

Cumpleaños

El problema

Estamos organizando el cumpleaños de nuestro hermano pequeño, que se va a traer a casa a unos cuantos amigos para ver una película en el salón. Aunque en clase son n niños (uno de ellos mi hermano), no tenemos hueco para tantos; tenemos sólo sitio para m . De visitas anteriores de los amigos, sabemos cuál es la preferencia de cada niño por cada uno de los sitios, de forma que tenemos una matriz de $n \times m$ enteros que nos marca su satisfacción si se le asigna ese puesto (que podría ser negativa).

Queremos decidir a qué niños invitar y qué puestos asignar a cada uno para maximizar la satisfacción total, teniendo en cuenta que:

- A nadie se le asignará un puesto que tenga satisfacción negativa. Prefiere no ir, antes de utilizar ese sitio.
- Los niños se han vuelto muy supersticiosos y en general si su número de lista de clase es k , tienden a evitar sentarse en el puesto i en el que $k \% m = i$ (con m el número de puestos totales). Es por esto que no se considerarán buenas asignaciones en las que esto ocurra más de $m=3$ veces (usando división entera).
- Obviamente, tu hermano debe ser invitado...
- Tanto los números en la lista de clase como los puestos están numerados empezando a contar desde el 0.

Se pide diseñar e implementar un algoritmo “vuelta atrás” que dados (i) el número n de niños de la clase; (ii) el número m de puestos en el salón; (iii) el identificador de nuestro hermano en la lista de clase; y (iv) la matriz de satisfacción, escriba la satisfacción que puede conseguirse con la mejor asignación posible. Si no hay tal asignación, se escribirá un 0.

Trabajo a realizar

Debe diseñarse el algoritmo “vuelta atrás” pedido, completando los apartados indicados entre comentarios en el archivo `plantilla.cpp` que se proporciona como apoyo. Debe implementarse, además, el algoritmo. El punto de entrada al mismo será la función `satisfaccion_maxima`. Si se considera necesario, deberá definirse e implementarse una generalización adecuada, y definir el algoritmo pedido como una inmersión de dicha generalización.

El archivo completo debe entregarse a través del juez en línea de la asignatura.

Programa de prueba

Se proporciona un programa de prueba que lee por la entrada estándar casos de prueba, los resuelve invocando a `satisfaccion_maxima`, e imprime los resultados.

Cada caso de prueba consistirá en una primera línea con tres números. Los dos primeros representan el número de niños de la clase n (como mucho n valdrá 20) y el número de huecos en el salón m ($0 < m \leq n$); el tercer número, h indica el número que ocupa en la clase mi hermano ($0 < h \leq n$). A continuación, aparecerán m líneas para indicar la satisfacción de cada persona con cada puesto. En concreto, la fila i -ésima ($0 \leq i < m$) contiene la satisfacción de cada niño con el puesto i -ésimo. La entrada termina con una línea que contiene únicamente -1.

Para cada caso de prueba se escribirá la satisfacción máxima que puede conseguirse.

A continuación, se muestra un ejemplo de entrada / salida:

Entrada	Salida
4 2 0 5 10 8 -10 10 3 8 -5 4 2 1 10 5 5 100 -1 -1 -1 -1 4 2 0 10 7 8 -10 -1 10 3 -5 4 2 0 10 100 1000 200 10 100 1000 100 -1	20 5 0 210