Tinta, suor e FPS



Depois de um semestre cansativo, você finalmente voltou para sua cidade para passar as férias com a família. Após meses sobrevivendo à base de café, trabalhos entregues às 23h59 e promessas de que “no próximo semestre eu me organizo”, seu plano era simples: descansar, assistir a alguns k-dramas, testar a nova atualização do Minecraft e tentar subir de novo para o Ferro II no Valorant sem perder as poucas amizades que ainda tem.

Porém, só dois dias depois que você voltou, comeca a perceber seus pais mais desapontados com você do que o normal. A confirmação das suas suspeitas veio naquela madrugada, quando você saiu do quarto para sua terceira janta e ouviu seus pais conversando na cozinha:

— A criatura precisa arrumar um emprego.

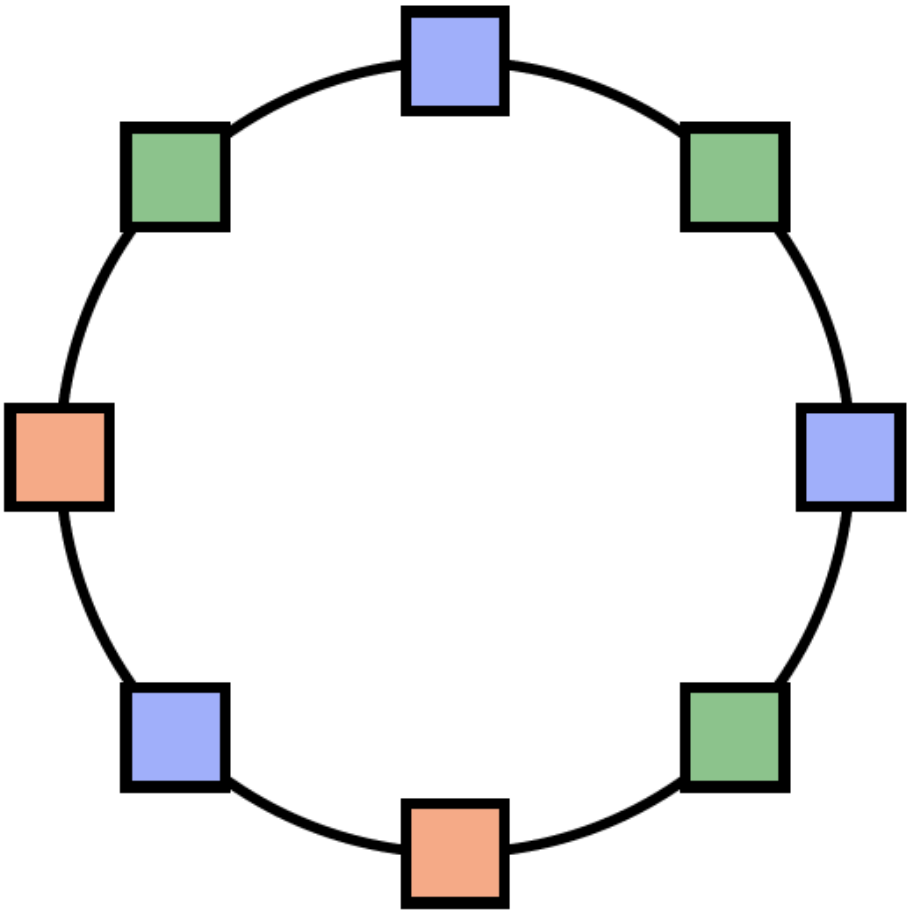
Para sua própria surpresa, a ideia de trabalhar não parece tão ruim. Você está juntando dinheiro para comprar uma GPU nova, pois continua convencido de que seu desempenho no Valorant é culpa do computador. Seus amigos discordam e afirmam que nenhuma placa de vídeo corrige problemas do nascimento, mas você ainda acredita no seu potencial.

Na manhã seguinte, você procura seu tio, que trabalha pintando casas e ainda mora com sua avó. Ele aceita sua ajuda: ele entra com a experiência como supervisor, e você carrega as latas e pinta.

Logo vocês são contratados para pintar um bairro com n casas dispostas em círculo, numeradas de 1 a n. Para cada $1 \leq i < n$, as casas i e $i+1$ são vizinhas.

A casa n também é vizinha da casa 1.

Os moradores têm uma exigência: **duas casas vizinhas não podem ter a mesma cor**. A regra surgiu depois de uma discussão sobre fachadas copiadas. Até hoje, mencionar “verde-abacate” causa desconforto nas reuniões do bairro.



Há m cores disponíveis, e cada casa deve receber exatamente uma delas. Seu tio prepara uma tabela em que $c_{i,j}$ representa o custo de pintar a casa i com a cor j.

Durante uma pausa para uma coquinha e um salgado, ele explica o acordo: o pagamento pelo serviço é fixo. Depois de descontar os custos da pintura, vocês dividirão igualmente o dinheiro que sobrar. Cada economia aproxima você da sua GPU nova. Com ela, talvez finalmente seja possível descobrir qual será sua próxima desculpa para errar todos os tiros.

Determine o menor custo total para pintar todas as casas respeitando a exigência dos moradores.

Entrada

A primeira linha contém dois inteiros n e m ($3 \leq n \leq 1000$ e $1 \leq m \leq 100$), que representam, respectivamente, a quantidade de casas e a quantidade de cores

disponíveis.

Cada uma das próximas n linhas contém m inteiros. O j-ésimo inteiro da i-ésima dessas linhas é $c_{i,j}$ ($1 \leq c_{i,j} \leq 10^6$), o custo de pintar a casa i com a cor j.

Saída

Imprima um único inteiro: o menor custo total para pintar todas as casas de modo que casas vizinhas tenham cores diferentes.

Caso não exista uma configuração válida, imprima `-1`.

Limites de execução

•**Tempo:** 1 segundo.
•**Memória:** 256 MB.

Exemplo 1

Entrada

```
4 3
1 5 5
5 1 5
5 5 1
1 3 5
```

Saída

```
6
```

Explicação

Uma configuração ótima utiliza, na ordem das casas, as cores 1, 2, 3 e 2. O custo total é:

$$1 + 1 + 1 + 3 = 6.$$

Todas as casas vizinhas têm cores diferentes, incluindo as casas 4 e 1.

Exemplo 2

Entrada

```
3 2
1 2
2 1
1 2
```

Saída

```
-1
```

Explicação

Em um círculo com três casas, cada casa é vizinha das outras duas. Portanto, não é possível pintá-las usando apenas duas cores sem que duas casas vizinhas tenham a mesma cor.

Tempo de execução máximo por teste: 1 s
Data de entrega: 20/11/2026
Entregas atrasadas sem penalidade até: 27/11/2026

Importante:

•O código tem que estar em linguagem C.
•Não olhem códigos de outros grupos ou da internet, exceto aqueles fornecidos.
•Os trabalhos poderão ser feitos individualmente ou em duplas. No caso de duplas, apenas um dos integrantes será responsável pelas submissões. O responsável deverá colocar, na primeira linha do código, o RA e o nome (exatamente conforme cadastrado na DAC).
•TODOS os membros do grupo devem participar e compreender completamente a implementação.
•Em caso de plágio, fraude ou tentativa de burlar o sistema, será atribuída nota 0 na disciplina aos envolvidos.
•Alguns alunos poderão ser solicitados a explicar, em detalhes, a implementação.
•Passar em todos os testes **não é garantia de obter a nota máxima**. A nota também dependerá do cumprimento das especificações do trabalho, da qualidade do código, da clareza dos comentários, das boas práticas de programação e do entendimento da matéria demonstrado em uma possível reunião.

•O responsável pela submissão deverá enviar, até a data de entrega, o código na plataforma **runcodes.hokama.com.br**.
•Seu código deverá executar dentro do tempo limite estabelecido para cada caso de teste.
•É esperado, e faz parte do aprendizado, que vocês tenham algumas dificuldades, como falhas de segmentação, loops infinitos, códigos sem a eficiência necessária etc. Portanto, não deixem para os últimos dias!
•Código de Honra: O uso de ferramentas de IA generativa não é recomendado. Caso o aluno opte por utilizá-las, compromete-se a tentar primeiro resolver o problema por conta própria, sem o auxílio de ferramentas de IA generativa, antes de recorrer a elas. Além disso, o aluno deve compreender completamente o código gerado ou sugerido pela IA e ser capaz de explicar sua implementação.