



Aquecimento

Tempo limite: 1000 ms — Memória limite: 256 MB

@rafayel

Bem-vindas à primeira Maratona Feminina do IFPB!

Este é o problema de boas-vindas, e funcionará como aquecimento para os demais desafios desta prova. Sua missão é saudar calorosamente todas as participantes deste grande evento.

Portanto, basta imprimir a mensagem:

```
BEM-VINDAS A MARATONA FEMININA DO IFPB!!!
```

Entrada

Este problema não possui entrada.

Saída

A saída do seu programa deve ser uma única linha contendo a mensagem de boas-vindas, conforme especificado. A mensagem deve ser EXATAMENTE a seguinte:

```
BEM-VINDAS A MARATONA FEMININA DO IFPB!!!
```

Certifique-se de que a mensagem esteja em letras maiúsculas e garanta que possua apenas uma quebra de linha ao final.

Exemplos

Entrada 1

Saída 1

```
BEM-VINDAS A MARATONA FEMININA DO IFPB!!!
```



Bot do POP

Tempo limite: 1000 ms — Memória limite: 256 MB

@rafayel, rev by @thiagogouveia e @stefan

Valéria, muito astuta, fez um concurso entre bots para escolher o nome do novo bot do POP.

Bard? Nam... GPT? Watson? Nem... O vencedor foi: C3P-POP. (risos)

As inscrições para Maratona POP 2023 estão abertas e a MAIOR das preocupações é que haja coffee break para todos!! Então, a primeira missão do C3P-POP é bem simples:

Dado o número de kits de coffee break comprados (cada kit alimenta exatamente uma pessoa) e o número de times inscritos até aqui (com exatamente 3 pessoas por time), C3P-POP deve manter as inscrições abertas ou fechá-las imediatamente?

Entrada

A entrada é composta por duas linhas.

A primeira linha da entrada traz um número inteiro N , representando o número de kits de coffee break comprados.

A segunda linha apresenta o número inteiro T , representando o número de times inscritos.

Considere $(30 \leq N \leq 100)$ e $(0 \leq T \leq 30)$.

Saída

Caso o número de kits comprados permita que mais times se inscrevam (e todos do time sejam alimentados, logicamente), seu programa deve imprimir **MANTER ABERTAS**

Caso o número times já tenha passado do limite (itxa.... times vão ficar com fome) ou tenha chegado exatamente ao limite, seu programa deve imprimir **FECHAR IMEDIATAMENTE**

Não esqueça de imprimir tudo em letras maiúsculas.

Exemplos

Entrada 1

42
30

Saída 1

FECHAR IMEDIATAMENTE

Entrada 2

45
7

Saída 2

MANTER ABERTAS



Contando Balões

Tempo limite: 1000 ms — Memória limite: 256 MB

@stefan, rev by @thiago.gouveia

É interessante que muitas maratonas de programação ocorram durante o ano, e um dos principais componentes para que elas aconteçam são os balões.

Por isso, sempre que possível, balões não utilizados em uma maratona são utilizados nas seguintes.

Nas maratonas que acontecem no IFPB, a professora Valéria faz questão de garantir que, ao início da maratona, a quantidade de balões disponíveis é o mínimo suficiente, considerando que todos os times resolverão todos os problemas!.

Sua tarefa é ajudá-la a contar a quantidade de balões que não foram distribuídos para os times em uma das maratonas realizadas no IFPB.

Entrada

A primeira linha da entrada possui dois inteiros, n ($2 \leq n \leq 50$) e q ($6 \leq q \leq 13$) — o número de times que participaram da maratona e a quantidade de problemas.

A segunda linha contém n inteiros $a_1, a_2, a_3, \dots, a_n$ ($1 \leq a_i \leq q$) — a quantidade de balões que o i -ésimo time recebeu.

Saída

Seu código deve imprimir um único número inteiro — a resposta do problema.

Exemplos

Entrada 1

```
5 7
7 7 7 7 7
```

Saída 1

```
0
```

Entrada 2

```
3 6
4 5 6
```

Saída 2

```
3
```

Entrada 3

```
5 7
5 6 2 3 4
```

Saída 3

15

Explicação

No primeiro caso, cada um dos 5 times resolveu os 7 problemas da maratona, portanto, todos os balões foram distribuídos. Maravilha! :).

No segundo caso, dos três times, o terceiro resolveu os 6 problemas da maratona, o segundo resolveu 5, e o primeiro resolveu 4. Então 3 balões não foram distribuídos do total disponível previamente.

No terceiro caso, 5 times fizeram uma prova contendo 7 problemas. O total de balões não distribuídos foi 15. Na ordem do primeiro para o quinto time, faltaram ser distribuídos $2 + 1 + 5 + 4 + 3 = 15$.



Dividindo as Cartas

Tempo limite: 1000 ms — Memória limite: 256 MB

@thiago.gouveia, rev by @kerven e @stefan

Após a Maratona de Programação da SBC de 2023, Andressa e Bianca ficaram tristes e felizes com o resultado. Tristes por não terem conseguido ficar em primeiro lugar do mundo, mas felizes por terem conseguido resolver 2 problemas.

Por isso, após a competição, resolveram estreitar os laços e aprofundar sua amizade. Todos sabemos que a melhor forma de fazer isso é com jogos de cartas! Logo colocaram Kerebe e Calven para cozinhar e começaram o jogo.

As regras são simples... Primeiro, elas pedem para um dos rapazes dizer um número N , entre 1 e 53, o número total de rodadas. Depois, de forma alternada, Andressa e Bianca, nessa sequência, pegam uma carta cada, até que as duas tenham pegado N cartas (somando as cartas das duas!). No final, ganha quem somou mais pontos.

Porém, duas regras foram adicionadas para deixar o jogo mais emocionante... Quem tira uma carta A tem os seus pontos zerados. Quem tira a carta coringa C ganha imediatamente.

Sua tarefa é simular o jogo e dizer quem venceu!

Entrada

A primeira linha da entrada possui um número inteiro N ($1 \leq N \leq 53$).

Seguem N linhas, cada uma representando uma carta, na sequência que foram pegadas por Andressa ou Bianca. Observe que Andressa sempre começa, alternando a vez com Bianca.

Cada carta é representada por exatamente uma letra maiúscula A, X, J, Q, K, C ou por um dígito 2, 3, 4, 5, 6, 7, 8, 9.

Caso a carta seja um dígito, ela vale exatamente o que está escrito.

Caso seja um A ou um C , ela faz exatamente o que está descrito na definição do problema.

Quaisquer outra letra vale 10.

Saída

Seu programa deve imprimir:

- ANDRESSA, caso Andressa vença
- BIANCA, caso Bianca vença
- ou EMPATE, caso ambas cheguem ao final do jogo com a mesma quantidade de pontos

Não esqueça de imprimir tudo em letras maiúsculas.

Exemplos

Entrada 1

5
5
Q
3
4
7

Saída 1 ANDRESSA
Entrada 2 6 X 5 6 7 A 5
Saída 2 BIANCA

Explicação

Na primeira entrada, a primeira linha indica que teremos 5 cartas no jogo.

Andressa começa, pegando a carta 5, totalizando 5 pontos.

Segue Bianca, que saca um Q, totalizando 10 pontos.

Na sua segunda jogada, Andressa consegue um 3, totalizando $5 + 3 = 8$ pontos.

Na sua segunda jogada, Bianca pega um 4, totalizando $10 + 4 = 14$ pontos.

Na jogada final, Andressa puxa um 7, chegando a $5 + 3 + 7 = 15$ pontos.

Como Andressa conseguiu 15 pontos e Bianca apenas 14, o programa imprime ANDRESSA.

Na Entrada 2, a primeira linha indica que teremos 6 cartas no jogo.

Como sempre, Andressa começa, pegando a carta X, totalizando 10 pontos.

Segue Bianca, pegando a carta 5, totalizando 5 pontos.

Andressa: $10 + 6 = 16$

Bianca: $5 + 7 = 12$

Como Andressa pega um A, sua pontuação volta para o zero!!!

Por fim, Bianca pega um 5, totalizando $5 + 7 + 5 = 17$, e vence por 17 a 0.



Embaralhada

Tempo limite: 1000 ms — Memória limite: 256 MB

@stefan, rev by @thiago.gouveia

A professora Valéria escreveu "*passou*" algumas vezes em um arquivo de texto no computador do POP. A quantidade de vezes foi o número de problemas que o time do IFPB resolveu em uma das maratonas de programação que participou.

Por exemplo, se o time resolveu 4 problemas, foi escrito: "*passoupassoupassoupassou*".

Stefan chegou no laboratório do POP, acessou o computador e viu o arquivo "*passou.txt*". Pensando ser um dos arquivos de Rugal, ele decidiu bagunçar com a ideia do professor.

Então ele escolheu uma das letras de "*passou*" e a considerou *fixa*. Em seguida, executou o seguinte procedimento algumas vezes:

1. Escolha qualquer letra do alfabeto latino exceto a *fixa*.
2. Coloque a letra escolhida no final do texto atual.
3. Embaralhe o texto.

Obviamente que ele utilizou um script em python para fazer isso.

Mais tarde, ele recebe uma ligação de Rugal.

— Fala Stefinha, tudo bem? Rapaz, alguém mexeu nos arquivos de Valéria aqui no computador do POP... Tu tem ideia de quem foi?

— Não, professor, não faço ideia...

— Pois é, vamos ter que ir atrás de quem fez isso...

— ...

Stefan sabe que fez besteira, mas ele se lembra do formato inicial do arquivo. Então foi, com cautela, até o computador no laboratório do POP e pegou uma cópia do arquivo modificado.

Ajude-o a sair dessa e evitar que descubram sua autoria, e principalmente, evitar que a professora Valéria fique brava com ele.

Você pode ajudá-lo informando a quantidade de problemas que o time do IFPB resolveu na maratona, dessa forma ele restaura o arquivo facilmente.

Entrada

A entrada consiste de uma string *s* contendo o texto embaralhado presente no arquivo copiado por Stefan.

Considere que *s* possui no mínimo 1 caractere e no máximo 1000.

Considere que *s* contém apenas caracteres minúsculos do alfabeto latino.

Saída

Seu programa deve imprimir um único número inteiro — a quantidade de problemas que o time do IFPB resolveu na maratona.

Exemplos

Entrada 1

```
facaumasubmissao
```


Saída 1 0
Entrada 2 uxoyszsqakp
Saída 2 1
Entrada 3 paxxouypsasouyss
Saída 3 2

Explicação

No primeiro caso, o time não *passou* problema algum.

No segundo caso, é possível notar que o time *passou* um único problema.



Fortalecendo Amizades

Tempo limite: 1000 ms — Memória limite: 256 MB

@rafayel, rev by @thiagogouveia e @stefan

Waléska é vice presidente do WIE (*Womens In Engineering*) e está organizando um Amigo Secreto para promover amizade e companheirismo entre as mulheres da área de engenharia.

Para agilizar a organização do evento, ela decidiu utilizar o C3P-POP para gerar a lista de quem entregará presentes a quem durante o evento. No entanto, o C3P-POP apresentou alguns bugs recentes devido a sua grande carga de trabalho durante as inscrições da maratona POP.

Portanto, para evitar que a lista utilizada não siga as regras básicas de um Amigo Secreto, Waléska decidiu pedir sua ajuda para criar um programa que verifica se a lista de presentes gerada pelo C3P-POP é válida ou não.

Seguem as regras:

- Nenhuma participante entregará um presente a si mesma;
- Cada participante receberá EXATAMENTE um presente.

Entrada

A primeira linha da entrada contém um número inteiro N ($1 \leq N \leq 10^5$), representando a quantidade de participantes do Amigo Secreto.

A segunda linha contém N números inteiros (a_1, a_2, \dots, a_N) , indicando que a i -ésima participante entregará um presente para a participante a_i .

Saída

Seu programa deve imprimir:

- S, caso a lista cumpra os dois requisitos.
- N, caso contrário.

⚠ Não esqueça de imprimir tudo em letras maiúsculas.

Exemplos

Entrada 1

```
5
3 2 1 5 4
```

Saída 1

```
N
```

Entrada 2

```
5
5 3 2 1 4
```

Saída 2

S

Explicação

No primeiro caso, as entregas de presentes acontecem da seguinte maneira:

- participante 1 presenteia a participante 3
- participante 2 presenteia a participante 2
- participante 3 presenteia a participante 1
- participante 4 presenteia a participante 5
- participante 5 presenteia a participante 4

Como a participante 2 entrega um presente a si mesma, o primeiro requisito não é cumprido.

No segundo caso, as entregas de presentes acontecem da seguinte maneira:

- participante 1 presenteia a participante 5
- participante 2 presenteia a participante 3
- participante 3 presenteia a participante 2
- participante 4 presenteia a participante 1
- participante 5 presenteia a participante 4

Portanto, os dois requisitos são cumpridos.



Ge...Ge...Geometria?

Tempo limite: 1000 ms — Memória limite: 256 MB

@rafayel e @stefan, rev by @thiagogouveia

Isabelle, integrante do POP e bem criativa, decidiu utilizar um padrão de azulejos triangulares para um projeto, com diferentes cores e tamanhos. Consultou vários locais de venda de azulejos até encontrar um que entregasse as peças no padrão desejado.

Cada peça de azulejo, inicialmente em formato quadrangular, deveria ser cortada para atender o padrão triangular de Isabelle. Com sua mente de engenheira, e tendo em mãos o catálogo disponibilizado pelo vendedor, ela calculou para cada peça qual seria a maior área triangular possível de se obter entre os mais variados tipos de corte.

Entrada

Na única linha da entrada há um número inteiro L ($2 \leq L \leq 10^4$) — o tamanho do lado de um dos azulejos presentes no catálogo do vendedor.

É garantido que L é par.

A unidade de medida do tamanho do lado do azulejo é irrelevante para o problema. Caso queira, você pode considerar que o tamanho do lado é dado em mm .

Saída

Seu código deve imprimir um único número inteiro — a área calculada por Isabelle.

Exemplos

Entrada 1

68

Saída 1

2312

Entrada 2

528

Saída 2

139392



Hamilton

Tempo limite: 1000 ms — Memória limite: 256 MB

@rafayel, rev by @thiagogouveia e @stefan

Não, infelizmente esse problema não é sobre o piloto heptacampeão da Fórmula 1, mas é sobre algo tão emocionante quanto.

Empolgada com o sucesso do primeiro Amigo Secreto, Kamilly, presidente do WIE (Women in Engineering), decidiu realizar um novo evento, mas dessa vez com algumas modificações para promover ainda mais conexões.

A primeira é que cada participante pode entregar um presente apenas para alguém que não conhecia antes do evento.

Além dessa modificação, Kamilly, seguindo o sábio conselho do renomado programador russo Igor Stefanovski, discípulo do mestre dos grafos Thiago Rugal, definiu que a sequência de troca de presentes deverá formar um ciclo hamiltoniano.

Isto significa que todas as participantes vão entregar e também receber um presente, e que a última pessoa que receberá presente será aquela que realizou a primeira entrega.

Portanto, além das restrições originais do Amigo Secreto:

- Nenhuma participante entregará um presente a si mesma;
- Cada participante receberá EXATAMENTE um presente.

Foram introduzidas as seguintes regras adicionais:

- A troca de presentes só é permitida entre participantes que não se conhecem;
- A primeira participante a realizar a entrega do presente será a última a receber.

Porém, antes de confirmar o evento, Kamilly precisa que você verifique se é possível realizar o Amigo Secreto como planejado. Para isso, serão disponibilizados o número de participantes (N), o número de relações de amizade entre as participantes (M) e a descrição destas relações.

Entrada

A primeira linha do programa possui um número N ($1 \leq N \leq 10$) — a quantidade de participantes.

A segunda linha possui um número M ($0 \leq M \leq \frac{N \times (N-1)}{2}$) — a quantidade de relações de amizade.

Cada uma das próximas M linhas é composta por dois inteiros distintos x e y ($1 \leq x < y \leq N$), que descreve que a participante com índice x já conhecia a participante de índice y antes do evento.

Observe que esta relação é recíproca: se x já conhecia y , y também já conhecia x .

Saída

Seu programa deve imprimir:

- S, caso seja possível criar uma sequência de entrega de presentes que cumpra todos os requisitos.
- N, caso contrário.

Não esqueça de imprimir tudo em letras maiúsculas.

Exemplos

Entrada 1

3 3 1 2 2 3 1 3
Saída 1 N
Entrada 2 3 1 1 2
Saída 2 N
Entrada 3 3 0
Saída 3 S

Explicação

No primeiro caso, todas as participantes se conhecem, então não há sequência válida.

No segundo caso, a participante 1 não pode receber nem entregar presente para a participante 2, então não há como fechar o ciclo.

No terceiro caso, nenhuma das participantes se conhecem, então qualquer sequência de troca de presentes é válida.



Indignada

Tempo limite: 1000 ms — Memória limite: 256 MB
@thiago.gouveia, rev by @stefan e @rafayel

Nem tudo são flores na Universidade. Como já era de se esperar, tanta dedicação às disciplinas de tecnologia (programação em especial) cobraram seu preço. H3llena não obteve pontuação suficiente para escapar da final de Álgebra. Ela ficou Indignada!

Mas, aos 45 do segundo tempo, surge um fio de esperança! Um dia antes da final a professora lançou um desafio, e quem resolvesse seria automaticamente aprovado. Será que H3llena conseguirá se safar?

O desafio era bem simples: dados 25 números inteiros positivos, seria possível organizá-los em uma matriz 5×5 de modo que a soma de cada linha seja um número ímpar e a soma de cada coluna seja também um número ímpar?

Entrada

A entrada possui apenas uma linha contendo exatamente 25 inteiros, separados por espaço. Nenhum inteiro é menor que 1 nem maior que 99.

Saída

Caso seja possível organizar os números da entrada como pedido, seu programa deve imprimir YES. Caso contrário seu programa deve imprimir NO.

Exemplos

Entrada 1

16 21 3 2 10 10 8 4 2 18 8 2 9 16 10 22 4 22 8 15 4 17 18 18 16

Saída 1

YES

Entrada 2

12 12 2 8 8 21 18 20 20 18 20 2 15 2 14 14 6 6 8 8 7 16 10 13 8

Saída 2

NO



J4mpa Game Girls on Bet

Tempo limite: 1000 ms — Memória limite: 256 MB

@thiago.gouveia, rev by @stefan e @rafayel

Nos períodos finais do seu curso, H3llena se dedicou intensamente aos seus estudos, principalmente no desenvolvimento de projetos.

Ela aprimorou suas habilidades em diversas linguagens de programação, especialmente Kotlin, Java, C++, Python, Haskell, Scala, Go, Rust, Lua, Javascript, Typescript, Bash, C++, Fortran, Ruby, C, C# e C++, porque C++ nunca é demais.

Ela trabalhou em programas cada vez mais complexos e desafiadores, mas também participou de inúmeros eventos de tecnologia e redes, conhecendo profissionais de diferentes áreas da academia e do mercado.

Em um desses eventos, H3llena conheceu a KeyTh-Lin, CEO da famosa empresa **J4mpa Game Girls on Bet**. KeyTh-Lin comentou que estava tendo dificuldade para computar os ODDs (jeito diferente de falar das probabilidades) de uma aposta com cartas. Mas você “sacou” a resposta na hora, já abriu o notebook e começou a “codar” junto da CEO. Em poucos minutos o problema estava resolvido. Anos depois você está tentando lembrar a sua solução.

No problema são dadas 3 pilhas de cartas, espalhadas de modo que qualquer carta pode ser escolhida de maneira equiprovável entre aquelas da mesma pilha. Cada carta tem um número (entre 2 e 9, inclusive), e uma cor (R ou B). O jogador pega exatamente uma carta de cada pilha e ganha se as 3 cartas forem da mesma cor. Dadas as cartas das 3 pilhas, qual a probabilidade de vitória do jogador?

Entrada

A primeira linha da entrada traz três números inteiros A, B, C ($3 \leq A, B, C \leq 10$), indicando o número de cartas nas pilhas 1, 2, e 3, respectivamente.

Seguem 3 linhas:

- A primeira com A cartas, separadas por espaço, representando a pilha 1;
- A segunda com B cartas, separadas por espaço, representando a pilha 2; e
- A terceira com C cartas, separadas por espaço, representando a pilha 3.

Cada carta é composta por um número seguido (sem espaço) por uma letra R ou B, representando sua cor.

Saída

A saída deve ser uma fração indicando a probabilidade de vitória do jogador.

Esta fração deve ser a menor possível (simplificada), e caso a probabilidade seja um inteiro (ou zero) esta não deve ser impressa em forma de fração.

Para que fique claro, caso o numerador da fração resultante seja divisível por seu denominador, o resultado dessa divisão deve ser impresso.

Exemplos

Entrada 1

```
4 4 4
5B 9R 2R 5B
6B 7B 8R 8B
9B 6R 9R 8R
```


Saída 1

3/16

Entrada 2

6 5 4
7R 6R 9R 4B 4B 3R
4R 7R 9R 7R 7B
2R 8R 6B 9R

Saída 2

5/12