



IX - Maratona POP Superior

26 de novembro de 2019

(Este caderno contém 11 problemas)

A PROVA TERÁ DURAÇÃO DE **TRÊS HORAS**

LEIA ATENTAMENTE AS INSTRUÇÕES ABAIXO ANTES DE INICIAR A PROVA

- A prova deve ser realizada em equipes de três pessoas;
- Observem o nome do arquivo que deve ser enviado para cada problema;
- Cada questão tem um tempo limite para execução;
- É permitido consultar material impresso durante a prova;
- Não é permitida a consulta de qualquer material online;
- Este caderno de tarefas é composto de 15 páginas (incluindo esta);
- Verifique se o caderno está completo.

Criado por: @thiago.gouveia

Com revisão de @wesley

Problema A – Ansiosamente Mariquinha

Nome do Programa: mariquinha.(c|cpp|py|java)

Tempo: 1 segundo

Mariquinha era uma menina do interior. Criada em um estilo de vida simples, rodeada pela natureza e pela vida pacata que só as cidades distantes dos grandes centros comerciais e empresariais podem proporcionar.

Quando passou no vestibular de medicina, Mariquina precisou se mudar para a cidade grande. Foi aí que tudo desandou. Mariquinha conheceu o Tw0tter e ficou ansiosamente viciada em verificar se alguém tinha falado algo sobre dela.

Compadecido com a situação da Mariquinha, você resolveu ajuda-la. Para isso, você escreveu um programa que recebe como entrada todas as postagens do Tw0tter do dia corrente e retorna o número de postagens que falou sobre a Mariquinha.

Entrada

A entrada possui apenas um caso de teste. A primeira linha da entrada tem um inteiro P ($1 \leq P \leq 50$), indicando o número de postagens feitas no Tw0tter no dia a ser processado. Seguem P linhas, cada uma representando uma postagem feita no Tw0tter, sendo que cada postagem pode ter no máximo 1024 letras maiúsculas, sinais de pontuação e espaços.

Saída

Dada a entrada conforme descrito acima, seu programa deve imprimir o número de postagens na quais Mariquinha foi citada, isto é, o número de linhas que possuem a palavra MARIQUINHA.

Casos de Teste

Entrada 1	Saída 1
7 MARIQUINHA FOTOGRAFOU AS CARTAS DE MARIQUINHA A TIA DO VIZINHO VIU O JORNAL XUAO OBSERVOU DE MODO QUE PAPITO BATEU NA CASA A GAROTA ANALISOU O JARDIM A TIA DO VIZINHO ASSISTIU AO PROGRAMA VOVO BATEU UM POUCO NAS CARTAS DE MARIQUINHA MARIQUINHA BATEU EM RETIRADA ONTEM	3
Entrada 2	Saída 2
4 MARIQUINHA MARIQUINHA MARIQUINHA MARIQUINHA MARIQUINHA MARIQUINHA MARIQUINHA M-ARIQUINHA M ARIQUINHA MARIQUINH A MARIQ UINHA	3

Criado por: @thiago.gouveia

Com revisão de @wesley

Problema B – Boa grana para Cheiço

Nome do Programa: cheque.(c|cpp|py|java)

Tempo: 1 segundo

Como reza um antigo ditado, a água sempre corre para o mar. O chefe de Cheiço, seu Omar, ganhou 10^{10} reais na loteria! Como uma pessoa de ótimo coração, ele resolveu dar um agrado ao seu melhor (e único) funcionário, no caso, Cheiço. Disse:

“Cheiço, você é único! Como ganhei muito money, vou te dar muito dinheiro!”

Mas dar dinheiro de bobeira assim para alguém não é da natureza do seu Omar, nem mesmo sendo para Cheiço. Então seu Omar lançou o seguinte desafio:

“Cheiço, táqui os dois cheques com seu pagamento desse mês (sim, Cheiço recebe quinzenalmente). Em um dos cheques, você pode escolher um número do valor que vou te pagar e mudar para qualquer outro número de sua escolha.”

Cheiço ficou preocupado. Não é sempre (na verdade esta foi a primeira vez, e provavelmente a última) que ele tem uma oportunidade para ganhar um dinheirinho a mais do seu Omar. Para não perder dinheiro, Cheiço pediu sua ajuda para calcular qual o maior valor que ele pode receber de pagamento segundo as regras do seu Omar.

Entrada

A entrada possui vários casos de teste. A primeira linha da entrada tem um inteiro T ($1 \leq T \leq 100$), indicando o número de casos de teste. Seguem T linhas, cada uma com dois inteiros $c1$ e $c2$ ($0 \leq c1, c2 \leq 10^7$)., separados por um espaço, representando o valor de cada um dos dois cheques do salário de Cheiço.

Saída

Para cada caso de teste, seu programa deve imprimir o maior valor em dinheiro que Cheiço poderá receber ao alterar exatamente um dígito de um dos cheques.

Casos de Teste

Entrada 1	Saída 1
5	18
9 9	198
90 99	910
100 10	1000
100 100	100008
99999 8	

Criado por: @thiago.gouveia

Problema C – Camila com C ou Kamila com K

Nome do Programa: camila.(c|cpp|py|java)

Tempo: 1 segundo

Na noite fria da última quinta-feira, durante o lual de aniversário do seu melhor amigo Victor, Eduardo e Mônica descobriram que estavam grávidos do seu terceiro filho. Na verdade, filha (é que havia uma cigana no lual e ela viu nas nuvens de fumaça formadas pela queima da borra do café de ontem que será uma menina).

Depois de muita discussão e de inúmeras sugestões, foi realizada uma votação e decidiram que o nome do bebê será Camila. O único problema que restou foi: Eduardo gostaria que o nome fosse Camila com C, enquanto Mônica prefere Kamila com K.

Mais uma vez a cigana Xyuxa deu a solução. Ela separou alguns dos seus baralhos (cada um com as cartas A, 2, 3, 4, 5, 6, 7, Q, J e K), embaralhou e deu cada uma das cartas em sequência, uma para Eduardo, uma para Mônica. Por fim, quem ficasse com mais cartas 'K' teria o direito de escolher o nome do bebê. Em caso de empate, Xuxya escolheria o nome.

Entrada

A entrada possui apenas um caso de teste. A única linha do caso de teste apresenta a sequência de cartas distribuída por Xyuxa. Haverá no mínimo 10 e no máximo 100 cartas e cada carta é representada por um caractere A, 2, 3, 4, 5, 6, 7, Q, J ou K. Não há espaços entre as cartas.

Saída

Para cada caso de teste, seu programa deve imprimir o nome escolhido para o novo bebê de Eduardo e Mônica.

Obs.: Xyuxa escolheu o belíssimo nome Jaricleidnya.

Casos de Teste

Entrada 1	Saída 1
KA234567QJ	Camila com C
Entrada 2	Saída 2
AK234567QJ	Kamila com K
Entrada 3	Saída 3
KKA234567QJA234567QJ	Jaricleidnya

Criado por: @thiago.gouveia

Problema D – D.E.A.B.C.

Nome do Programa: deabc.(c|cpp|py|java)

Tempo: 1 segundo

O grande Dr. Arqueólogo Strongman Vog acaba de fazer mais uma descoberta épica no litoral da Paraíba: junto ao esqueleto de um gigante dinossauro aquático, Vog e sua equipe encontraram os vestígios de uma antiga civilização extinta. Depois de uma análise meticulosa e detalhada da arquitetura piramidal desta civilização, Vog descobriu que seu sistema de classes sociais era baseado nas pirâmides D.E.A.B.C.

Cada pessoa nesta civilização possuía uma classe de 1 a 5, sendo a classe 1 a mais importante e a classe 5 a menos. Cada cidade desta civilização era composta por um conjunto de pirâmides e cada pirâmide podia ser acessada apenas por membros da classe social a qual a pirâmide pertencia. Cada pirâmide é construída usando até 5 tipos de tijolos, denominados D, E, A, B ou C. Segue exemplo de pirâmide:

```

      A
     C E
    E A E
   B E B C
    
```

Para subir ao topo da pirâmide, deve-se começar em qualquer tijolo do nível inferior e acessar o próximo nível, andando sempre para cima, para um tijolo diretamente à sua esquerda ou diretamente à sua direita. Por exemplo, a partir do tijolo 'E' do nível inferior, é possível acessar os tijolos 'E' ou 'A' do nível acima. Para determinar a classe da pirâmide, basta calcular o menor número de tipos de tijolos diferentes que possibilite a subida ao topo da pirâmide. Na pirâmide de exemplo, este número é 2: A e E.

Entrada

A entrada possui apenas um caso de teste. A primeira linha da entrada tem um inteiro N ($1 \leq N \leq 100$), indicando número de níveis da pirâmide. Seguem N linhas: n_1, n_2, \dots, n_n . A linha n_x representa o nível x da pirâmide (de cima para baixo) e apresenta os x tijolos que compõem este nível, separados por um espaço.

Saída

A saída deve ser apenas um número inteiro, representando a classe da pirâmide de entrada.

Casos de Teste

Entrada 1	Saída 1
4 A C E E A E B E B C	2

Criado por: @thiago.gouveia

Com revisão de @wesley

Problema E – É Primo Primo Primo Primo Primo ... ?

Nome do Programa: primo.(c|cpp|py|java)

Tempo: 1 segundo

Desde a criação do Universo (na verdade desde a antiguidade, mas vamos respeitar a minha licença poética) nós conhecemos a definição dos números primos:

“Um número x é dito primo se e somente se o conjunto de seus divisores possui exatamente dois elementos: 1 e x .”

Neste ponto, vale salientar que o número 1 não é primo, uma vez que o conjunto de seus divisores tem apenas um elemento, quando por definição são necessários dois.

Uma definição um pouco mais recente é a do número primo primo primo primo primo primo ... (por simplicidade p^p):

“Um número x é dito p^p se e somente se x é primo e o número y , obtido a partir da soma dos dígitos de x , também é p^p .”

A pergunta que surge é: dados dois números inteiros a e b , quantos números p^p existem no intervalo fechado entre a e b (isto é, incluindo os próprios a e b).

Entrada

A entrada possui vários casos de teste. A primeira linha da entrada tem um inteiro T ($1 \leq T \leq 10^3$), indicando o número de casos de teste. Seguem T linhas, cada uma com dois números inteiros a e b ($1 \leq a \leq b \leq 10^4$).

Saída

Para cada caso de teste, seu programa deve imprimir em uma linha os inteiros a , b e c separados por um espaço em branco, onde c representa a quantidade de números p^p que existem no intervalo fechado entre a e b .

Casos de Teste

Entrada 1	Saída 1
2 101 103 102 103	101 103 1 102 103 0
Entrada 2	Saída 2
8 13 19 10 29 2 12 2 11 1 11 1 12 3 23 3 8	13 19 0 10 29 3 2 12 5 2 11 5 1 11 5 1 12 5 3 23 5 3 8 3

Criado por: @thiago.gouveia

Com revisão de @wesley

Problema F – Final do top compras Brasil

Nome do Programa: final.(c|cpp|py|java)

Tempo: 2 segundos

O programa top compras Brasil na TV já foi famoso. Vale lembrar: neste programa o concorrente é deixado livre dentro de um supermercado durante certo período de tempo. Ao final deste período, o valor de tudo que o concorrente conseguiu colocar no carrinho é somado, e caso o total passe de determinado montante, o concorrente leva para casa o prêmio máximo. Caso contrário, o concorrente não leva nada.

Hoje o programa se encontra em profunda decadência. Sua audiência está baixíssima e seus seguidores não migraram do Orkut e nem da TV a tubo até os dias atuais. Boniclauderson, presidente da prestigiadíssima emissora RAS, teve uma ideia genial para salvar o programa. A cada três programas, um seria IMPOSSÍVEL de ganhar! Genial, não? Isso aumentaria imediatamente a lucratividade do quadro em 33%! A nova versão do programa recebeu o nome de Final do Top Compras Brasil.

O que Boniclauderson não esperava era que você, programador profissional de nível competitivo avançado, tivesse sido selecionado para participar do top compras Brasil. Seu plano é criar um aplicativo para verificar se o programa que você vai participar pode ou não pode ser vencido.

Entrada

A entrada possui apenas um caso de teste. A primeira linha da entrada tem dois inteiros N ($1 \leq N \leq 10^3$) e M ($1 \leq M \leq 10^5$). N representa, ao mesmo tempo, o número de itens diferentes no supermercado e o peso máximo que o carrinho pode levar, e M o valor que se deve alcançar para levar o prêmio máximo. Seguem N linhas, cada uma com dois inteiros: p e v ($1 \leq p, v \leq 10^3$), representando um item com peso p e valor v . É garantido que o supermercado tem estoque suficiente de cada item para que você possa levar quantos quiser.

Saída

Caso exista alguma combinação de itens cujo valor seja pelo menos igual a M e o peso não seja maior que N , o programa deve imprimir SIM. Caso contrário, deve imprimir NAO.

Casos de Teste

Entrada 1	Saída 1
5 16 1 1 2 7 3 8 4 10 5 12	NAO

Criado por: @thiago.gouveia

Com revisão de @wesley

Problema G – G238729 o planeta

Nome do Programa: g2387298.(c|cpp|py|java)

Tempo: 3 segundos

Há um pequeno planeta rochoso localizado nos ermos da Via Láctea. Seu nome: GRandInt. Aconteceu que no momento da concepção deste problema, seu nome foi G2387298.

Um primeiro fato interessante sobre este planeta é que o nome de todos os seus habitantes é um número inteiro. O segundo é que ele é habitado apenas por bactérias. Mais interessante ainda é que todo o planeta foi colonizado por apenas uma bactéria (sim, uma bactéria que se divide se transformando em duas. A primeira é dita pai e a segunda é dita filha. Ah, sim! Ela pode se dividir novamente depois).

O sistema criminal de G2387298 tem uma peculiaridade: se duas bactérias cometem um crime juntas, a bactéria mais jovem que originou as duas criminosas também deve ser presa. Na esma linha, para o sistema judicial de G2387298, considera-se que uma bactéria também originou a si própria. Ah, uma bactéria pode fazer dupla com ela mesma. 184476854546546, prefeito de G2387298, está achando o sistema de detecção de bactérias muito lento. Então ele contratou você para fazer um sistema mais rápido.

Entrada

A primeira linha da entrada tem dois inteiros **N** ($1 \leq N \leq 5000$) e **Q** ($1 \leq Q \leq 10000$). **N** representa o número de bactérias no planeta e **Q** o número de perguntas que serão feitas ao sistema. Segue uma linha com '--'. Seguem **N-1** linhas, cada uma com dois inteiros **p** e **f** ($0 \leq p, f \leq 10^8$), indicando que a bactéria **p** é pai da bactéria **f**. É garantido que toda bactéria (exceto uma) tem exatamente um pai. Segue uma linha com '--'. Seguem **Q** perguntas, uma por linha. Cada linha contém os inteiros **a** e **b**, nomes de duas bactérias.

Saída

Para cada pergunta, seu programa deve imprimir uma linha com os inteiros **a**, **b** e **c**, separados por um espaço. **c** é o nome da bactéria mais jovem que originou **a** e **b**.

Casos de Teste

Entrada 1	Saída 1
5 3	6 7 9
--	8 8 8
100 9	8 6 100
100 8	
9 7	
9 6	
--	
6 7	
8 8	
8 6	

Criado por: @thiago.gouveia

Problema H – Honestidade é o seu nome

Nome do Programa: honestidade.(c|cpp|py|java)

Tempo: 2 segundos

Antigamente, muitos anos antes do nascimento do seu tataravô, em algum momento próximo ao início dos tempos para a raça humana, existiu uma pequena vila chamada *LoopString*. Nesta pequena vila, também havia um ladrão chamado *LoopString*. *LoopString* ficou famoso por dois motivos:

1 – Por sua completa falta de honestidade.

2 – Por escolher suas vítimas com base nos *loops* que se formam em seus nomes.

Sempre antes de assaltar alguém, *LoopString* perguntava seu nome e calculava qual a menor distância entre duas letras iguais (*LoopDistance*). Exemplo: OOLONEIA possui distância 1, pois possui uma letra 'o' na posição 1 e outra na posição 2. ABAXIAL, por sua vez possui distância 2 ('a' nas posições 1 e 3). *LoopString* só assaltava pessoas cujo nome possuísse a *LoopDistance* par.

Entrada

A entrada contém vários casos de teste. A primeira linha da entrada tem um inteiro T ($1 \leq T \leq 250$), indicando o número de casos de teste. Seguem T linhas, cada uma representando um caso de teste. Cada caso de teste é o nome de uma pessoa que *LoopString* cogitou assaltar, composto por no mínimo 1 e no máximo 500 letras maiúsculas.

Saída

Para cada caso de teste, seu programa deve imprimir o número do caso (Caso #:), seguido pelo nome que está sendo avaliado, seguido pela a menor distância entre duas letras iguais no nome em questão.

Caso o nome não possua nenhuma letra repetida, o seu programa deve considerar que a *LoopDistance* é -1.

Casos de Teste

Entrada 1	Saída 1
5	Caso 1: ABCD -1
ABCD	Caso 2: MARIA 3
MARIA	Caso 3: PERNAMBUCO -1
PERNAMBUCO	Caso 4: BRASIL -1
BRASIL	Caso 5: MARIANNA 1
MARIANNA	

Criado por: @thiago.gouveia

Com revisão de @wesley

Problema I – Imperadores e o Dratado de Dortesilhas

Nome do Programa: dratado.(c|cpp|py|java)

Tempo: 1 segundo

É sabido por todos os grandes estudiosos de História da história que os dois grandes conquistadores da era naval foram os magníficos impérios de Eortugal e da Pspanha. O que poucos sabem é que esta era de glórias só foi possível pois os dois imperadores possuíam um acordo secreto, o famigerado Dratado de Dortesilhas.

Segundo a epístola, a cada nova ilha (ou continente) descoberta por qualquer navegador de uma destas superpotências, os chefes de estado devem sentar e definir uma linha vertical. Toda ilha (ou continente) que estiver à esquerda desta linha pertencerá a Pspanha, enquanto que tudo à direita será regido por Eortugal.

Como se pode esperar, a escolha desta linha é determinante para o controle total da ilha (ou continente), visto que o imperador com mais ilha terá mais recursos e poderá, no futuro, dominá-la. Sua tarefa é escrever um programa que calcule qual império ficou com o maior pedaço da ilha (ou continente).

Como os dados são muito antigos, você não conseguiu um mapa da ilha (ou continente), mas sim um conjunto de pontos que fazem parte desta. A ilha (ou continente), então, é o menor polígono que contém todos estes pontos.

Entrada

A entrada contém apenas um caso de teste. A primeira linha apresenta dois inteiros N ($3 \leq N \leq 100$) e L ($0 \leq L \leq 120$), indicando, respectivamente, o número de pontos e a coordenada x da linha que dividirá a ilha (ou continente).

Cada uma das próximas N linhas traz dois inteiros x_n e y_n ($0 \leq x_n, y_n \leq 120$), que representam um ponto da ilha (ou continente).

Os pontos não são dados em nenhuma ordem específica e é garantido que não há três pontos colineares entre si.

Saída

Caso a área à esquerda da linha horizontal seja maior ou igual que a área à direita, o programa deve imprimir PSPANHA. Caso contrário, deve imprimir EORTUGAL.

Casos de Teste

Entrada 1	Saída 1
3 10 1 5 11 1 11 11	PSPANHA

Entrada 2	Saída 2
3 3 11 1 1 5 11 11	EORTUGAL
Entrada 3	Saída 3
4 15 10 10 30 10 20 50 20 30	EORTUGAL
Entrada 4	Saída 4
5 1 3 1 2 5 5 5 4 4 3 9	EORTUGAL
Entrada 5	Saída 5
5 10 3 1 2 5 5 5 4 4 3 9	PSPANHA
Entrada 6	Saída 6
5 3 3 10 3 5 5 15 7 1 7 10	EORTUGAL
Entrada 7	Saída 7
5 7 7 10 7 1 5 15 3 10 3 5	PSPANHA

Entrada 8	Saída 8
5 7 7 10 7 1 5 15 3 10 3 5	EORTUGAL
Entrada 9	Saída 9
10 7 2 4 2 8 4 1 4 10 5 5 6 1 6 10 8 4 8 8 7 6	PSPANHA

Criado por: @thiago.gouveia

Com revisão de @wesley

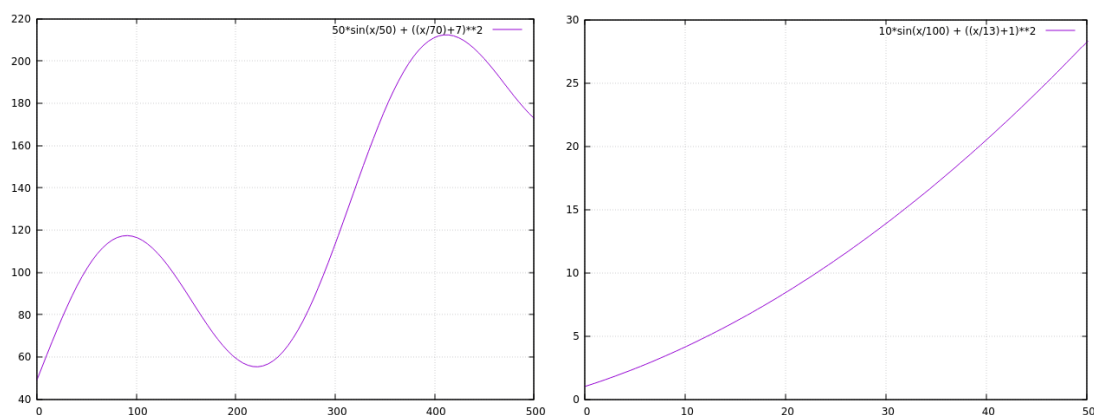
Problema J – Jato da Escapada não trivial

Nome do Programa: jato.(c|cpp|py|java)

Tempo: 1 segundo

Sempre que vê uma moto, Timerinho, filho de Timério, fica com muito medo e vai correndo para casa. Isso nós sabemos desde a seletiva POP-Superior-Avançado-2019. A novidade é que Timerinho também adota esta postura quando usa outros meios de transporte. Sempre que anda de avião, Timerinho fica com medo de OVNI's e vai da origem para o destino por um caminho não trivial.

De forma resumida, de acordo com Timerinho, os OVNI's esperam que ele use o menor caminho entre a sua origem e seu destino: uma linha reta. Então, para não ser pego pelos (supostos) OVNI's, Timerinho varia a altitude do seu avião entre o ponto inicial e o ponto final da trajetória. As figuras abaixo exibem dois caminhos feitos por Timerinho na semana passada. O eixo x representa o tempo de voo, enquanto que o eixo y representa a altitude do avião. Não se preocupe se o avião alcança alturas negativas! O zero não representa o solo, mas um valor seguro relativo à posição da torre de controle.



O problema de Timerinho é que ele não percebeu que variar a altitude aumenta o tamanho do trajeto percorrido, gastando, assim, mais combustível. Como um bom controlador de voo, sua tarefa é descobrir se o plano de voo de Timerinho é possível com o combustível que ele tem no avião ou se ele vai cair (neste caso, você pode avisá-lo antes, salvando sua vida).

Entrada

A entrada contém apenas um caso de teste, que é descrito em duas linhas. O caso de teste é composto pelo tempo que o avião passa sobre seu controle, pela distância máxima que o avião pode percorrer dado o combustível que ele possui inicialmente e pelo plano de voo. Segundo o plano de voo de Timerinho, a altitude do avião a cada momento é descrita pela seguinte função (com o argumento do seno em radianos):

$$y = a * \text{seno}(x/b) + [(x/c) + d]^2,$$

tal que a , b , c e d são números inteiros e $(-600 \leq a, b, c, d \leq 600, b \neq 0, d \neq 0)$.

A primeira linha da entrada apresenta dois números inteiros **T** e **C** ($-20000 \leq T, C \leq 20000$), representando, respectivamente, o tempo que o avião passa sob o seu controle e a distância máxima que o avião pode percorrer dado o combustível que ele possui no ponto **x=0**. A segunda linha apresenta os números **a**, **b**, **c** e **d** separados por um espaço em branco.

Saída

Caso o combustível seja suficiente para o caminho proposto por Timerinho, o programa deve imprimir a palavra **SAFE**. Caso contrário, deve imprimir a palavra **CRASH**.

Casos de Teste

Entrada 1	Saída 1
500 612 50 50 70 7	CRASH
Entrada 2	Saída 2
50 60 10 100 13 1	SAFE

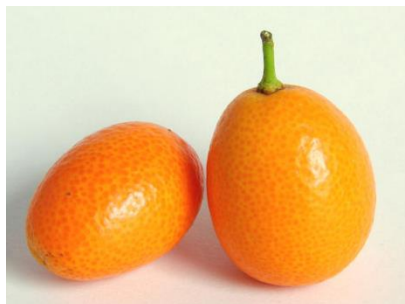
Nota: Os casos de teste acima representam os gráficos exibidos na descrição do problema.

Criado por: @thiago.gouveia

Problema K – Kumquat

Nome do Programa: kumquat.(c|cpp|py|java)

Tempo: 1 segundo



A Citrus Japônica (conhecida na China como kumquat e no Japão como kinkan) é uma árvore cujo fruto se parece muito com uma laranja. A maior produção brasileira deste fruto vem da encantadora cidade de Somsispaia, no coração da região mais fria de POPIOÓPOLIS, um destino turístico romântico, tranquilo e muito charmoso (tudo o que você precisa neste inverno).

Sen-Barui, dono de um grande campo de produção de kinkan, está tendo dificuldades para calcular a produção deste ano. Você poderia ajudá-lo?

Entrada

A entrada contém apenas um caso de teste, que é descrito em duas linhas. Na primeira linha há um número inteiro **N** ($3 \leq N \leq 100$), indicando o número de árvores na plantação de Sen-Barui. Na segunda linha há outro número inteiro, **K** ($3 \leq K \leq 100$) que representa o número de frutos que cada árvore produz em um ano.

Saída

A saída deve ser o número de frutos produzido na plantação de Sen-Barui em um ano.

Casos de Teste

Entrada 1	Saída 1
5 5	25
Entrada 2	Saída 2
3 100	300
Entrada 3	Saída 3
100 17	1700