

Caderno de Provas – *A União*

2^a A União – 2019/2

UDESC – IFMS – UTFPR

Servidor BOCA (Arena Joinville):

<http://200.19.107.69/boca/>



Organização e Realização:

Karina Girardi Roggia (coordenação geral), Peter Laureano Brendel e Vinícius Gasparini (coordenação técnica), Lucas Hermann Negri, Cláudio Sá, Marcos Creuz, Gabriela Creuz, Felipe Weiss, Adilson Jonck (Makers).

Patrocinador 2019: Neogrid

Lembretes:

- Informamos que algumas questões desta prova são de nossa autoria. Contudo, outras foram selecionadas e adaptadas de competições passadas das seguintes universidades: Universidade de Waterloo, Universidade de Stanford.
No fim da prova, um link para as questões originais será fornecido e, ficará salvo nos arquivos da Maratona de Programação Udesc.
- Aos *javanheiros*: **o nome da classe deve ser o mesmo nome do arquivo a ser submetido**.
Ex: **classe** `petrus`, nome do arquivo `petrus.java`;
- Exemplo de leitura de entradas que funcionam:

```
Java: (import java.util.Scanner)
Scanner in = new Scanner(System.in);
ou
Scanner stdin = new Scanner(new BufferedReader(new InputStreamReader(System.in)));
```

```
C: (#include <stdio.h>)
int integer1; scanf("%d", &integer1);
```

```
C++: (#include <iostream>)
int integer1; std::cin >> integer1;
```

Exemplo de saída de entradas:

```
Java: System.out.format("%d %d\n", integer1, integer2);
C: printf("%d %d\n", integer1, integer2);
C++: std::cout << integer1 << " " << integer2 << std::endl;
```

- É permitido consultar livros, anotações ou qualquer outro material impresso durante a prova;
- A correção é automatizada, portanto, **siga atentamente as exigências da tarefa quanto ao formato da entrada e saída conforme as amostras dos exemplos**. Deve-se considerar entradas e saídas padrão;
- Todos os compiladores (Java, Python, C e C++) são padrões da distribuição Ubuntu versão 16.04 (gcc C11 habilitado);
- Procure resolver o problema de maneira eficiente. Se o tempo superar o limite pré-definido, a solução não é aceita. As soluções são testadas com outras entradas além das apresentadas como exemplo dos problemas;
- Teste seu programa antes de submetê-lo. A cada problema detectado (erro de compilação, erro em tempo de execução, solução incorreta, formatação imprecisa, tempo excedido ...), há penalização de 20 minutos. O tempo é critério de desempate entre duas ou mais equipes com a mesma quantidade de problemas resolvidos;
- Utilize o *clarification* para dúvidas da prova. Os juízes podem **opcionalmente** atendê-lo com respostas acessíveis a todos;
- Algumas interfaces estão disponíveis nas máquinas Linux, que podem ser utilizada no lugar da *Unity*. Para isto, basta dar *logout*, e selecionar a interface desejada. Usuário e senha: *udesc*;
- Ao pessoal local: **cuidado com os pés sobre as mesas para não desligarem nenhum estabilizador/computador de outras equipes!**

Patrocinador e Agradecimentos

- Neogrid – Patrocinador oficial do ano de 2019;
- DCC/UDESC;
- Aos bolsistas deste ano pelo empenho;
- Alguns, muitos outros anônimos.

A União

2ª Seletiva Interna da UDESC IFMS UTFPR – 2019

10 de agosto de 2019

Conteúdo

| | | |
|----|-------------------------------|----|
| 1 | Problema A: Piscinas Malucas | 5 |
| 2 | Problema B: Bora Viajar | 6 |
| 3 | Problema C: Cercados | 8 |
| 4 | Problema D: Divisão | 9 |
| 5 | Problema E: Código Morse | 11 |
| 6 | Problema F: Kombi | 13 |
| 7 | Problema G: Copos | 14 |
| 8 | Problema H: Básico | 15 |
| 9 | Problema I: Vamos às Compras | 16 |
| 10 | Problema J: Feriado | 18 |
| 11 | Problema K: Pixels no Círculo | 20 |

Atenção quanto aos nomes e números dos problemas!!!

Informamos que algumas questões desta prova são de nossa autoria. Contudo, outras foram selecionadas e adaptadas de competições passadas das seguintes universidades: Universidade de Waterloo, Universidade de Stanford.

No fim da prova, um link para as questões originais será fornecido e, ficará salvo nos arquivos da Maratona de Programação Udesc.

1 Problema A: Piscinas Malucas

Arquivo: A.[c|cpp|java|py]

Tempo limite: 1 s

A professora Karina, a.k.a Kaqui, é apaixonada por matemática. Nessas férias convidou todas suas turmas para passar um dia no Parque (Aqu+Quadr)*ático. O legal desse parque aquático é que a cada mês o formato das piscinas é automaticamente alterado de forma que algumas regiões tenham hidromassagem e outras não. Para automatizar essa piscina o Parque (Aqu+Quadr)*ático conta com regiões definidas por quadrados. Sendo assim, a piscina é definida por diversos quadrados grandes o suficiente para uma pessoa ficar sobre. Em alguns meses estes quadrados estão com hidromassagem ativada e, em outros não.

Kaqui quando viu o parque ficou impressionada e logo se perguntou se era possível que ela e todos os alunos que foram viajar ficassem agrupados na piscina para aproveitar a hidromassagem e discutir se $P = NP$.

Problema

Para não dificultar a conversa, Kaqui quer que todos fiquem agrupados formando um quadrado. Assim, todos ocupam exatamente uma hidromassagem e todos estão a uma distância aceitável um dos outros. Seu trabalho é dizer, dada a configuração total da piscina e a quantidade de alunos que viajarão com Kaqui, se é possível agrupá-los conforme definido.

Entrada

A entrada inicia com três inteiros N, C e K , que representam: o tamanho da piscina, quantidade de configurações e a quantidade de alunos respectivamente. Seguem C linhas, cada uma com quatro inteiros $C_{x1}, C_{y1}, C_{x2}, C_{y2}$, que representam os cantos de uma configuração de hidromassagens. Observe que nada impede de uma configuração estar sobre outra.

Restrições

- $K \leq N \leq 10^3$.
- $C \leq 10^3$.

Saída

Você deve imprimir “Tenta mes que vem Kaqui”, se não for possível agrupá-los conforme desejado. Caso contrário, imprima “Bora pra piscina” e, na próxima linha, as coordenadas do canto inferior direito da região onde todos deverão se agrupar.

| Exemplo de Entrada | Exemplo de Saída |
|-----------------------------|-------------------------|
| 4 2 3 1 1 1 4 2 3 3 4 | Bora pra piscina 2 3 |

2 Problema B: Bora Viajar

Arquivo: B.[c|cpp|java|py]

Tempo limite: 1 s

O Brasil é um país muito grande, a extensão do nosso território é de aproximadamente 8516000 km².

Neste ano, a final brasileira da maratona de programação ocorrerá na cidade de Campina Grande, na Paraíba. Os maratonistas da UDESC estão se preparando muito para participar, a distância de Joinville e Campina Grande é de aproximadamente 3203 km. Você precisa descobrir se é possível fazer a viagem de ida e volta utilizando a van da UDESC, que é capaz de percorrer 200 km com o tanque cheio quando guiada pelo grandíssimo Jorge.

Problema

Você receberá a distância, relacionada a Joinville, de cada um dos postos sobre a via de Joinville até Campina Grande, e deverá informar se é possível fazer a viagem de ida e volta.

Entrada

A entrada contém vários casos de teste. Cada caso inicia com uma linha contendo um inteiro N indicando o número de postos de gasolina para Jorge abastecer a van, as N seguintes linhas possuem um inteiro k e indicam em que quilometragem da via o posto se encontra. O último caso ocorre quando $N = 0$.

Restrições

- $0 \leq k \leq 3203$.

Saída

Para cada entrada você deve escrever “Melhor pegar um aviao Jorge” caso não seja possível fazer a viagem e voltar ou “Bora viajar rapaziada” caso contrário.

| Exemplo de Entrada | Exemplo de Saída |
|--|--|
| 1 900 17 3200 3000 2800 2600 2400 2200 2000 1800 1600 1400 1200 1000 800 600 400 200 0 0 | Melhor pegar um aviao Jorge Bora viajar rapaziada |

3 Problema C: Cercados

Arquivo: C.[c|cpp|java|py]

Tempo limite: 1 s

O Brasil não um país cercado: temos uma saída para o oceano. Alguns países são cercados por terra, isto é, precisam passar por pelo menos um outro território para alcançar o oceano.

Problema

Seu trabalho é determinar, para cada país em um mapa, quais são cercados e o quão cercados. Consideramos o 0 para representar que um país toca o oceano (não é cercado) em pelo menos uma direção: horizontal, vertical ou diagonal. Atravessar as fronteiras de um país significa passar por uma célula adjacente de um país diferente do original.

Observe que um país não se conecta a si mesmo. Neste caso, defina o quão cercado o país é a partir do valor mínimo de cada região conectada.

Entrada

A primeira linha contém dois inteiros N e M . Para cada uma das N linhas seguintes segue uma string de tamanho M formada por caracteres maiúsculos. Cada país é representado por um único caractere, exceto a letra W, reservada para representar a água (Water) dos oceanos para determinar o quão cercado um país é.

Restrições

- $1 \leq N, M \leq 10^3$

Saída

A saída consiste de uma lista de países separados por linha. Para cada país imprima o seu caractere correspondente seguido do quão cercado ele é conforme apresentado no exemplo.

| Exemplo de Entrada | Exemplo de Saída |
|--------------------|------------------|
| 7 10 | A 1 |
| WWWWCCDEW | B 2 |
| WWWWCCEEEW | C 0 |
| WTWWCCCCW | D 1 |
| WWFFFFFFWW | E 0 |
| WWFAAAAFWW | F 0 |
| WWFABCAFFW | T 0 |
| WWFAAAAFWW | |

4 Problema D: Divisão

Arquivo: D.[c|cpp|java|py]

Tempo limite: 9 s

Jonck e seu irmão Junior, cansaram da vida de universitário e decidiram se separar e dividir seus bens igualmente. Cada um de seus N apartamentos em Joinville tem um valor estimado entre $R\$1000000,00$ e $R\$40000000,00$. Jonck vai receber alguns destes apartamentos e Junior outros, o restante será vendido e o dinheiro dividido igualmente.

Jonck e Junior são muito gananciosos e não aceitam que o outro receba propriedades com valor total maior. A soma do valor dos apartamentos de Jonck deve ser igual a soma dos apês de Junior. Se este for o caso, Jonck e Junior gostariam de receber as propriedades de maior valor possível.

Problema

Será fornecido o valor dos N apartamentos, você deverá calcular o valor dos apês que serão vendidos de forma que o restante dos apartamentos possa ser dividido conforme eles desejam.

Entrada

A entrada consiste de vários casos de teste. A primeira linha de cada caso de teste é um inteiro N , o número de apartamentos que Jonck e Junior possuem. Seguem N linhas com um número k representando o valor de cada apartamento. A entrada termina quando $N = 0$, este caso não deverá gerar saídas.

Restrições

- $N \leq 24$.
- $k \leq 3 * 10^7$

Saída

Para cada caso de teste retorne uma única linha, o valor da soma dos apartamentos que serão vendidos.

| Exemplo de Entrada | Exemplo de Saída |
|---|------------------|
| 5 6000000 30000000 3000000 11000000 3000000 0 | 41000000 |

Exemplo

Suponha conforme na entrada, que Jonck e Junior possuem 5 apartamentos valorados em $R\$6000000,00$, $R\$30000000,00$, $R\$3000000,00$, $R\$11000000,00$ e $R\$3000000,00$. Para satisfazer os dois, um deles receberá o apartamento que vale $R\$6000000,00$, e o outro os dois apês de $R\$3000000,00$. Desta forma, os que restam serão vendidos para um valor total de $R\$41000000,00$ que será dividido entre os 2.

5 Problema E: Código Morse

Arquivo: E.[c|cpp|java|py]

Tempo limite: 1 s

O Código Morse é um sistema de representação de letras e demais símbolos através de um sinal intermitente. Originalmente desenvolvido por Samuel Morse em 1835, funciona através de um dispositivo que utiliza a corrente elétrica para controlar eletroímãs para as transmissões.

Claudiosa é um amante da linguagem Picat e desenvolveu um programa capaz de receber Código Morse. O problema é que ele não conseguia entender as mensagens que estavam chegando, ao analisar as mensagens pequenas ele percebeu que uma mesma mensagem em Código Morse poderia representar diversas cadeias de caracteres.

Abaixo a formatação escolhida por Claudiosa para a tabela Morse:

```

-----
|chr| morse |chr| morse |
| a | .-     | n | -.     |
| b | -...   | o | ---    |
| c | -.-.   | p | .--.   |
| d | -..    | q | --.-   |
| e | .       | r | .-.    |
| f | ..-    | s | ...    |
| g | --.    | t | -      |
| h | ....   | u | ..-    |
| i | ..     | v | ...-   |
| j | .---   | w | .--    |
| k | -.-    | x | -.-    |
| l | .-..   | y | -.-    |
| m | --     | z | --..   |
-----

```

Problema

A cadeia de sinais são separados em sinal curto e sinal longo, uma mensagem recebida por Claudiosa é, por exemplo, como segue: “

.--.

Esta cadeia pode ter as seguintes mensagens:

- p
- an
- eg
- we
- ate
- eme
- etn

- ette

Totalizando 8 cadeias distintas.

O seu trabalho é informar a Claudiosa quantas cadeias de caracteres diferentes existem para uma dada entrada de seu programa.

Entrada

A entrada é uma cadeia de caracteres em Código Morse de tamanho N .

Restrições

- $N \leq 10^5$.

Saída

Você deverá retornar para Claudiosa a quantidade de cadeias de caracteres distintas formadas pela entrada módulo $10^9 + 7$.

| Exemplo de Entrada | Exemplo de Saída |
|--------------------|------------------|
| .--. | 8 |

6 Problema F: Kombi

Arquivo: F.[c|cpp|java|py]

Tempo limite: 1 s

Gebrial Bazzolo vive em um mundo estranho cujo nome não será divulgado. Lá, recentemente foi criada a Kombi e Bazzolo, como um bom amante do automobilismo, adquiriu a sua.

Após comprar sua maravilhosa Kombi, Bazzolo precisou colocar uma placa e, durante o processo, percebeu que as placas eram formadas apenas por letras aleatórias, o que tornaria difícil para ele a tarefa de lembrar de sua placa.

Diante de várias placas diferentes Bazzolo queria escolher aquela tivesse mais opções de anagramas distintos, porque assim ele teria mais chances de encontrar alguma palavra para lembrar de sua placa.

Problema

O seu trabalho é ajudar Gebrial a escolher sua placa perfeita.

Entrada

A primeira linha da entrada consiste de um inteiro N , a quantidade de placas para Gebrial escolher. As N linhas que seguem contém uma string s de caracteres maiúsculos representando uma possível placa para ele.

Restrições

- $N \leq 10^5$.
- $8 \leq s \leq 13$.

Saída

Você deverá retornar para Bazzolo a placa que melhor se adequa as suas especificações e quantos anagramas diferentes é possível fazer com ela, em caso de múltiplas opções escolha a menor lexicograficamente.

| Exemplo de Entrada | Exemplo de Saída |
|--|-------------------------|
| 5 ABCDEF GHIK ABCUHAESDEQWW AAAADWADWDF EEEEEEEEEEEEEE ABCDEFGHIJ | ABCUHAESDEQWW 778377600 |

7 Problema G: Copos

Arquivo: G.[c|cpp|java|py]

Tempo limite: 1 s

Convidamos todos os maratonistas presentes aqui hoje para uma confraternização no OPA após a prova! Mas calma, ainda há uma questão nessa página.

Os copos do OPA são de 300ml e 500ml, suponha que não há como encher metade do copo porque é (quase) impossível acertar o ponto exato. Será que é possível encher um copo com exatamente 200ml? A resposta é sim: mas não contarei como pra não entregar a questão.

Problema

Considere agora um caso genérico: serão dados dois valores aleatórios para o volume dos copos, o seu trabalho é descobrir se é possível conseguir um volume específico de sua bebida preferida utilizando estes dois copos.

Seguem algumas regras para tirar dúvidas:

- Você pode encher qualquer um dos copos na fonte infinita do Opa quantas vezes quiser.
- Transferir o conteúdo do copo maior para o menor: o menor ficará cheio e o restante vai sobrar no maior, de maneira intuitiva.

Entrada

A primeira linha da entrada contém um inteiro C . Cada uma das próximas C linhas consiste de três inteiros: $a\ b\ d$, onde a, b representam o volume dos dois copos e, d é o volume desejado.

Restrições

- $1 \leq C \leq 10^5$.
- $1 \leq a, b \leq 10^7$.
- $d \leq \max(a, b)$.

Saída

Para cada um dos C casos você deve imprimir Sim, se for possível obter d através de a e b , e Nao caso contrário.

| Exemplo de Entrada | Exemplo de Saída |
|--------------------|------------------|
| 3 | Sim |
| 8 1 5 | Nao |
| 4 4 3 | Sim |
| 5 3 4 | |

8 Problema H: Básico

Arquivo: H.[c|cpp|java|py]

Tempo limite: 1 s

Fomos invadidos por alienígenas que aparentemente vieram em paz. Precisamos nos comunicar com os visitantes para garantir nossa segurança e uma grande equipe foi alocada para buscar comunicação. A sua parte do trabalho é tentar ensinar e aprender o sistema numérico dos ETs. Você percebe uma relação entre a quantidade de dedos nas mãos e a base numérica utilizada. Os aliens em questão possuem 16 dedos no total e utilizam a base 16 para realizar cálculos.

Problema

Como você pensa no futuro decidiu adiantar possíveis problemas. O seu trabalho é transformar números entre bases quaisquer.

Entrada

A primeira linha da entrada contém um único inteiro, o número de casos de teste. Cada caso de teste consiste de três inteiros X, Y, Z , a base origem, base destino e o valor a ser convertido, respectivamente. X e Y estarão escritos na base 10, o número Z estará escrito na base X , considere o seguinte mapeamento de dígitos:

[A=10, B=11, C=12, ..., Z=35]

Restrições

- $2 \leq X, Y \leq 36$.
- $Z \leq 4 * 10^9$.

Saída

Para cada linha da entrada imprima uma única linha, o valor de Z escrito na base Y .

| Exemplo de Entrada | Exemplo de Saída |
|--------------------|------------------|
| 1 16 10 11 | 17 |

9 Problema I: Vamos às Compras

Arquivo: I.[c|cpp|java|py]

Tempo limite: 2 s

Você recentemente se mudou para um novo apartamento e tem uma longa lista de coisas para comprar. Infelizmente, para comprar esse monte de coisas você precisará ir para várias lojas. Você não tem muito tempo livre, então gostaria de minimizar o tempo necessário dirigindo para comprar todos seus produtos.

Problema

A cidade que você vive é organizada como um conjunto de intersecções conectadas por ruas. A sua casa e todas as lojas estão localizadas em alguma destas intersecções. Seu trabalho é encontrar a menor rota que inicia na sua casa, visita todas as lojas necessárias e retorne para sua casa.

Entrada

A primeira linha da entrada contém um único inteiro N . O número de casos de teste. Cada uma dos N casos de teste iniciam com dois inteiros I, R , o número de intersecções e de ruas respectivamente. A sua casa está na intersecção 0. R linhas seguem, cada uma com três inteiros: X, Y, D que indica que as intersecções X e Y estão conectadas por uma rua bidirecional de comprimento D . A próxima linha contém um inteiro S , a quantidade de lojas que você precisa visitar. As S linhas seguintes contém um inteiro indicando em qual intersecção cada uma das lojas está localizada. É garantido acesso a todas as lojas partindo de sua casa.

Restrições

- $1 \leq I, R \leq 10^5$.
- $1 \leq S \leq 10$.

Saída

Para cada caso de teste imprima um único inteiro, o tamanho do menor caminho possível para sair de sua casa, visitar todas as lojas necessárias e voltar.

| Exemplo de Entrada | Exemplo de Saída |
|--|------------------|
| 1 4 6 0 1 1 1 2 1 2 3 1 3 0 1 0 2 5 1 3 5 3 1 2 3 | 4 |

10 Problema J: Feriado

Arquivo: J.[c|cpp|java|py]

Tempo limite: 1 s

Um grande empresário deseja expandir seus negócios para além de seu país natal. Porém, seus consultores o alertaram sobre os diversos problemas que poderiam surgir no decorrer desta empreitada, sendo o mais preocupante deles, é claro, os feriados estrangeiros.

Para minimizar os problemas com feriados estrangeiros, os consultores chegaram a uma brilhante conclusão: para cada país de interesse seria necessário calcular o número máximo de feriados que poderiam cair em “dias de semana” (de segunda a sexta) ao mesmo tempo, e então com base nestes números, ver qual país seria o melhor para se investir.

Problema

Entretanto, nenhum destes consultores estava a fim de fazer estes cálculos e contrataram você para isto. Sendo assim, sua tarefa é fazer um programa que recebe as datas de todos os feriados de um determinado país e imprime a maior quantidade possível de feriados que podem cair em “dias de semana” ao mesmo tempo, desconsiderando anos bissextos.

Entrada

A entrada é constituída de diversos casos de teste. Cada caso teste se inicia com um inteiro N , que indica o número de feriados do país em questão. Cada uma das próximas N linhas contém um inteiro F que representa o dia do ano daquele feriado (e.g., $F=3$ seria “03 de Janeiro”, $F=32$ seria “01 de Fevereiro”, $F=256$ seria “13 de Setembro”), sendo que todas as N linhas possuem valores distintos. O último caso de teste é seguido por uma linha com o valor 0.

Restrições

- $1 \leq N \leq 365$
- $1 \leq F \leq 365$

Saída

Para cada caso teste, imprima o número máximo de feriados que podem cair em “dias de semana” ao mesmo tempo.

| Exemplo de Entrada | Exemplo de Saída |
|--|------------------|
| 6 53 54 55 56 57 58 2 205 211 4 1 17 40 112 0 | 5 2 3 |

11 Problema K: Pixels no Círculo

Arquivo: K.[c|cpp|java|py]

Tempo limite: 2 s

O prof. Claudiusvirus Linus quase lecionou a disciplina de PIM (Processamento de Imagens) ... mas ficou encantado pelo número de problemas de nosso cotidiano e a teoria desta disciplina. O primeiro problema era o conceito de *resolução de telas* ou vídeo. A resolução indica quantos pixels iluminam uma determinada região da tela. Por exemplo, numa TV de alta-definição de 1080p quase 1 milhão de pixels são iluminados. Esta é a sua tarefa: dado um tipo de *região*, saber quantos pixels são iluminados ou acesos.

Assuma que nosso visor é uma grade cartesiana onde cada pixel se encontra em um quadrado como uma unidade perfeita. Isto é, um pixel ocupa um quadrado com as diagonais opostas em $(0,0)$ e $(1,1)$. Se este problema fosse contar quadrados, isto seria trivial. Contudo, vamos contar os pixels, mas em uma *região* circular. Um círculo é desenhado pela especificação de seu centro e um raio.

Assim, em nosso visor ou *display*, um pixel é iluminado se qualquer parte estiver coberta pelo círculo desenhado. Caso algum ponto como arco (dois vértices da *grid*) ou vértices forem tocados, estes não serão iluminados. Veja ilustração da figura 1.

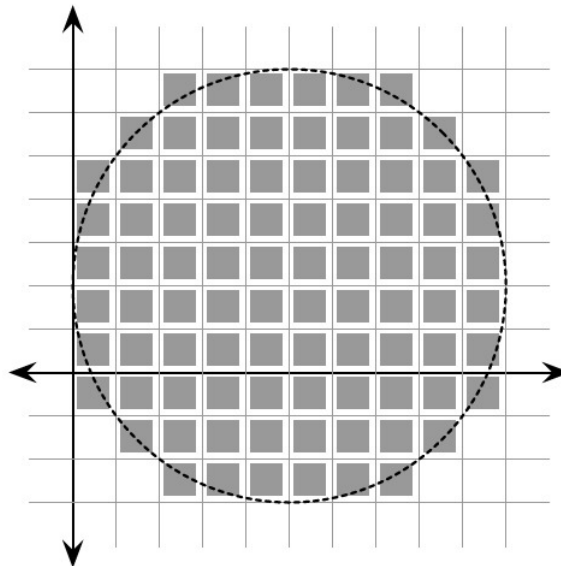


Figura 1: Os quadrados escuros estão iluminados

Problema

Seu trabalho nesta disciplina é calcular precisamente quantos pixels serão iluminados quando um círculo for descrito a partir de posição (x,y) e um raio de comprimento r é desenhado.

Entrada

A entrada consiste de vários casos de teste, os quais são um por linha. Cada caso de teste consiste de 3 valores inteiros: x , y e r , que especificam o centro do círculo, (x,y) , e o raio r . Os casos se encerram com uma linha composta por 0 0 0, a qual não é processada. Veja os exemplos que se seguem.

Restrições

- $1 \leq x, y, r \leq 10^6$.

Saída

Para cada caso de teste, apresente o número de pixels iluminados, definidos pelo círculo desenhado. Assuma que o círculo inteiro se ajusta dentro da área do *display*

| Exemplo de Entrada | Exemplo de Saída |
|--|-------------------|
| 1 1 1 5 2 5 2194 1497 283 0 0 0 | 4 88 252676 |