

Universidade Federal de Goiás – UFG  
Instituto de Informática – INF  
Bacharelados (Núcleo Básico Comum)

Algoritmos e Estruturas de Dados 1 – 2022/2

Torneio de Resolução de Problemas 05 - Métodos de Ordenação

Turma: INF0286 – Prof. Cedric Luiz de Carvalho

---

## 1 Máquina de Café

O Sr. Vanila é dono de uma sofisticada cafeteria na centro da cidade – a StarCoffee – no dia da Proclamação da República, decidiu que irá distribuir cafés grátis como campanha de *marketing* durante todas as 24h deste dia.

Ele decidiu instalar  $p$  máquinas de café na StarCoffee. Cada uma das máquinas pode servir uma pessoa por minuto. Ele recebeu a lista de  $n$  pessoas que irão visitar sua cafeteria, todos com hora marcada.

O momento da chegada de cada pessoa é denotado por dois números naturais  $(h, m)$ , onde  $h$  é a hora da visita e  $m$  é o minuto da visita naquela hora<sup>1</sup>.

Devido à tecnologia, no século XXI somos todos(as) *impacientes*, e, assim, se uma pessoa tiver que esperar para obter o seu café após sua chegada na StarCoffee, a pessoa fica **brava**. O Sr. Vanila está preocupado com o custo desta campanha de *marketing* e quer configurar o número mínimo de máquinas de café para que nenhuma pessoa fique brava durante sua estada na StarCoffee. Suponha, para simplificação da resolução, de que não há a menor possibilidade de que qualquer uma das máquinas instaladas apresente defeito durante todo o dia da promoção.

Você, como exímio programador de computadores, recebeu a tarefa de ajudá-lo a encontrar o número mínimo de máquinas de café necessárias para que todos(as) os(as) participantes da campanha de *marketing* fiquem felizes, ou seja, **não bravos(as)**.

### Entrada

A primeira linha da entrada contém  $t \in \mathbb{N}^*$ , o número de casos de testes, com  $1 \leq t \leq 100$ .

Cada grupo de  $(n + 1)$  linhas seguintes corresponde a um caso de teste, onde  $n$  representa o número de pessoas a serem servidas naquele dia, com  $n \in \mathbb{N}^*$  e  $n \leq 10.000$ .

A primeira linha de cada caso de teste contém o valor de  $n$ . As  $n$  linhas seguintes contém dois números naturais,  $h$  e  $m$ , separados por um espaço em branco, representando cada cliente específico.

### Saída

Para cada caso de teste fornecido, o programa deve imprimir, numa única linha, o número mínimo de máquinas de café necessário para atender ao problema anteriormente definido.

---

<sup>1</sup>O horário definido por  $h:m$  representa o exato momento em que a pessoa entra na cafeteria e se coloca à frente de uma das máquinas de café disponíveis, onde  $0 \leq h \leq 23$  e  $0 \leq m \leq 59$ .

## Restrições

Lembre-se: Todas as restrições definidas devem ser satisfeitas.

## Exemplos

Entrada	Saída
1 7 10 20 5 40 10 20 23 11 5 50 10 30 17 12	2

Entrada	Saída
2 8 8 0 9 10 9 11 8 0 8 0 16 40 8 0 16 41 10 13 0 13 1 13 2 13 3 13 0 13 1 13 0 13 0 13 0 13 4	4 5

Entrada		Saída	
1		12	
30			
9 20			
9 20			
9 20			
9 25			
9 22			
9 20			
9 20			
9 20			
9 21			
9 23			
9 20			
9 22			
9 20			
9 21			
9 24			
9 20			
9 21			
9 23			
9 22			
9 20			
9 22			
9 25			
9 22			
9 20			
9 20			
9 23			
9 25			
9 22			
9 24			
9 25			

---

## 2 Olimpíadas

O Comitê Olímpico Internacional (COI) está visitando as cidades candidatas a sediar as Olimpíadas de 2032 e, desta vez, Goiânia é uma das cidades concorrentes, mas a competição entre as cidades candidatas está muito acirrada, pois Buenos Aires está sendo considerada uma forte candidata.

O COI tem um conjunto de exigências que devem ser obedecidas pelas cidades candidatas, como: (1) boas *arenas* para os jogos (ginásios, campos de futebol, pistas de atletismo, parque aquático,...); (2) bons alojamentos; (3) um plano para o tráfego de veículos durante os jogos, etc.

Durante sua visita à Goiânia, o COI colocou ainda mais uma exigência: a demonstração da qualidade dos sistemas de informática. Especificamente, o COI quer que a organização local demonstre a sua capacidade em informática produzindo um programa de computador que gere a classificação final dos países, considerando o número de medalhas recebidas pelos atletas de cada país participante da futura olimpíada.

Para ser justo, o COI abriu um concurso que permite que equipes de estudantes de quaisquer faculdades, centros universitários e universidades (públicas e privadas) apresentem seus *programas de computador* para cumprir a tarefa. O programa que melhor resultado obtiver numa *bancada de testes* a ser proposta pelo COI, em momento oportuno, será o escolhido.

### Tarefa

Você está na equipe que o INF/UFG designou para *vencer* este concurso.

Sua tarefa é escrever um programa  $\mathbb{C}$  que, dada a informação dos países que receberam medalhas de ouro, prata e bronze em cada modalidade esportiva, gere a lista de classificação dos países na competição.

Nesta tarefa, os países serão identificados por números inteiros:  $1, 2, 3, \dots, n$ , com  $2 \leq n \leq 200$ .

O melhor colocado deve ser o país que conseguiu o maior número de medalhas de ouro, havendo empate entre dois ou mais países, o melhor colocado é o país que conseguiu o maior número de medalhas de prata. Novamente havendo empate, o melhor colocado é o país que recebeu o maior número de medalhas de bronze. Por fim, se ainda assim houver empate, o melhor classificado é o que tem o maior número de identificação.

### Entrada

A primeira linha da entrada contém dois números naturais  $n$  e  $m$ , separados por um único espaço em branco, e indicando, respectivamente, o número de países e número de modalidades esportivas envolvidas na competição ( $1 \leq m \leq 50$ ). Os países são identificados por números inteiros de 1 a  $n$ .

Cada uma das  $m$  linhas seguintes contém três números inteiros  $O$ ,  $P$  e  $B$ , separados por um único espaço em branco entre eles, representando os identificadores dos países cujos atletas receberam, respectivamente, medalhas de ouro, prata e bronze.

Assim, se uma das  $m$  linhas contém os números 3 2 1, significa que nessa modalidade a medalha de ouro foi ganha pelo país cuja identificação é 3, a de prata pelo país cuja identificação é 2 e a de bronze pelo país cuja identificação é 1.

Uma linha com 10 10 10, significa que o país cuja identificação é 10 recebeu as três medalhas daquela modalidade (ouro, prata e bronze).

### Saída

Seu programa deve imprimir, na saída padrão, uma única linha contendo  $n$  números, separados por um único espaço em branco entre eles, representando os países na ordem decrescente de classificação, da esquerda para a direita.

Exemplos

Entrada	Saída
2 2 2 1 2 1 2 2	2 1

Entrada	Saída
4 3 3 2 1 4 3 1 4 3 1	4 3 2 1

Entrada	Saída
3 3 3 1 2 2 3 1 1 2 3	3 2 1

---

### 3 Corrida Escolar

Na escola onde “*Nelsinho*” há, uma vez por ano, uma tradicional corrida de estudantes ao redor de seu prédio principal.

Apesar, evidentemente, de haver estudantes dos mais variados períodos, todos(as) os(as) alunos(as) da escola são convidados(as) a participar, sendo impossível que todos participem da mesma corrida, mas em diferentes categorias.

Para contornar esse problema, os(as) professores(as) cronometram o tempo que cada aluno(a) consome para dar cada uma das voltas ao redor do prédio principal, e depois comparam os tempos para descobrir a classificação final, de acordo com a categoria a qual pertence cada estudante.

#### Tarefa

Sua tarefa é, sabendo o número de competidores(as), o número de voltas de que consistiu a corrida e os tempos de cada aluno(a) competidor(a), descobrir quem foi o(a) aluno(a) vencedor(a), para que ele(a) possa receber uma medalha comemorativa.

Há, obviamente, duas categorias distintas: masculina (M) e feminina (F).

#### Entrada

A primeira linha da entrada contém dois naturais,  $n$  e  $m$ , representando o número de competidores(as) e o número de voltas da corrida, respectivamente, sendo que  $(1 \leq n \leq 200)$  e  $(1 \leq m \leq 5)$ .

Na sequência, cada linha representa um competidor(a) e, portanto, haverá  $n$  linhas representando os competidores(as).

Em cada linha haverá  $(m + 1)$  números naturais: o primeiro será 1 para indicar que se trata de um *menino* e 2 para indicar se trata de uma *menina*, e os  $m$  seguintes indicam o tempo consumido em cada uma das voltas dadas por aquele(a) competidor(a). O tempo  $t$  é expresso em segundos e, por isso, garante-se que não houve dois(duas) competidores(as) que gastaram o mesmo tempo para completar a corrida inteira, ou seja, o tempo total de todas as suas voltas.

Para simplificar a solução do problema considera-se que nenhum(a) dos(as) competidores(as) desistiu da corrida, ou seja, todos(as) concluíram as  $m$  voltas da corrida da qual participou. Sabe-se também que  $1 \leq t \leq 1000$ .

#### Saída

A saída deve consistir de dois números naturais, separados por um único espaço em branco entre eles, que correspondem, respectivamente, ao vencedor da prova masculina à vencedora da prova feminina.

#### Exemplos

Entrada	Saída
2 3 1 2 1 2 2 1 2 3	1 2

Entrada	Saída
4 3 1 3 2 1 1 4 3 1 2 1 4 2 2 6 3 1 7	1 3

Entrada	Saída
3 3 2 3 5 6 2 1 2 3 1 1 1 1	3 2

---

## 4 Fermat e os Números Primos

Pierre de Fermat (1607 – 1665) foi advogado e magistrado francês que tinha, como *hobby*, estudar Matemática. Sua biografia mostra que ele era um homem cortês e amigável, mas que não era propenso a relacionamentos próximos: preferia comunicar-se por meio de cartas.

Numa carta datada de 25 de dezembro de 1640, ele escreveu para seu amigo, o padre Martin Mersenne (1588 – 1648), que havia feito uma descoberta “*maravilhosa*”: de que alguns números primos podiam ser escritos como a soma do quadrado de dois números naturais.

Por exemplo:

- $2 = 1 + 1 = 1^2 + 1^2$ ;
- $5 = 1 + 4 = 1^2 + 2^2$ ;
- $13 = 4 + 9 = 2^2 + 3^2$ ;
- $17 = 1 + 16 = 1^2 + 4^2$ .

Por outro lado, identificou que outros números primos, como 3, 7, 11, 19 e 23 não podiam ser decompostos desta maneira.

Assim, os números primos que podem ser decompostos como a soma do quadrado de dois números são chamados, atualmente, de “*Números Primos de Fermat*” ou, simplesmente, “*Primos de Fermat*”.

### Tarefa

Sua tarefa é escrever um programa  $\mathbb{C}$  que seja capaz de imprimir os “*Primos de Fermat*” que são menores ou iguais a um certo número  $n$ ,  $n \in \mathbb{N}^*$ , fornecido.

### Entrada

A primeira linha da entrada contém o número de casos de teste  $t$ , sendo que  $1 \leq t \leq 50$ .

Na linha seguinte haverá os  $t$  números, sempre separados por um único espaço em branco entre eles. Sabe-se também que  $1 \leq t_i \leq 10^6$ , com  $1 \leq i \leq t$ .

### Saída

Cada um dos  $t$  casos de teste deverá ter sua saída impressa numa única linha, como os “*Primos de Fermat*” separados entre si por um único espaço em branco.

Nos casos de teste de ordem ímpar ( $1^\circ$ ,  $3^\circ$ ,  $5^\circ$ ,  $7^\circ$ , ...), os números devem ser impressos em ordem *crescente*, enquanto que nos casos de teste de ordem par ( $2^\circ$ ,  $4^\circ$ ,  $6^\circ$ ,  $8^\circ$ , ...) eles devem ser impressos em ordem *decrescente* – veja os exemplos a seguir.

### Exemplos

Entrada	Saída
1 50	2 5 13 17 29 37 41



Entrada	Saída
3 50 80 100	2 5 13 17 29 37 41 73 61 53 41 37 29 17 13 5 2 2 5 13 17 29 37 41 53 61 73 89 97

Entrada	Saída
7 28 23 30 12 25 13 11	2 5 13 17 17 13 5 2 2 5 13 17 29 5 2 2 5 13 17 13 5 2 2 5

**Observação:**

No primeiro exemplo, como há apenas um caso de teste, os números, na saída, devem ser impressos em ordem crescente.

No segundo exemplo há três casos de teste: o primeiro (50) e o terceiro (100) devem ter seus números impressos, na saída, em ordem crescente, enquanto que o segundo (80) deve ter seus números impressos em ordem decrescente.