumny 0
a w wersji pełnej, tj. z "rysunkiem" drzewa, dla  $n = 4$  – tak:
```

```
+-\ Kolumna 0 wiersz 0 nie jest atakowany, stawiamy hetmana
| +- Kolumna 1 wiersz 0 jest atakowany przez hetmana z kolumny 0
| +- Kolumna 1 wiersz 1 jest atakowany przez hetmana z kolumny 0
| +-\ Kolumna 1 wiersz 2 nie jest atakowany, stawiamy hetmana
| | +- Kolumna 2 wiersz 0 jest atakowany przez hetmana z kolumny 0
| | +- Kolumna 2 wiersz 1 jest atakowany przez hetmana z kolumny 1
| | +- Kolumna 2 wiersz 2 jest atakowany przez hetmana z kolumny 0
| | +- Kolumna 2 wiersz 3 jest atakowany przez hetmana z kolumny 1
| +-\ Kolumna 1 wiersz 3 nie jest atakowany, stawiamy hetmana
| +- Kolumna 2 wiersz 0 jest atakowany przez hetmana z kolumny 0
| +-\ Kolumna 2 wiersz 1 nie jest atakowany, stawiamy hetmana
| | +- Kolumna 3 wiersz 0 jest atakowany przez hetmana z kolumny 0
```

| | +- Kolumna 3 wiersz 1 jest atakowany przez hetmana z kolumny 1
 | | +- Kolumna 3 wiersz 2 jest atakowany przez hetmana z kolumny 2
 | | +- Kolumna 3 wiersz 3 jest atakowany przez hetmana z kolumny 0
 | +- Kolumna 2 wiersz 2 jest atakowany przez hetmana z kolumny 0
 | +- Kolumna 2 wiersz 3 jest atakowany przez hetmana z kolumny 1
 +- \ Kolumna 0 wiersz 1 nie jest atakowany, stawiamy hetmana
 | +- Kolumna 1 wiersz 0 jest atakowany przez hetmana z kolumny 0
 | +- Kolumna 1 wiersz 1 jest atakowany przez hetmana z kolumny 0
 | +- Kolumna 1 wiersz 2 jest atakowany przez hetmana z kolumny 0
 | +- \ Kolumna 1 wiersz 3 nie jest atakowany, stawiamy hetmana
 | +- \ Kolumna 2 wiersz 0 nie jest atakowany, stawiamy hetmana
 | | +- Kolumna 3 wiersz 0 jest atakowany przez hetmana z kolumny 2
 | | +- Kolumna 3 wiersz 1 jest atakowany przez hetmana z kolumny 0
 | | +- Kolumna 3 wiersz 2 nie jest atakowany, stawiamy hetmana - SUKCES!
 | | +- Kolumna 3 wiersz 3 jest atakowany przez hetmana z kolumny 1
 | +- Kolumna 2 wiersz 1 jest atakowany przez hetmana z kolumny 0
 | +- Kolumna 2 wiersz 2 jest atakowany przez hetmana z kolumny 1
 | +- Kolumna 2 wiersz 3 jest atakowany przez hetmana z kolumny 0
 +- \ Kolumna 0 wiersz 2 nie jest atakowany, stawiamy hetmana
 | +- \ Kolumna 1 wiersz 0 nie jest atakowany, stawiamy hetmana
 | | +- Kolumna 2 wiersz 0 jest atakowany przez hetmana z kolumny 0
 | | +- Kolumna 2 wiersz 1 jest atakowany przez hetmana z kolumny 1
 | | +- Kolumna 2 wiersz 2 jest atakowany przez hetmana z kolumny 0
 | | +- \ Kolumna 2 wiersz 3 nie jest atakowany, stawiamy hetmana
 | | +- Kolumna 3 wiersz 0 jest atakowany przez hetmana z kolumny 1
 | | +- Kolumna 3 wiersz 1 nie jest atakowany, stawiamy hetmana - SUKCES!
 | | +- Kolumna 3 wiersz 2 jest atakowany przez hetmana z kolumny 0
 | | +- Kolumna 3 wiersz 3 jest atakowany przez hetmana z kolumny 2
 | +- Kolumna 1 wiersz 1 jest atakowany przez hetmana z kolumny 0
 | +- Kolumna 1 wiersz 2 jest atakowany przez hetmana z kolumny 0
 | +- Kolumna 1 wiersz 3 jest atakowany przez hetmana z kolumny 0
 +- \ Kolumna 0 wiersz 3 nie jest atakowany, stawiamy hetmana
 +- \ Kolumna 1 wiersz 0 nie jest atakowany, stawiamy hetmana
 | +- Kolumna 2 wiersz 0 jest atakowany przez hetmana z kolumny 1
 | +- Kolumna 2 wiersz 1 jest atakowany przez hetmana z kolumny 0
 | +- \ Kolumna 2 wiersz 2 nie jest atakowany, stawiamy hetmana
 | | +- Kolumna 3 wiersz 0 jest atakowany przez hetmana z kolumny 0
 | | +- Kolumna 3 wiersz 1 jest atakowany przez hetmana z kolumny 2
 | | +- Kolumna 3 wiersz 2 jest atakowany przez hetmana z kolumny 1
 | | +- Kolumna 3 wiersz 3 jest atakowany przez hetmana z kolumny 0
 | +- Kolumna 2 wiersz 3 jest atakowany przez hetmana z kolumny 0
 +- \ Kolumna 1 wiersz 1 nie jest atakowany, stawiamy hetmana
 | +- Kolumna 2 wiersz 0 jest atakowany przez hetmana z kolumny 1
 | +- Kolumna 2 wiersz 1 jest atakowany przez hetmana z kolumny 0

| +- Kolumna 2 wiersz 2 jest atakowany przez hetmana z kolumny 1

| +- Kolumna 2 wiersz 3 jest atakowany przez hetmana z kolumny 0

+-- Kolumna 1 wiersz 2 jest atakowany przez hetmana z kolumny 0

+-- Kolumna 1 wiersz 3 jest atakowany przez hetmana z kolumny 0

To są przykłady, ale należy je odwzorować dość wiernie, choć mogą różnić się kolejnością poddrzew lub podaną kolumną atakującego hetmana (może być ich więcej niż jedna "do wyboru", podajemy dowolną).

Samemu zastanów się, jak zrealizować w pamięci takie drzewo. Zauważ, że będzie ono n-arne, gdzie n będzie znane dopiero podczas uruchomienia, więc zamiast stałej liczby pól w struct-cie, do których odwoływał(a)byś się w kodzie przez wyrażenia podobne do `node->left`, `node->right`, trzeba będzie mieć w nim tablicę (długości n) wskaźników na dzieci.

Wskazane może być podzielenie kodu na moduły – tu również decyzja należy do Ciebie. Jeśli rozwiązanie będzie zawierać kilka plików, to po pierwsze powinny one we właściwy sposób dołączać odpowiednie nagłówki, oraz poddać się rozdzielnej kompilacji w następujący sposób (jeśli w katalogu mamy tylko pliki z Twojego rozwiązania):

```
gcc -Wall -Wextra -Werror -xc -std=c11 -c *.c  
gcc *.o -lm
```

Zadanie 2. Najwyższy numerolog Mezopotamii próbuje ustalić zestaw szczęśliwych liczb na przyszły rok. Oczywiście, nie mogą one zależeć od czynników tak kapryśnych jak układ gwiazd czy przelot ptaków; szczęśliwych liczb musi być nie za dużo, nie za mało, a w sam raz. Dlatego WOW (Wydział Obliczeń Wróżbiarskich) codziennie musi testować różne koncepcje nadwornego numerologa. Dzisiejszego ranka zadanie jest prostsze niż zwykle. Należy wczytać liczby x, y i policzyć, ile jest liczb w przedziale $[x, y]$ takich, że jednocześnie:

- suma pierwszych czterech cyfr jest podzielna przez 22,
- suma cyfr na pozycjach parzystych jest większa od sumy cyfr na pozycjach nieparzystych o przynajmniej 19,
- liczba zawiera trzy sąsiednie cyfry '3' i dwie sąsiednie cyfry '7'.

Pierwsze cyfry to te najbardziej znaczące, do tego najbardziej znacząca cyfra jest na pozycji nieparzystej. Liczba 17784333 spełnia ostatni warunek.

Pewnym kłopotem jest to, że możnowładcy babilońscy dawno znudzili się małymi liczbami, dlatego numerolog musi operować na dużych wartościach!

Wejście

W pierwszym i jedynym wierszu wejścia znajdują się naturalne liczby $0 < x, y < 10^{13}$ (tj. 10000000000000).

Wyjście

Na standardowe wyjście należy podać odpowiedź na zapytanie Arcynumerologa.

Przykłady

Przykład A

Wejście

1 1000000

Wyjście

0

Żadna z tak małych liczb nie spełnia wymogów.

Przykład B

Wejście

1 1777073339

Wyjście

1

Tylko liczba 1777073339 spełnia wymogi. Suma pierwszych czterech cyfr to $1+7+7+7=22$. Suma cyfr na pozycjach parzystych to $7+7+7+3+9=33$, nieparzystych $1+7+3+3=14$. Liczba zawiera trzy sąsiednie cyfry '7', ale tym lepiej.

Przykład C

Wejście

1800000000 2000000000

Wyjście

6

Szukane liczby to 1939333778, 1939333779, 1939333877, 1939333977, 1939773338 oraz 1939773339.

Uwagi

Jak można wywnioskować na podstawie limitów na wielkość liczb na wejściu, pętla sprawdzająca kolejno wszystkie liczby w przedziale nie może zadziałać zbyt dobrze. Polecamy rozważyć generowanie liczb rekurencyjnie, cyfra po cyfrze, jak w programach rekurencji z nawrotami. Można wtedy przerywać procedurę w odpowiednich momentach, gdy nie ma już szans na uzyskanie na końcu poprawnej liczby. Przykładowo, gdy pierwsze cztery cyfry nie spełniają warunku, nie ma sensu generować pozostałych cyfr. Potrzebne mogą być jeszcze nieco bardziej złożone dodatkowe warunki, np. gdy wygenerowaliśmy już większość cyfr i jest pewne, że nie uda nam się

'zmieścić' na pozostałych miejscach trzech sąsiednich trójek, lub uzyskać odpowiednio wysokiej sumy cyfr na indeksach parzystych, itd.