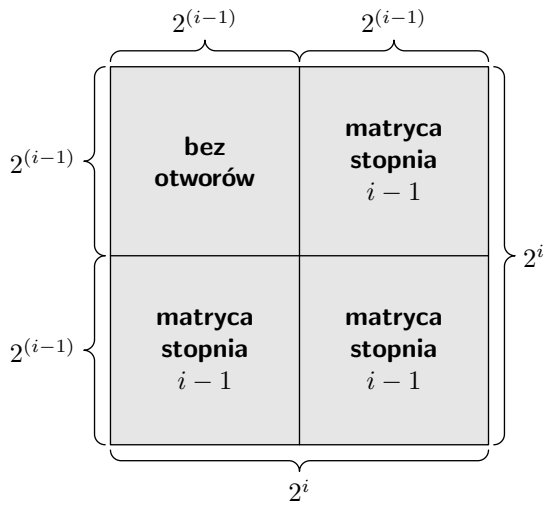
Wojciech Rytter
autor zadania, opracowanie

Marcin Engel
program wzorcowy

Pracownia malarska

Pracownia malarska przygotowuje seryjną produkcję obrazów. Obrazy będą wykonywane za pomocą kwadratowych matryc o różnych stopniach. Matryca stopnia i składa się z 2^i wierszy i 2^i kolumn. Na przecięciu pewnych wierszy i kolumn znajdują się otwory. Matryca stopnia 0 ma jeden otwór.

Dla $i > 0$ matryca stopnia i składa się z czterech kwadratów o rozmiarach $2^{(i-1)} \times 2^{i-1}$.

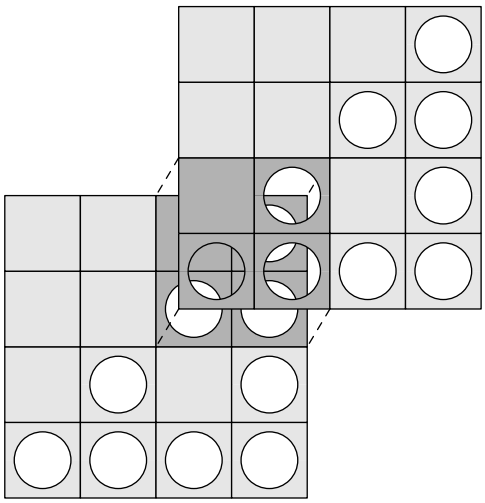


Oba prawe kwadraty oraz dolny lewy kwadrat są matrycami stopnia $i-1$. W górnym lewym kwadracie nie ma żadnych otworów. Obraz otrzymuje się w następujący sposób. Najpierw ustala się trzy nieujemne liczby całkowite n, x, y. Następnie umieszcza się dwie matryce stopnia n jedna na drugiej i górną matrycę przesuwa się o x kolumn w prawo i y wierszy w górę. Tak otrzymany wzorzec zostaje umieszczony na białym płótnie i na wspólną część obu matryc nanosi się żółtą farbę. W efekcie na płótnie pojawią się żółte plamy tylko w tych miejscach, w których otwory obu matryc pokrywają się.

42 *Pracownia malarska*

Przykład

Przyjrzyj się dwóm matorycom stopnia 2 przedstawionym na rysunku.



Górna matryca została przesunięta o 2 kolumny w prawo i o 2 wiersze w górę. W trzech miejscach otwory z obu matryc pokrywają się.

Zadanie

Napisz program, który:

- *wczytuje z pliku tekstowego MAL.IN stopień obu matryc oraz współrzędne przesunięcia górnej matrycy;*
- *oblicza liczbę żółtych plam na płótnie;*
- *zapisuje wynik w pliku tekstowym MAL.OUT.*

Wejście

Pierwszy wiersz pliku tekstowego MAL.IN zawiera liczbę całkowitą n, $0 \leqslant n \leqslant 100$. Jest to stopień matryc używanych w produkcji obrazów.

W drugim wierszu zapisana jest liczba całkowita x, zaś w trzecim liczba całkowita y, $0 \leqslant x, y \leqslant 2^n$. Liczba x jest liczbą kolumn, a y jest liczbą wierszy, o które należy przesunąć górną matrycę.

Wyjście

W pierwszym wierszu pliku wyjściowego MAL.OUT należy zapisać liczbę plam na płótnie.

Przykład

Dla pliku wejściowego MAL.IN:

```
2
2
2
```

poprawnym rozwiązaniem jest plik wyjściowy MAL.OUT:

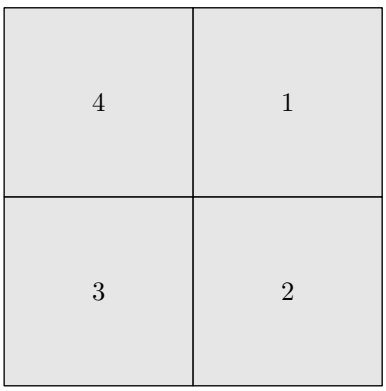
```
3
```

Twój program powienien szukać pliku MAL.IN w katalogu bieżącym i tworzyć plik MAL.OUT również w bieżącym katalogu. Plik zawierający napisany przez Ciebie program w postaci źródłowej powinien mieć nazwę MAL.???, grzie zamiast ??? należy wpisać co najwyżej trzyliterowy skrót nazwy użytego języka programowania. Ten sam program w postaci wykonalnej powinien być zapisany w pliku MAL.EXE.

ROZWIĄZANIE

Oznaczmy przez M_i matrycę o rozmiarach $2^i \times 2^i$ skonstruowaną zgodnie z opisem zadania. Jej wiersze ponumerujmy z dołu do góry, a kolumny od lewej do prawej, rozpoczynając od zera. Ponumerujmy także ćwiartki matrycy, tak jak na rysunku 1.

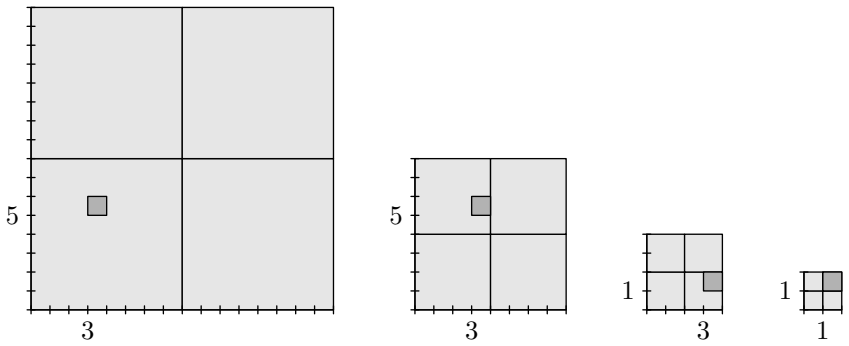
Rys. 1 Numeracja ćwiartek matrycy.



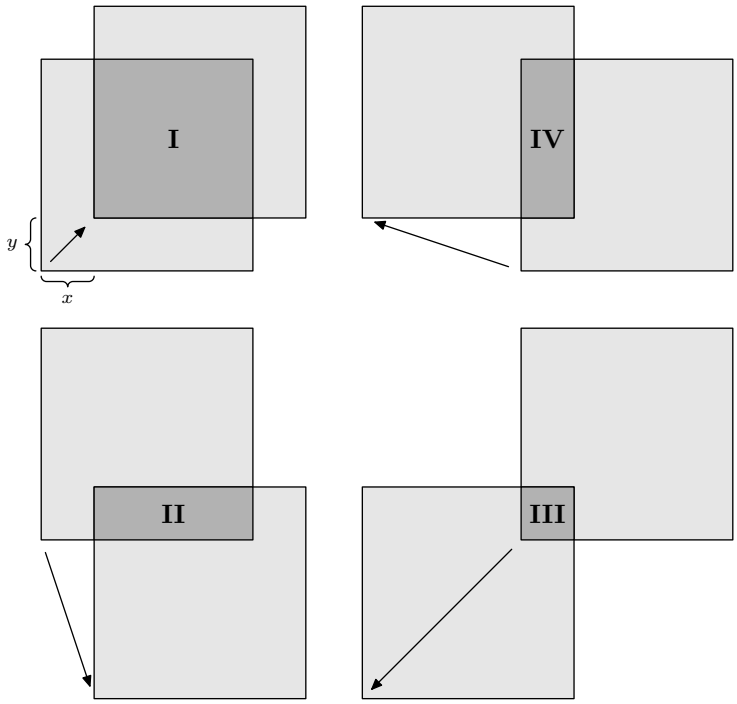
Początkowa matryca ma rozmiary $2^n \times 2^n$. Wyróżnijmy pole matrycy o współrzędnych (x, y). Należy ono do jednej z czterech ćwiartek. Zapiszmy numer tej ćwiartki i odrzućmy resztę matrycy. Otrzymaliśmy znów matrycę, na której leży wyróżnione pole (być może ma teraz inne współrzędne). Powyższy proces możemy powtarzać, dopóki nie otrzymamy matrycy o rozmiarach 1×1. Zapisywane numery ćwiartek tworzą ciąg, który nazywać będziemy **adresem ćwiartkowym** pola (x, y).

Skonstruujemy teraz algorytm, który będzie odpowiadał na pytanie nieco trudniejsze niż to postawione w treści zadania. Można w tym widzieć zjawisko podobne do występującego przy dowodzeniu przez indukcję, gdzie czasami łatwiej dowodzić silniejszej tezy indukcyjnej, gdyż tym samym możemy korzystać z silniejszego założenia indukcyjnego.

Rys. 2 Sekwencja kolejnych położeń pola $(3, 5)$ w coraz mniejszych ćwiartkach. Pole jest kolejno w ćwiartce trzeciej, czwartej, drugiej i pierwszej, stąd jego adresem ćwiartkowym jest $(3, 4, 2, 1)$.



Rys. 3 Wspólne części typu I, II, III i IV dwóch matryc o tym samym stopniu, przesuniętych względem siebie.



Spójrzmy na rysunek 3. Przedstawiono na nim cztery typy wspólnych części dwóch matryc stopnia i. Są to części wspólne matrycy stopnia i ze swoimi kopiami przesuniętymi o wektory $[x, y]$, $[x, y - 2^i]$, $[2^i - x, 2^i - y]$ i $[2^i - x, y]$. Będziemy je nazywać odpowiednio częściami wspólnymi typu I, II, III i IV. Niech $W_1(i)$, $W_2(i)$, $W_3(i)$ i $W_4(i)$ oznaczają odpowiednio liczby pokrywających się otworów w częściach

wspólnych typu I, II, III i IV matryc stopnia i.

Załóżmy teraz, że punkt (x, y) leży w pierwszej ćwiartce matrycy stopnia i i spróbujmy wyrazić liczby $W_1(i)$, $W_2(i)$, $W_3(i)$, $W_4(i)$ przy pomocy liczb $W_1(i-1)$, $W_2(i-1)$, $W_3(i-1)$, $W_4(i-1)$. Wygodnie w tym celu posłużyć się pomocniczymi macierzami o rozmiarach 4×4. Wprowadźmy macierz

$$A_1 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

taką, że

$$\begin{aligned} W_1(i) &= a_{11} \cdot W_1(i-1) + a_{12} \cdot W_2(i-1) + a_{13} \cdot W_3(i-1) + a_{14} \cdot W_4(i-1), \\ W_2(i) &= a_{21} \cdot W_1(i-1) + a_{22} \cdot W_2(i-1) + a_{23} \cdot W_3(i-1) + a_{24} \cdot W_4(i-1), \\ W_3(i) &= a_{31} \cdot W_1(i-1) + a_{32} \cdot W_2(i-1) + a_{33} \cdot W_3(i-1) + a_{34} \cdot W_4(i-1), \\ W_4(i) &= a_{41} \cdot W_1(i-1) + a_{42} \cdot W_2(i-1) + a_{43} \cdot W_3(i-1) + a_{44} \cdot W_4(i-1). \end{aligned}$$

Łatwo sprawdzić (umieszczając jedną matrycę nad drugą tak, by punkt przesunięcia był w pierwszej ćwiartce i badając położenia ćwiartek między sobą), że w tym przypadku

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

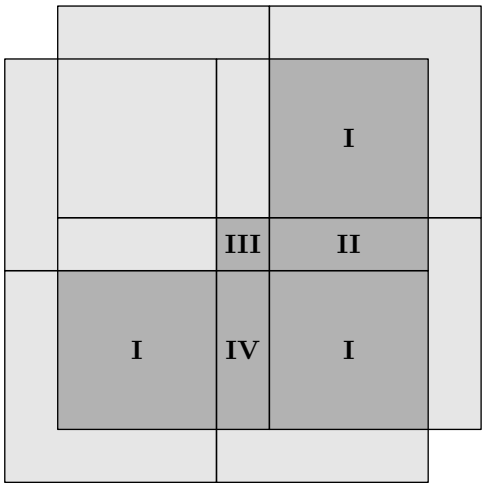
W k-tym wierszu macierzy trzeba zapisać, jak się wyraża $W_k(i)$ za pomocą $W_1(i-1)$, $W_2(i-1)$, $W_3(i-1)$ i $W_4(i-1)$. Pokażemy przykładowo, w jaki sposób obliczamy pierwszy wiersz macierzy A_3. W tym przypadku punkt (x, y) leży w trzeciej ćwiartce. Część wspólna składa się z trzech części typu I, oraz po jednej części typu II, III i IV (w sensie wspólnych części matryc stopnia $i-1$). Stąd pierwszym wierszem w macierzy A_3 jest $[3, 1, 1, 1]$ (zob. rysunek 4).

W powyższy sposób wyznaczamy macierze A_1, A_2, A_3 i A_4 dla przypadków, gdy dolny lewy róg przesuniętej matrycy znajduje się odpowiednio w pierwszej, drugiej, trzeciej i czwartej ćwiartce. Są one podstawową strukturą danych potrzebną do rozwiązania zadania.

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 3 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 3 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_4 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 3 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Rys. 4 Pierwszym wierszem macierzy A_3 jest $[3, 1, 1, 1]$, co oznacza, że jeżeli dolny lewy róg nałożonej matrycy znajduje się w pierwszej ćwiartce, to mamy trzy wspólne części typu I oraz po jednej wspólnej części typu II, III i IV dla matryc stopnia o jeden mniejszego. (Pamiętaj, że nie zaciemnione fragmenty części wspólnej nałożonych matryc nie zawierają otworów.)



Dla uproszczenia zapisu wprowadźmy jeszcze operację mnożenia wektora* przez macierz:

$$[v_1, v_2, v_3, v_4] \otimes \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = [r_1, r_2, r_3, r_4],$$

gdzie

$$\begin{aligned} r_1 &= [v_1 \cdot a_{11} + v_2 \cdot a_{21} + v_3 \cdot a_{31} + v_4 \cdot a_{41}], \\ r_2 &= [v_1 \cdot a_{12} + v_2 \cdot a_{22} + v_3 \cdot a_{32} + v_4 \cdot a_{42}], \\ r_3 &= [v_1 \cdot a_{13} + v_2 \cdot a_{23} + v_3 \cdot a_{33} + v_4 \cdot a_{43}], \\ r_4 &= [v_1 \cdot a_{14} + v_2 \cdot a_{24} + v_3 \cdot a_{34} + v_4 \cdot a_{44}]. \end{aligned}$$

Przy zastosowaniu powyższej notacji algorytm można zapisać następująco (zakładamy, że $[x, y]$ to wektor przesunięcia, a n to rozmiar matrycy):

```
1: adres := adresĆwiartkowy(x, y);
2: v := [1,0,0,0];
3: for t := 1 to n do begin
4:    i := adres[t];
5:    v := v ⊗ A_i
6: end;
```

* Pisząc „wektor” mamy tu na myśli skończony, niekoniecznie dwulementowy, ciąg liczb. Tutaj nie interesuje nas interpretacja geometryczna wektora. Słowo „wektor” w takim znaczeniu jest powszechnie używane w algebrze liniowej.

7: *wypisz*$(v[1])$

Poprawność algorytmu można uzasadnić zauważając, że zawsze po wykonaniu linii 5 prawdziwy jest warunek (tzw. niezmiennik):

$$wynik = v[1] \cdot W_1(n-t) + v[2] \cdot W_2(n-t) + v[3] \cdot W_3(n-t) + v[4] \cdot W_4(n-t),$$

gdzie *wynik* oznacza wartość wypisywaną przez algorytm w linii 7. (Przyjmujemy $W_1(0) = 1$ i $W_2(0) = W_3(0) = W_4(0) = 0$.)

Mnożenie wektora przez macierz możemy zastąpić gotowym ciągiem operacji, na przykład mnożenie wektora v przez macierz A_4 sprowadza się do:

1: $w[1] := v[1] + v[2]$;
2: $w[2] := 3 \cdot v[2]$;
3: $w[3] := v[2] + v[3]$;
4: $w[4] := v[1] + v[3]$;
5: $v := w$

Możemy wcześniej wyznaczyć ręcznie 4 ciągi operacji i zastąpić mnożenie przez macierz tymi operacjami. Będzie to na pewno bardziej efektywne obliczeniowo (w sensie czasu), chociaż notacja z macierzami wydaje się bardziej przekonywająca.

Na przykład dla matrycy o rozmiarze 16×16 i przesunięcia [3,5] poprawną odpowiedzią będzie 14, co obliczamy następująco:

inicjacja:	$adres := [3, 4, 2, 1]$
iteracja 1:	$[1, 0, 0, 0] * A_3 := [3, 1, 1, 1]$
iteracja 2:	$[3, 1, 1, 1] * A_4 := [4, 3, 2, 4]$
iteracja 3:	$[4, 3, 2, 4] * A_2 := [8, 6, 6, 12]$
iteracja 4:	$[8, 6, 6, 2] * A_1 := [14, 12, 18, 18]$
zakończenie:	*wypisz*(14)

TESTY

Oprócz testów poprawnościowych dla danych wejściowych o niedużych rozmiarach, przeprowadzono testy wydajnościowe dla danych o maksymalnym rozmiarze. Pozwoliło to na odsianie rozwiązań nadmiernie zużywających pamięć. Do sprawdzenia rozwiązań zawodników użyto 11-tu testów MAL0.IN–MAL10.IN.

- MAL0.IN — test z treści zadania.
- MAL1.IN–MAL3.IN, MAL10.IN — testy poprawnościowe dla $n \leqslant 5$ i x, y w przedziale [0..31].
- MAL4.IN, MAL5.IN — testy dla przypadków brzegowych: przesunięcie zerowe, maksymalne przesunięcie po obu współrzędnych.
- MAL6.IN–MAL9.IN — testy wydajnościowe.