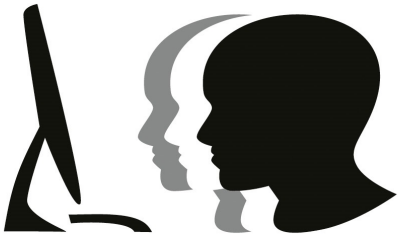

MARATONA DE PROGRAMAÇÃO

InterFatecs

Taquaritinga - 2017

Final - 26/08/2017

Caderno de problemas

Patrocínio:

Realização:

www.interfatecs.com.br

1 Instruções

Este caderno contém 12 problemas – identificados por letras de A até L, com páginas numeradas de 3 até 25. Verifique se seu caderno está completo.

Informações gerais

1. Sobre a competição
   (a) A competição possui duração de 5 horas (início as 13:00h término as 18:00h);
   (b) NÃO é permitido acesso a conteúdo da Internet ou qualquer outro meio eletrônico ou digital;
   (c) É permitido somente acesso a conteúdo impresso (Cadernos, Apostilas, Livros);
   (d) É vedada a comunicação entre as equipes durante a competição, bem como a troca de material de consulta entre elas;
   (e) Cada equipe terá acesso a 1 computador dotado do ambiente de submissão de programas (BOCA), dos compiladores, link-editores e IDEs requeridos pelas linguagens de programação permitidas;
   (f) NÃO é permitido o uso de notebooks ou outro tipo de computador ou assistente pessoal;
   (g) Os problemas têm o mesmo valor na correção.
2. Sobre o arquivo de solução e submissão:
   (a) O arquivo de solução (o programa fonte) deve ter o mesmo nome que o especificado no enunciado (logo após o título do problema);
   (b) Confirme se você escolheu a linguagem correta e está com o nome de arquivo correto antes de submeter a sua solução.
3. Sobre a entrada
   (a) A entrada de seu programa deve ser lida da entrada padrão (não use interface gráfica);
   (b) Seu programa será testado em vários casos de teste válidos além daqueles apresentados nos exemplos. Considere que seu programa será executado uma vez para cada caso de teste.
4. Sobre a saída
   (a) A saída do seu programa deve ser escrita na saída padrão;
   (b) Não exiba qualquer outra mensagem além do especificado no enunciado.

MARATONA DE PROGRAMAÇÃO
InterFatecs

Problema A

Frequência

Arquivo fonte: frequencia.{ c | cpp | java }
Autor: Erico Veriscimo (ETEC Guaianazes)

Iago estuda em uma instituição muito rígida e está faltando muito por motivos pessoais. Ele está na dúvida se conseguirá frequência suficiente para ser aprovado e, caso consiga, ainda gostaria de saber quantos dias pode faltar e ainda assim ter frequência suficiente para ser promovido. Na instituição em que estuda, a frequência mínima para não ser retido por falta é de 75% de presença e, como você é um grande amigo dele, irá desenvolver um programa capaz de tirar sua dúvida e ainda informar quantos dias ele pode faltar sem ser retido por ausências.

Entrada

Cada caso de teste contém três números inteiros D, F e R, ($30 \leq D \geq F + R \geq 0$), que especificam, respectivamente, o número de dias letivos do curso, o número de dias em que Iago frequentou a unidade e o número de dias restantes para acabar o curso.

Saída

Caso Iago vá todos os dias que ainda faltam para acabar o curso e mesmo assim não consiga atingir o mínimo de frequência (75%), imprima o texto "sem chance". Caso Iago vá todos os dias que ainda faltam e sua frequência atingir 75% a 77%, imprima "soh um milagre" e, caso sua frequência seja maior do que 77%, imprima "tranquilo" e a quantidade de dias que ele pode faltar e ainda assim não ser retino por falta. Finalize com uma quebra de linha.

Exemplo de Entrada 1	Exemplo de Saída 1
`100 76 0`	`soh um milagre`

Exemplo de Entrada 2	Exemplo de Saída 2
`70 10 10`	`sem chance`

Exemplo de Entrada 3	Exemplo de Saída 3
`100 10 90`	`tranquilo pode faltar 25 dia(s)`

Esta página foi propositadamente deixada em branco.

MARATONA DE PROGRAMAÇÃO
InterFatecs

Problema B

Influência

Arquivo fonte: influencia.{ c | cpp | java }
Autor: Lucio Nunes de Lira (Fatec São Paulo)

Luiz é aluno do curso de Metalurgia e planeja ser candidato à presidência de sua sala. Como não é inocente, sabe que precisa mensurar sua influência perante seus companheiros para saber se tem chances de vencer. Para tanto, Luiz usará o Posicionamento Tático^{TM} (PT), uma técnica baseada no reconhecimento do próprio nível de influência e do local onde se senta na sala de aula (vide Figura B.1).

O PT funcionará assim: (I) Luiz sentará em uma cadeira de alguma fileira da sala; (II) a partir desse local, tentará influenciar todos os companheiros que estão $adjacentes$ a ele, na vertical, horizontal e diagonais; (III) será influenciado todo aquele com um nível de influência menor do que o dele; (IV) todo influenciado tentará fazer o mesmo com seus adjacentes, considerando seu próprio nível de influência; (V) o procedimento terminará quando não for possível influenciar mais ninguém.

Figura B.1: Uma possível sala com fileiras, cadeiras e alunos com seus respectivos níveis de influência.

Ao final, o futuro candidato quer saber quantos foram influenciados, incluindo ele, e se essa quantidade corresponde à maioria da turma.

Entrada

A primeira linha da entrada contém dois inteiros F e C $(1 \leq F, C \leq 100)$ indicando a quantidade de fileiras e cadeiras por fileira da sala, respectivamente; a segunda linha têm dois inteiros FL e CL $(1 \leq FL \leq F)$ e $(1 \leq CL \leq C)$ indicando, respectivamente, a fileira e a cadeira onde Luiz está; as próximas F linhas terão C inteiros I $(1 \leq I \leq 1000)$, separados por espaços, que representam os níveis de influência dos alunos.

Saída

Imprima a quantidade de influenciados, incluindo Luiz, e também a palavra "$maioria$" (sem aspas e minúsculo), se essa quantidade representa a maioria dos alunos. Finalize com uma quebra de linha.

Exemplo de Entrada 1

```
5 6
3 4
1 4 3 4 7 3
4 2 3 4 3 7
4 4 2 5 6 4
5 5 4 3 2 4
1 5 4 4 1 9
```

Exemplo de Saída 1

```
12
```

Exemplo de Entrada 2

```
5 6
2 6
1 4 3 4 7 3
4 2 3 4 3 7
4 4 2 5 6 4
5 5 4 3 2 4
1 5 4 4 1 9
```

Exemplo de Saída 2

```
17 maioria
```

Exemplo de Entrada 3

```
4 10
2 5
7 7 7 7 7 7 7 7 7 7
7 7 7 7 6 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7
```

Exemplo de Saída 3

```
1
```

Exemplo de Entrada 4

```
3 10
2 5
6 6 6 6 6 6 6 6 6 6
6 6 6 6 7 6 6 6 6 6
6 6 6 6 6 6 6 6 6 6
```

Exemplo de Saída 4

```
9
```

MARATONA DE PROGRAMAÇÃO
InterFatecs

Problema C

Camadas Alfabéticas

Arquivo fonte: camadas.{ c | cpp | java }
Autor: Lucio Nunes de Lira (Fatec São Paulo)

Maria adora desafios! Recentemente elaborou um que consiste em desenhar um asterisco em um papel, escolher um número natural N e desenhar N camadas de letras em volta do asterisco (vide Figura C.1). Como todo bom desafio, existem regras: (I) as camadas contornam o asterisco como um quadrado; (II) cada camada é composta pelo mesmo caractere; (III) o máximo de camadas é 52, sendo as 26 mais externas compostas por letras maiúsculas e as 26 mais internas por minúsculas, ambas em ordem alfabética; (IV) entre cada caractere da mesma camada existe um espaço em branco. Seu objetivo é criar um programa que, dado N, faça o desenho das adoráveis "camadas alfabéticas".

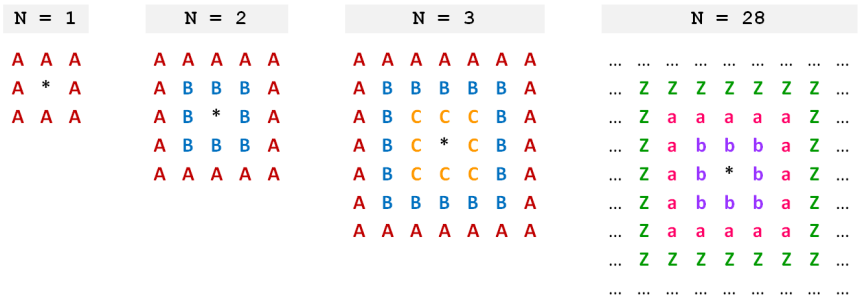

Figura C.1: Exemplos de desenhos feitos por Maria.

Entrada

A entrada contém um número natural N $(1 \leq N \leq 52)$ representando o número de camadas.

Saída

Deverá ser impresso o desenho, conforme o proposto por Maria. Finalize com uma quebra de linha.

Exemplo de Entrada 1	Exemplo de Saída 1
1	A A A A * A A A A

Exemplo de Entrada 2	Exemplo de Saída 2
2	A A A A A A B B B A A B * B A A B B B A A A A A A

Esta página foi propositadamente deixada em branco.

MARATONA DE PROGRAMAÇÃO
InterFatecs

Problem D

Superstition

Source file: superstition.{ c | cpp | java }
Author: Anderson V. de Araujo/Lucas Tsutsui (UFMS)

Vinicius is an Information Systems student at FACOM/UFMS and will take an exam in "Algorithms and Programming I"for the first time next week. The professor warned everyone in advance that the exam will be composed of N questions and each question will have two answer alternatives, True or False. As Vinicius is very superstitious, he doesn't answer two consecutive questions with the alternative 'True'. He believes that there is a low probability that the professor would stick with 'True' as the correct answer for two consecutive questions. Therefore, the technique developed by him to answer the N questions can be summarized as follows: for all i $(1 \leq i \leq N)$, if Vinicius answers the question i with True, he will certainly answer the question $i + 1$, if any, with False. This way, he expects to answer a larger number of questions and get a good grade in the discipline.

While he was preparing himself for the exam, Vinicius was very curious to know how many different ways he could answer the exam using the elaborated method. To do so, he wants to create a program that will help him determine the number of possible ways to answer the N exam questions following his technique. For example, if N=3, there are 5 possible ways to answer the exam questions: FFF, FFV, FVF, VFF and VFV (note that FVV, VVF, and VVV are not valid answers because they do not follow the method developed by him). Can you help him with this task?

Input

The input is composed of a line with an integer N $(1 \leq N \leq 10^5)$, representing the number of questions in the exam.

Output

As the number of possible ways to answer the N exam questions can be great, print only the reminder of the division of this number by $10^9 + 7$. Place a line break after the value.

Example of Input 1	Example of Output 1
3	5

Example of Input 2	Example of Output 2
10	144

Example of Input 3	Example of Output 3
514	168686355

Example of Input 4	Example of Output 4
2791	344675565

Esta página foi propositadamente deixada em branco.

MARATONA DE PROGRAMAÇÃO
InterFatecs

Problema E

Posição de Equilíbrio

Arquivo fonte: equilibrio.{ c | cpp | java }
Autor: Julio Lieira (Fatec Lins)

Neste problema você precisa achar qual posição de uma dada string é o seu ponto de equilíbrio. Para tanto, as regras são:

- A string pode conter somente letras, espaço em branco e dígitos (0,1,2,...,9);
- Cada letra possui um peso, segundo a tabela 1 abaixo;

Tabela 1: Peso de cada letra

1	2	3	4	5	6	7	8	9	10	11	12	13	13	12	11	10	9	8	7	6	5	4	3	2	1
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

- Letra maiúscula tem o mesmo peso de sua correspondente minúscula;
- O peso do caractere espaço em branco é 1;
- O peso dos dígitos é o seu próprio valor, exceto o 0 que vale 1, ou seja, o 0 vale 1, o 1 vale 1, o 2 vale 2, o 3 vale 3, o 4 vale 4, o 5 vale 5, o 6 vale 6, o 7 vale 7, o 8 vale 8 e o 9 vale 9;
- A posição de equilíbrio será a posição que divide a string em duas partes com a menor diferença, em módulo, entre a soma dos pesos de seus caracteres. Pode ser a posição exata de um determinado caractere ou a posição entre dois caracteres. Em caso da existência de mais de uma posição com a menor diferença, prevalecerá a que ocorrer primeiro, da esquerda para a direita.

No exemplo da Figura 1 o ponto de equilíbrio da string "String para teste1" é a posição entre os caracteres 'espaço em branco' e 'p', ou seja, entre os caracteres nas posições 7 e 8 (considere que o primeiro caractere da string está na posição 1). Note que a soma dos pesos dos caracteres do lado esquerdo da posição de equilíbrio é 54 e a do lado direito é 56, com diferença igual a 2.

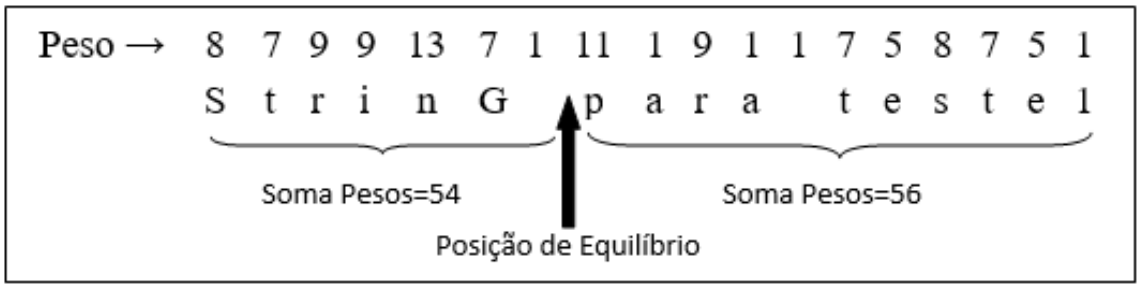

Figura E.1: Exemplo 1

Já no exemplo da Figura 2 o ponto de equilíbrio passa a ser a posição 8, exatamente onde se encontra o caractere 'p'. Note que neste caso, quando a posição de equilíbrio é a posição exata de um caractere, o peso do referido caractere é dividido entre os dois lados para a soma dos pesos.

Entrada

A entrada consiste de uma string de no mínimo 2 e no máximo 100 caracteres.

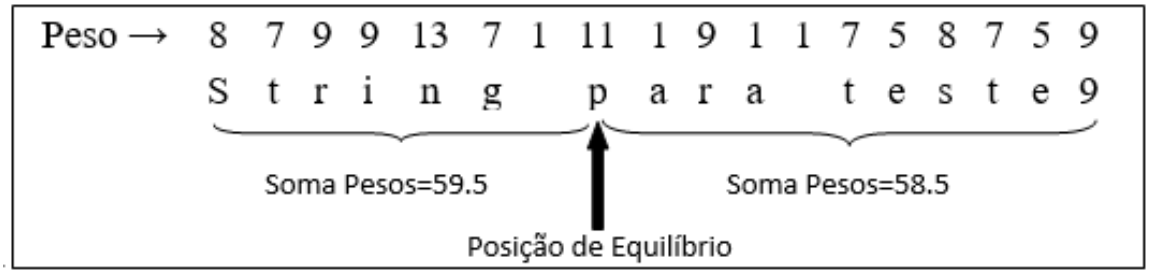

Figura E.2: Exemplo 2

Saída

Caso o ponto de equilíbrio seja em uma posição exata de um caractere da string, imprima na saída o número correspondente a posição do ponto de equilíbrio. Caso o ponto de equilíbrio esteja entre dois caracteres, imprima na saída as posições correspondentes separadas por uma vírgula. Finalize a saída com uma quebra de linha.

Exemplo de Entrada 1	**Exemplo de Saída 1**
`String para teste1`	`7,8`

Exemplo de Entrada 2	**Exemplo de Saída 2**
`String para teste9`	`8`

Exemplo de Entrada 3	**Exemplo de Saída 3**
`Final InterFatecs 2017`	`8,9`

Exemplo de Entrada 4	**Exemplo de Saída 4**
`O BBBicho Vai Pegarrr`	`12,13`

MARATONA DE PROGRAMAÇÃO
InterFatecs

Problema F

Encurtador de Url

Arquivo fonte: url.{ c | cpp | java }
Autor: Julio Lieira (Fatec Lins)

A cada dia é mais comum o problema de manipular urls enormes. O problema se agrava quando temos que enviar este link em um serviço de mensagens instantâneas por celular. Assim, os serviços de encurtar URL tem ganhado visibilidade a ponto de despertar a gigante Google que, além de outras empresas como Bit.ly, TinyURL e Migre.me, também oferece esse serviço em https://goo.gl. Por exemplo, o link http://www.interfatecs.com.br/interfatecs/download/apresentacao.pdf, pode ser encurtado para https://goo.gl/1BcS56 usando o serviço da Google.

Uma abordagem muito utilizada para implementar um serviço encurtador de url é armazenar a url longa em uma base de dados. Para isso, a Tabela poderia conter três colunas: uma contendo um identificador inteiro auto incrementado, outra para a string longa (url a ser encurtada) e outra com uma string curta correspondente. O problema, então, se resume em associar o identificador à string curta e vice-versa, e de uma maneira única.

A empresa XYZ em que você trabalha está necessitando de um serviço encurtador de URL e você foi encarregado de implementá-lo. No momento, você está implementando as rotinas de codificação e decodificação do identificador usado na Tabela da base de dados que armazenará as URLs encurtadas.

A codificação consiste em mapear o identificador inteiro em uma string curta de até 6 caracteres. Considerando como caracteres válidos as letras do alfabeto, minúsculas [a..z] e maiúsculas [A..Z] e os dígitos [0..9], temos um total de 62 caracteres, possibilitando 56.800.235.584 (62^6) strings de 6 caracteres. Associando o valor 0 à letra a, o valor 1 a letra b, ..., o valor 25 a letra z, o valor 26 a letra A, o valor 27 a letra B, ..., o valor 51 a letra Z, o valor 52 ao dígito 0, o valor 53 ao dígito 1, ..., o valor 61 ao dígito 9, temos um sistema numérico de base 62. Agora o problema de codificação se resume em converter o identificador inteiro da base 10 para nosso sistema de base 62. Por exemplo, suponha que o valor do identificador inteiro seja 125 na base 10, convertendo para nossa base:

$125_{(10)} = 2x62^1 + 1x62^0 = [2,1]$

Mapeando os índices 2 e 1 para os símbolos de nosso sistema de base 62, temos os símbolos $cb_{(62)}$. Portanto, cb seria a string curta correspondente ao identificador 125, e que seria retornada para compor a URL encurtada: http://xyz.br/cb

O processo de decodificação faz o contrário, ou seja, dada a string curta representando um número em nosso sistema de base 62, deve convertê-la para um número na base 10, o qual é o identificador para recuperar o registro da base de dados que contém a string longa. Por exemplo, considere a url curta http://xyz.br/dZYjX Utilizando-se a posição de cada símbolo em nosso sistema de base 62, fazemos a conversão:

$$\begin{array}{cccccccccccccc} & & \mathbf{d} & & \mathbf{Z} & & \mathbf{Y} & & \mathbf{j} & & \mathbf{X} & & \\ \text{dZYjX}_{(62)} & = & \mathbf{3}x62^4 & + & \mathbf{51}x62^3 & + & \mathbf{50}x62^2 & + & \mathbf{9}x62^1 & + & \mathbf{49}x62^0 & = & 56676543_{(10)} \end{array}$$

Portanto, o valor 56676543 é o identificador do registro que guarda a url longa associada, bastando fazer uma busca na base pelo registro com esse identificador (WHERE id = 56676543).

Entrada

A entrada é composta por um caractere 'C' ou 'D', indicando a operação a ser realizada (C para Codificação e D para Decodificação), seguido por um espaço e depois por um inteiro I ($1 \leq I \leq 56800235583$) ou por uma string curta de até 6 caracteres, os mesmos usados como símbolos do sistema de base 62 definido acima.

Saída

Caso a operação solicitada seja uma Codificação (C), imprima na saída a string curta correspondente. Caso seja uma operação de Decodificação, imprima o número inteiro correspondente. Finalize com uma quebra de linha.

Exemplo de Entrada 1	Exemplo de Saída 1
C 125	cb

Exemplo de Entrada 2	Exemplo de Saída 2
D dZYjX	56676543

Exemplo de Entrada 3	Exemplo de Saída 3
C 56800235583	999999

Exemplo de Entrada 4	Exemplo de Saída 4
D 999999	56800235583

Exemplo de Entrada 5	Exemplo de Saída 5
C 34108806112	LoveMe

MARATONA DE PROGRAMAÇÃO
InterFatecs

Problema G

Teorema de Zeckendorf

Arquivo fonte: teorema.{ c | cpp | java }
Autor: Antonio Cesar de Barros Munari (Fatec Sorocaba)

O matemático belga Edouard Zeckendorf publicou, em 1972, um trabalho contendo um teorema afirmando que qualquer número natural pode ser expresso como sendo a soma de termos distintos e não consecutivos da famosa sequência de Fibonacci. Essa sequência de termos costuma ser chamada de Sequência de Zeckendorf. Segundo o teorema, que passou a ser conhecido como Teorema de Zeckendorf, garante-se duas coisas:

1. Existência: todo número natural possui uma sequência de Zeckendorf, e
2. Unicidade: cada número natural N possui apenas uma sequência de Zeckendorf.

Sabe-se que cerca de 20 anos antes da publicação do trabalho de Zeckendorf, o matemático dinamarquês Gerrit Lekkerkerker já havia percebido essa propriedade dos números de Fibonacci. Mas apesar disso o nome do teorema popularizou-se com o sobrenome do belga, talvez por que Lekkerkerker seja mais complicado, quem sabe? Sua tarefa neste problema é, para um número natural informado, determinar sua correspondente sequência de Zeckendorf.

Entrada

A entrada é composta por uma linha contendo um inteiro N ($0 \leq N \leq 2^{31} - 1$).

Saída

O programa deverá imprimir entre parênteses a sequência de Zeckendorf, com os termos em ordem decrescente de seus valores, separados por um sinal de adição e sem qualquer espaço adicional. Finalizar a saída com uma quebra de linha.

Exemplo de Entrada 1	**Exemplo de Saída 1**
100	(89+8+3)

Exemplo de Entrada 2	**Exemplo de Saída 2**
71	(55+13+3)

Exemplo de Entrada 3	**Exemplo de Saída 3**
0	(0)

Exemplo de Entrada 4	**Exemplo de Saída 4**
5	(5)

Exemplo de Entrada 5	**Exemplo de Saída 5**
1111	(987+89+34+1)

Exemplo de Entrada 6	**Exemplo de Saída 6**
1000000	(832040+121393+46368+144+55)

Esta página foi propositadamente deixada em branco.

MARATONA DE PROGRAMAÇÃO
InterFatecs

Problema H

Hash Function

Arquivo fonte: hash.{ c | cpp | java }
Autor: Antonio Cesar de Barros Munari (Fatec Sorocaba)

Uma função de dispersão, geralmente chamada de *hash function*, recebe um valor chave e, a partir dele, produz um valor numérico específico. Em geral estamos interessados em mapear um domínio relativamente grande de valores para um contradomínio proporcionalmente bem menor de valores e muitas abordagens podem ser utilizadas. Uma função de dispersão que produz uma distribuição equilibrada de valores, sem privilegiar excessivamente alguns elementos do contradomínio é considerada uma boa função. Vários métodos podem ser utilizados e, neste problema, estamos interessados em um deles em particular.

É uma variação do chamado "método da dobra" e consiste em decompor o valor da chave (expresso em binário) em várias partes de igual tamanho, combiná-las por meio de uma operação de bits de tal maneira que a combinação binária resultante seja o valor final do mapeamento. Em geral as partes binárias são combinada por meio de uma operação XOR (também chamada de OR exclusivo, pois dois dígitos binários combinados com XOR só produzirão 1 se forem diferentes entre si, em caso contrário produzirão 0).

Para ilustrar o método, vamos imaginar o valor de chave 1597 e tamanho das partes igual a 4 bits. A representação em binário do valor 1597 é 11000111101 e essa cadeia binária pode ser subdividida em 3 partes de 4 bits: 0110, 0011 e 1101. Fazendo então 0110 XOR 0011 dá 0101 que, combinado por sua vez com 1101 produz 1000. Portanto a função de dispersão que recebe como parâmetros 1597 e 4 retorna o valor 8 (que é a tradução decimal de 1000). Se a função tivesse recebido 1597 e tamanho 5, teríamos também 3 partes de 5 bits: 00001, 10001 e 11101 e o valor final produzido em binário seria 01101 que, em decimal, é o valor 13. Neste problema você receberá um valor de chave e um tamanho específico e deverá aplicar o método para determinar o *hash value* correspondente.

Entrada

A entrada é composta por uma linha contendo um inteiro N em formato decimal ($0 \leq N \leq 2^{31} - 1$) que representa o valor chave e um inteiro K ($0 < K \leq 10$) informando a comprimento em bits que cada parte da representação binária de N deve possuir.

Saída

Imprimir uma linha contendo o valor (em representação decimal) produzido pelo método descrito, para os valores informados na entrada. Finalizar a saída com uma quebra de linha.

Exemplo de Entrada 1	**Exemplo de Saída 1**
1597 4	8

Exemplo de Entrada 2	**Exemplo de Saída 2**
1597 5	13

Exemplo de Entrada 3	**Exemplo de Saída 3**
2147483647 10	1022

Esta página foi propositadamente deixada em branco.

MARATONA DE PROGRAMAÇÃO
InterFatecs

Problema I

Cabeamento

Arquivo fonte: cabos.{ c | cpp | java }
Autor: Antonio Cesar de Barros Munari (Fatec Sorocaba)

Joilson trabalha para uma empresa de serviços de comunicação a cabo, sendo o responsável pela parte de elaboração de orçamentos técnicos. No momento um dos projetos em andamento é o dimensionamento de uma rede dentro de um imenso condomínio de alto padrão numa localidade próxima. A coleta dos diversos pontos que deverão integrar a rede já foi realizada, e consiste de um par de coordenadas geográficas para cada ponto, indicando a sua latitude e longitude. Para estimar o menor custo possível, que será utilizado em uma estimativa inicial, Joilson precisa saber qual a distância total (em metros) de cabo necessário para interligar todos os pontos. Nesse cálculo cada ponto estará interligado direta ou indiretamente a todos os outros, mas as ligações devem ser feitas de maneira a produzir a menor metragem possível.

Como esse cálculo é apenas um valor de referência inicial e Joilson sabe que os pontos estão localizados relativamente próximos entre si do ponto de vista geográfico, o cálculo das distâncias em metros a partir das coordenadas geográficas pode ser simplificado e desprezar a curvatura da Terra. Também pode ser assumido que todos os pontos encontram-se no mesmo hemisfério quanto à latitude e também quanto à longitude.

Resta a você, estagiário de informática da empresa, calcular a distância total solicitada por Joilson. Para realizar o cálculo simplificado da distância entre dois pontos relativamente próximos cujas latitudes e longitudes são conhecidas, é preciso calcular as distâncias latitudinal e longitudinal entre os pontos e depois aplicar uma variação da conhecida fórmula de distância entre pontos no plano cartesiano que você já aprendeu na escola. Para ilustrar o cálculo vamos supor um ponto A cuja latitude é 3º 59' 30" S e longitude 38º 31' 21" W e um ponto B com latitude 4º 01' 22" S e longitude 38º 32' 20" W.

A distância latitudinal (DLA) entre os pontos A e B é dada por:

DLA = (3 – 4) * 60 + (59 – 1) + (30 – 22) / 60

= -1 * 60 + 58 + 8 / 60

= -60 + 58 + 2 / 15

= -1.866667

Enquanto que, similarmente, a distância longitudinal (DLO) é dada por:

DLO = (38 – 38) * 60 + (31 – 32) + (21 – 20) / 60

= 0 * 60 + –1 + 1 / 60

= 0 –1 + 1 / 60

= -0.983333

A distância total em metros (DT) entre os pontos A e B é então determinada por:

DT = $\sqrt{(DLA * 1852)^2 + (DLO * 1852)^2}$ = 3907.407907 m

Entrada

A entrada é iniciada por uma linha contendo um inteiro N ($2 \leq N \leq 1000$) que indica a quantidade de pontos a serem considerados. Seguem-se então N linhas, cada uma descrevendo um ponto a ser considerado na rede, na forma YG YM YS YH XG XM XS XH ($0 \leq YG, YM, YS, XM, XS \leq 90$; $0 \leq XG \leq 180$; $YH \in \{'N', 'S'\}$; $XH \in \{'E', 'W'\}$), onde YG indica os graus de latitude, YM os minutos, YS os segundos e YH o hemisfério (norte ou sul); XG indica os graus de longitude, XM os minutos e XS os segundos, enquanto que XH representa o hemisfério (leste

ou oeste). Considere que os valores numéricos fornecidos na entrada são todos inteiros.

Saída

O programa deverá imprimir um real com duas casas após o ponto decimal, correspondente à menor distância total para a interligação entre todos os pontos da rede. Imprimir uma quebra de linha após o resultado. Utilize tipos de dados reais de precisão dupla para os cálculos.

Exemplo de Entrada 1	Exemplo de Saída 1
2 3 59 30 S 38 31 21 W 4 1 22 S 38 32 20 W	3907.41

Exemplo de Entrada 2	Exemplo de Saída 2
5 21 24 21 N 48 30 18 E 21 24 21 N 48 30 15 E 21 24 23 N 48 30 18 E 21 24 22 N 48 30 21 E 21 24 11 N 48 30 28 E	654.39

Exemplo de Entrada 3	Exemplo de Saída 3
10 23 30 7 S 47 27 28 W 23 29 45 S 47 27 30 W 23 30 1 S 47 27 1 W 23 30 10 S 47 27 48 W 23 29 15 S 47 27 55 W 23 30 15 S 47 28 30 W 23 29 52 S 47 27 35 W 23 30 32 S 48 1 1 W 23 30 50 S 48 12 6 W 23 30 5 S 48 15 3 W	91159.57

MARATONA DE PROGRAMAÇÃO
InterFatecs

Problema J

Racha Cuca?

Arquivo fonte: racha.{ c | cpp | java }
Autor: Danilo Ruy Gomes (Fatec Itapetininga)

Desde o Egito antigo, passando por guerras e sobrevivendo até nossos tempos, um dos passatempos prediletos por todo o mundo é o de encontrar palavras embaralhadas num conjunto de letras, mais conhecido como caça palavra. Para quem não conhece, este jogo consiste de várias linhas enfileiradas de maneira vertical e horizontal, onde se escondem diversas palavras sejam na vertical, horizontal ou diagonal.

Como a evolução também chegou a esse antigo desafio, você foi convocado a iniciar os testes para uma nova plataforma de jogos criando um algoritmo que dado um conjunto de letras e um conjunto de palavras deverá localizar e contar quantas palavras existem. O detalhe é que além das palavras na vertical, horizontal, diagonal principal, diagonal secundária, também poderá haver palavras inseridas ao contrário. Um exemplo é mostrado na figura abaixo:

Palavras:

MANDE
HOJE
BULE

M	**H**	**O**	**J**	**E**
Y	**A**	V	R	**L**
A	B	**N**	E	**U**
Y	V	V	**D**	**B**
V	S	W	S	**E**

Repare que há uma palavra encontrada na diagonal principal (MANDE), outra na vertical inversa (BULE) e outra na horizontal (HOJE). Com base nesta ideia, crie seu algoritmo imprimindo a quantidade de palavras encontradas. Leve em consideração que tal algoritmo não haverá casos de palavras repetidas nem palavras acentuadas.

Entrada

Na primeira linha da entrada é informando o tamanho do caça palavras que deverá ser um número ímpar N ($3 \leq N \leq 11$). Nas próximas N linhas são informadas as letras que irão compor seu caça palavras, no formato dos exemplos. Em seguida a quantidade de palavras P($3 \leq P \leq 7$) e depois as palavras que devem ser sempre menores que o tamanho N conforme a quantidade P.

Saída

Seu programa deve imprimir uma única linha contendo um número inteiro com o total de palavras encontradas. Nem todas as palavras informadas poderão estar no caça palavras. Finalize com uma quebra de linha.

Exemplo de Entrada 1	Exemplo de Saída 1
7 FACABAR AARABQT TOMABMQ EECIGAA CFYYLNA TAUHHIB WCHNQMA 5 FATEC FACA CAFE FAMILIA BOLA	4

Exemplo de Entrada 2	Exemplo de Saída 2
9 QCRWCALOB BZQUEIJOC CCBSRTTTF VWSSVTTTG BSSSEQQBH CSSSJQPQQ XSSSATPQS DSZBYUOAA XBSBUUASS 5 BOLA QUEIJO CERVEJA CHURROS BOLACHA	3

MARATONA DE PROGRAMAÇÃO
InterFatecs

Problema K

Paridade Numérica

Arquivo fonte: paridade.{ c | cpp | java }
Autor: Reinaldo Arakaki (Fatec São José dos Campos)

Carlos, excêntrico bilionário, está atrás de um algoritmo de encriptação de uma cadeia de números e chegou no algoritmo de paridade númerica. Ajude Carlos a implementar o algoritmo.

Dado um número inteiro n, de no máximo 9 algarismos, ele será sempre representado por 3 números, o primeiro número representa o número de algarismos, o segundo número a quantidade de números pares e o terceiro número a quantidade de números ímpares.

Entrada

A entrada consiste de um valor inteiro N ($1 \leq N \leq 999999999$).

Saída

Escreva na saída o número obtido, seguido de uma quebra de linha.

Exemplo de Entrada 1	Exemplo de Saída 1
987654321	945

Exemplo de Entrada 2	Exemplo de Saída 2
77	202

Exemplo de Entrada 3	Exemplo de Saída 3
89087	532

Exemplo de Entrada 4	Exemplo de Saída 4
91999	505

Esta página foi propositadamente deixada em branco.

MARATONA DE PROGRAMAÇÃO
InterFatecs

Problema L

Don Perrito

Arquivo fonte: perrito.{ c | cpp | java }
Autor: Leandro Luque (Fatec Mogi das Cruzes)

Don Perrito (DP) adora cachorros e astronomia. Recentemente comprou um casal de labradores e não hesitou em chamá-los de Vênus e Júpiter. Outra paixão de DP é a astrologia. Toda manhã, após admirar quadros de cachorros, planetas e estrelas enquanto caminha pela sua casa, ele lê o horóscopo do dia.

Na última semana, DP leu a seguinte mensagem: "A Lua se une a Mercúrio retrógrado em Virgem e recebe um ótimo aspecto de Plutão indicando mudanças positivas em sua vida. O momento é bom para iniciar novos empreendimentos. Como Vênus está se aproximando rapidamente de Júpiter, estes empreendimentos podem lhe trazer muito dinheiro."

Pronto! Para DP estava claro: ele iria empreender e seria com corrida de cachorro. Afinal, o horóscopo havia deixado claro: Vênus está se aproximando rapidamente de Júpiter.

Como DP não gostava de brincar com as mensagens do horóscopo, tratou logo de construir uma pista circular e comprar um novo *gadget* que, colocado no cachorro, informava quantos metros ele havia percorrido. Sua ideia era treinar Vênus e Júpiter para que ganhassem as corridas.

Como deseja controlar quanto seus cachorros treinam por dia, DP pediu sua ajuda para, dado o raio R (em metros) da pista circular e a quantidade de metros percorridos pelos cachorros, determinar quantas voltas completas eles deram na pista. Apenas lembrando seu amigo francês Pierre, o perímetro de um círculo é igual a 2*π*R, com π = 3,1415.

Entrada

A entrada é composta por uma única linha com um número de ponto flutuante com duas casas decimais R ($0 < R \leq 100$), especificando o raio da pista, e um número inteiro P ($0 \leq P \leq 10000$), especificando a quantidade de metros que os cachorros percorreram.

Saída

Imprima um número inteiro indicando a quantidade de voltas completas que os cachorros deram na pista. Finalize com uma quebra de linha.

Exemplo de Entrada 1	Exemplo de Saída 1
10.00 50	0

Exemplo de Entrada 2	Exemplo de Saída 2
1.05 15	2