Przykład

Poniżej podajemy przykłady obrazków poprawnych, naprawialnych i nienaprawialnych rozmiaru 5×5. Niepoprawne bity parzystości w obrazkach zostały wyróżnione podkreśleniem.

011000	011011	011011	011011	011011
111010	11101**1**	11101**1**	11101**1**	11101**1**
111111	111111	111111	111111	111111
011000	011000	011000	01100**1**	01100**1**
110011	110110	110110	110110	110110
11011	11**1**00	11000	11000	**0**1000
poprawny	naprawialny	naprawialny	nienaprawialny	nienaprawialny

Podaj liczbę obrazków poprawnych, liczbę obrazków naprawialnych oraz liczbę obrazków nienaprawialnych. Ponadto podaj największą liczbę błędnych bitów parzystości występujących w jednym obrazku.

64.4.

W obrazku naprawialnym wystarczy zmienić jedną wartość, aby uzyskać obrazek poprawny. Dokładniej, jeśli niepoprawne są bity parzystości *i*-tego wiersza i *j*-tej kolumny, wystarczy zmienić *j*-ty piksel w *i*-tym wierszu. Jeśli niepoprawny jest dokładnie jeden bit parzystości (wiersza **albo** kolumny), wystarczy zmienić ten bit parzystości.

Przykład

Rozważmy następujące dwa obrazki naprawialne rozmiaru 5×5 (niepoprawne bity parzystości w obrazkach zostały podkreślone).

Obraz 1:	Obraz 2:
011011	011011
11**1**01**1**	11101**1**
111111	111111
011000	011000
110110	110110
11**1**00	11000

Zmieniając trzecią jedynkę w drugim wierszu pierwszego obrazka, uzyskamy obrazek poprawny (niepoprawne były bity parzystości trzeciej kolumny i drugiego wiersza). Zmieniając niepoprawny bit parzystości w drugim wierszu drugiego obrazka z 1 na 0, również uzyskamy obrazek poprawny.

Podaj numery obrazków naprawialnych, przyjmując, że numery kolejnych obrazków w pliku to 1, 2, 3 itd. Przy numerze każdego obrazka naprawialnego podaj numer wiersza i kolumny wartości, którą wystarczy zmienić, aby uzyskać obrazek poprawny.

Zadanie 65.

Wiązka zadań *Ułamki*

W pliku `dane_ulamki.txt` znajduje się 1000 par liczb naturalnych dodatnich, mniejszych niż 12 000. Każda para liczb jest zapisana w osobnym wierszu, liczby w wierszu rozdzielone są pojedynczym znakiem odstępu. Parę liczb zapisanych w tym samym wierszu interpretujemy jako ułamek, którego licznikiem jest pierwsza liczba, a mianownikiem — druga liczba.

64.1.

Obrazek nazywamy *rewersem*, jeśli liczba występujących w nim pikseli czarnych jest większa od liczby pikseli białych.

Przykład: W obrazku z powyższego przykładu występuje 18 pikseli czarnych i 7 pikseli białych. Zatem jest on rewersem.

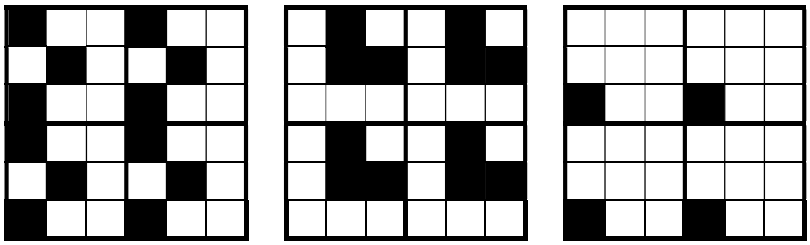
Podaj, ile jest w pliku obrazków, które są rewersami. Podaj też największą liczbę pikseli czarnych występujących w jednym obrazku.

64.2.

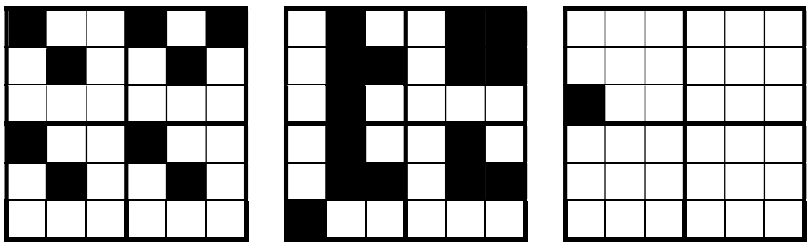
Obrazek rozmiaru $n \times n$ będziemy nazywać *rekurencyjnym,* jeśli n jest parzyste oraz obrazek składa się z 4 kopii tego samego obrazka rozmiaru $\frac{n}{2} \times \frac{n}{2}$.

Przykład

Poniżej podajemy 3 obrazki rozmiaru 6×6, które są rekurencyjne.



Natomiast poniższe obrazki nie są rekurencyjne:



Podaj liczbę obrazków rekurencyjnych w pliku wejściowym. Ponadto podaj opis pierwszego obrazka rekurencyjnego występującego w pliku. W opisie obrazka pomiń bity parzystości (pamiętaj, że obrazek składa się z 20 wierszy po 20 pikseli, które reprezentujemy jako ciąg zer i jedynek).

64.3.

Obrazek nazywamy *poprawnym*, jeśli wszystkie bity parzystości są w nim poprawne (zarówno w wierszach, jak i kolumnach). Obrazek nazywamy *naprawialnym*, jeśli nie jest poprawny, a jednocześnie co najwyżej jeden bit parzystości wiersza i co najwyżej jeden bit parzystości kolumny jest w nim niepoprawny.

Natomiast *nienaprawialnym* nazywamy obrazek, który nie jest poprawny i nie jest naprawialny.

63.3.

Liczbą półpierwszą nazywamy taką liczbę, która jest iloczynem dwóch liczb pierwszych. Podaj, ile ciągów z pliku `ciagi.txt` jest reprezentacją binarną liczb półpierwszych. Dodatkowo podaj największą i najmniejszą liczbę półpierwszą w zapisie dziesiętnym.

Przykład

Dla zestawu ciągów:

```
100010
1101001001
1100101
1111111111
10010110000010010010
```

podkreślone ciągi są zapisem binarnym liczb półpierwszych, ponieważ:

$(100010)_2 = 34 = 2 * 17$, więc jest liczbą półpierwszą;

$(1101001001)_2 = 841 = 29 * 29$, więc jest liczbą półpierwszą;

$(1100101)_2 = 101 = 101 * 1$;

$(1111111111)_2 = 1023 = 3 * 11 * 31$;

$(10010110000010010010)_2 = 614546 = 2 * 307273$, więc jest liczbą półpierwszą.

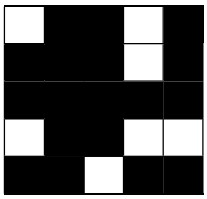
Zadanie 64.

Wiązka zadań *Obrazki*

Bit parzystości ciągu złożonego z zer i jedynek jest równy 0, gdy w ciągu tym występuje parzysta liczba jedynek, w przeciwnym razie bit parzystości jest równy 1.

Czarno-biały obrazek rozmiaru $n \times n$ składa się z n wierszy po n pikseli. Każdy wiersz pikseli reprezentujemy jako ciąg zer i jedynek, każdy biały piksel reprezentujemy przez 0, czarny — przez 1. Na końcu każdego wiersza dodany jest bit parzystości, podobnie pod ostatnim wierszem obrazka dołączony jest wiersz bitów parzystości każdej z n kolumn. **Bitów parzystości nie traktujemy jako części obrazka.**

Przykład: Poniżej podajemy obrazek rozmiaru 5×5 oraz jego reprezentację, wraz z odpowiednimi bitami parzystości (bity parzystości zostały podkreślone):



Obrazek

```
0 1 1 0 1 1
1 1 1 0 1 0
1 1 1 1 1 1
0 1 1 0 0 0
1 1 0 1 1 0
1 1 0 0 0
```

Reprezentacja

Plik `dane_obrazki.txt` składa się z opisu 200 czarno-białych obrazków o rozmiarze 20×20 pikseli. **Sąsiednie obrazki oddzielone są w pliku pustym wierszem.**

Napisz program(-y), który poda odpowiedzi na pytania postawione w poniższych zadaniach. Odpowiedzi zapisz w pliku `wyniki_obrazki.txt`. Odpowiedź do każdego zadania rozpocznij w nowym wierszu, poprzedzając ją numerem zadania.