|  |  |
| --- | --- |
|  | **ЮГОЗАПАДЕН УНИВЕРСИТЕТ „СВ. НЕОФИТ РИЛСКИ“ БЛАГОЕВГРАД**  **XXVII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ, 8-10 май, 2015 г.** |

**Задача J. ИЗОБРАЖЕНИЕ**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

Черно-бели изображения се представят като растер с размери *N* x *N* (виж фигурата). Напишете програма, която намира максималното лице (измерено в брой пиксели) на правоъгълник от изображението, съставен само от черни пиксели.

**Вход:** Програмата трябва да може да обработва няколко примера при едно изпълнение. На първия ред на **стандартния вход** ще бъде зададен броят *T* на тестовите примери. Всеки тестов пример започва с ред, на който ще е зададен размерът *N* на растера. Следват *N* реда с по един битов низ с дължина *N* на всеки от тях, представящи самото изображение. Белите пиксели са кодирани с нули, а черните – с единици.

**Изход:** За всеки тестов пример програмата трябва да изведе на отдeлен ред на **стандартния изход** намереното лице на максимален черен правоъгълник.

**Ограничения:** 5 **≤** *N* **≤** 2048.

**Пример:**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1  10  0000000000  0111000000  0011000000  0001010100  0001111110  0000111100  0000111100  0000111100  0000100100  0000100100 | 16 |

|  |  |
| --- | --- |
|  | **SOUTH-WEST UNIVERSITY „ST. NEOFIT RILSKI“ BLAGOEVGRAD**  **XXVII BULGARIAN COLLEGIATE PROGRAMMING CONTEST, May 8-10, 2015** |

**Task J. IMAGE**

Black and white images are presented in a raster sized *N* x *N* pixels (see the figure). Write a program to find the maximum surface (measured with the number of pixels) of a rectangle from the image composed of only black pixels.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

**Input**: The program must be able to handle a few test cases, in one run. The first line of the **standard input** will contain one integer *T* – the number of test cases. Each test case will start with a line which contains the size *N* of the raster. Then *N* lines follow with a bit string of length *N* on each of them representing the image. White pixels are encoded with zeros and black – with ones.

**Output:** For each test case the program should print on separate line of the **standard output** the surface of a maximum black rectangle.

**Restrictions:** 5 **≤** *N* **≤** 2048.

**Example:**

|  |  |
| --- | --- |
| Input | Output |
| 1  10  0000000000  0111000000  0011000000  0001010100  0001111110  0000111100  0000111100  0000111100  0000100100  0000100100 | 16 |