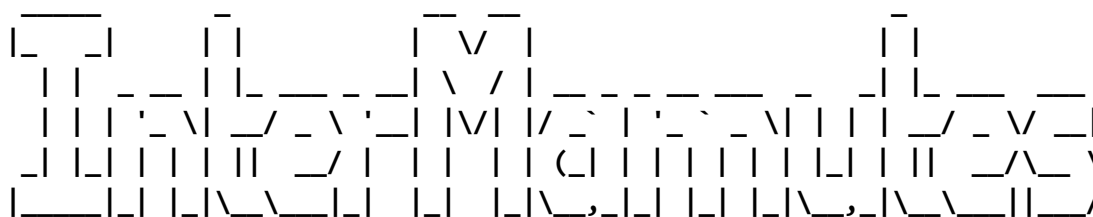


1ª



Fase única

30 de novembro de 2024

Caderno de questões

Organização e apoio

Fatec
Taquaritinga

CPS
Centro
Paula Souza

 **SÃO
PAULO**
GOVERNO DO ESTADO

Instruções

- Este caderno contém 6 problemas – identificados por letras de A até F, com páginas numeradas de 4 até 14. Verifique se seu caderno está completo.

Informações gerais

1. Sobre a competição
 - a. A competição possui duração de 2 horas (início as 09:30h término as 11:30h);
 - b. NÃO é permitido acesso a conteúdo da Internet ou qualquer outro meio eletrônico digital;
 - c. NÃO é permitido o uso de ferramentas de auxílio à codificação, como GitHub Copilot, Tabnine, Amazon CodeWhisperer ou similar;
 - d. É permitido somente acesso a conteúdo impresso em papel (cadernos, apostilas, livros);
 - e. Não é permitida a comunicação com o técnico ou qualquer outra pessoa que não seja a equipe para tirar dúvidas sobre a maratona
 - f. Cada equipe terá acesso a 1 (um) computador dotado do ambiente de submissão de programas (BOCA), dos compiladores, link-editores e IDEs requeridos pelas linguagens de programação permitidas;
 - g. NÃO é permitido o uso de notebooks, smartphones, ou outro tipo de computador ou assistente pessoal;
 - h. Todos 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;
 - c. NÃO insira acentos ou outros caracteres especiais no arquivo-fonte.
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.
5. Versões das linguagens
 - a. gcc version 11.3.0 (lembre-se que não existem bibliotecas do Windows)

- b. Python 3.10.6
- c. Java 17.0.7+7-Ubuntu-0ubuntu122.04.2 (lembre-se que as classes devem estar fora de pacotes)

Problema A

Contas a pagar

Arquivo fonte: contas.{ c | cpp | java | py }

Autor: Olimpíada Brasileira de Informática– OBI2023– Prog. Nível Sênior– Fase 1

Vô João está aposentado, tem boa saúde, mas a vida não está fácil. Todo mês é um sufoco para conseguir pagar as contas! Ainda bem que ele é muito amigo dos donos das lojas do bairro, e eles permitem que ele fique devendo.

Depois de pagar aluguel, conta de luz, conta de água, conta do telefone celular e conta do mercado, Vô João ainda tem que pagar as contas do Açougue, da Farmácia e da Padaria.

Dados o valor que Vô João tem disponível e o valor das contas do Açougue, Farmácia e Padaria, escreva um programa para determinar quantas contas, entre as três que ainda não foram pagas, Vô João consegue pagar.

Input

A entrada contém quatro linhas. A primeira linha contém um inteiro V ($0 \leq V \leq 2000$), o valor que Vô João tem disponível para pagar as contas. A segunda linha contém um inteiro A ($1 \leq A \leq 1000$), o valor da conta do Açougue. A terceira linha contém um inteiro F ($1 \leq F \leq 1000$), o valor da conta da Farmácia. A quarta linha contém um inteiro P ($1 \leq P \leq 1000$), o valor da conta da Padaria.

Output

Seu programa deve produzir uma única linha, contendo um único inteiro, o maior número de contas que Vô João consegue pagar.

Exemplo de entrada 1

Exemplo de saída 1

100 30 40 30	3
-----------------------	---

Exemplo de entrada 2**Exemplo de saída 2**

50	0
82	
99	
51	

Problema B

Resgate em Queda Livre

Arquivo fonte: `resgate.{c | cpp | java | py}`

Autor: Leandro Zatesko, UFFS (beecrowd | 1552)

Ó, meu Deus! Um grupo de pessoas está caindo em queda livre! Elas saltaram todas exatamente ao mesmo tempo de vários aviões que estavam exatamente à mesma altura. A intenção era realizar o maior e mais belo salto sincronizado da História. No entanto, o malévolo Loki, para se deleitar com a insignificância humana, sabotara os paraquedas, e agora a única esperança está numa ação conjunta do Homem-Aranha com o Homem-de-Ferro. Como ambos são muito nerds, notaram que as pessoas estavam caindo todas num mesmo plano paralelo ao solo, a despeito da resistência do ar e de outros fatores. Então, bolaram um plano infalível. Primeiro, o aracnídeo unirá todas as pessoas através de cabos de teia entre elas.

Uma vez que não haja pessoa que não esteja conectada ao grupo, o playboy poderá eletromagnetizar o grupo todo e, segurando na mão de uma apenas das pessoas do grupo, pousar todas elas em segurança.

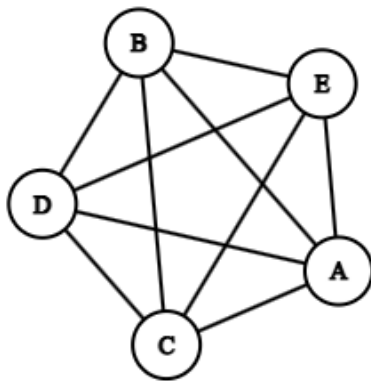


Figura B.1: Todas possíveis ligações

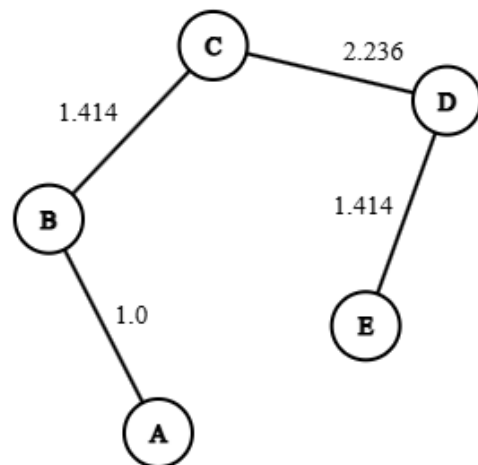


Figura B.2: Menor custo de teia

Mas não há muito tempo para divagações. O Homem-Aranha precisa agir rápido, o que no caso dele significa gastar o mínimo possível de teia. Para tanto, o Homem-de-Ferro em seu screen projetou numa malha cartesiana o plano em que as pessoas estão, usando o centímetro como unidade de medida, e obteve as coordenadas de cada pessoa na malha. Agora, J.A.R.V.I.S. está computando qual o mínimo necessário de teia de que o Homem-Aranha precisará. Dependendo da resposta, o Homem-de-Ferro não esperará pelo garoto e improvisará alguma outra peripécia.

Fórmula para calcular distância entre dois pontos: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Input

A entrada é constituída por vários casos de teste. A primeira linha de entrada contém um inteiro C que determina a quantidade de casos de teste. Cada caso de teste começa com um inteiro positivo n ($n \leq 500$), o qual representa o número de pessoas no grupo. Seguem, então, n linhas, cada uma designando uma pessoa do grupo pelas suas coordenadas x e y na malha ($0 \leq x, y \leq 10^4$).

Output

Para cada caso de teste, seu programa deverá imprimir uma linha contendo o valor com precisão de duas casas decimais correspondente ao comprimento mínimo de teia, em metros, necessário para se conectarem todas as pessoas do grupo. Atente para que o separador das casas decimais seja . (ponto), não , (vírgula).

Exemplo de entrada 1

```
2
5
0 0
0 100
100 200
200 400
300 300
4
1 5
1 4
2 3
3 2
```

Exemplo de saída 1

```
6.06
0.04
```

Problema C

Celular do Juvenal (InterFatecs 2017)

Arquivo fonte: celular.{ c | cpp | java | py }

Autor: Prof. Antonio Cesar de Barros Munari (Fatec Sorocaba)

Juvenal é um hipster. E, talvez até por ser hipster, é esquisitão. E tem um celular antigão, por que ele acha isso muito descolado. E nesta fase de sua vida hipster, Juvenal só usa o aparelho para voz e texto, por que ele é descolado e o aparelho, limitado. A figura 1 mostra como é o celular do Juvenal. No teclado, se ele está digitando texto, com um toque ele consegue a letra ‘a’ (em minúsculas), com dois toques, consegue um ‘b’ (em minúsculas), com três toques, um ‘c’ (em minúsculas), com quatro toques, um ‘A’ (em maiúsculas) e assim por diante. Nesse modo de edição de textos, para digitar um espaço em branco ele precisa pressionar uma vez a tecla do zero, como mostra a imagem.



Figura J.1: Visão parcial do celular do Juvenal.

Para um determinado texto digitado pelo hipster em seu aparelho antigão, determine a quantidade mínima de vezes que ele precisa pressionar as teclas do celular a partir do momento em que entrou em modo texto.

Input

A entrada é composta por uma cadeia não vazia composta por caracteres alfabéticos maiúsculos e minúsculos e eventuais espaços em branco. O comprimento dessa cadeia é de no máximo 200 caracteres.

Output

O programa deverá imprimir um inteiro correspondente à quantidade total de toques requeridos para a digitação da mensagem. Após o valor numérico deverá ser colocada uma quebra de linha.

Exemplo de entrada 1

batata	7
--------	---

Exemplo de saída 1

Exemplo de entrada 2

BatatA	13
--------	----

Exemplo de saída 2

Exemplo de entrada 3

Batatinha quando nasce	42
------------------------	----

Exemplo de saída 3

Exemplo de entrada 3

Centro Paula Souza	49
--------------------	----

Exemplo de saída 3

Problema D

Boleto (InterFatecs 2018)

Arquivo fonte: boleto.{ c | cpp | java | py }

Autor: Danilo Ruy Gomes (Fatec Itapetininga)

Desde o início de 2017, o Brasil vem mudando a forma de cobrança de boletos bancários. Anteriormente, qualquer pessoa podia emitir um boleto, desde que tivesse um CNPJ e uma conta em banco. Com as novas mudanças, além do CNPJ da empresa, os bancos vêm impondo que todos os boletos devem ser registrados, ou seja, com CPF, endereço e CEP válido. A NTG Soluções em Tecnologia trabalha com desenvolvimento de sistemas e vem sofrendo demais com essas novas mudanças, visto que antigamente ela utilizava uma API de terceiro para o desenvolvimento do layout de boletos e agora a própria NTG vem desenvolvendo todo o algoritmo para recebimento de boletos. Para a questão do intermédio entre as empresas e os bancos, utilizasse um arquivo de texto num padrão conhecido como CNAB, com suas variantes. Dentro deste arquivo, há 240 caracteres em cada linha (CNAB 240) onde da coluna n até $n+x$, está uma informação, como número do boleto, data de vencimento, valor, etc. A tabela seguinte mostra um exemplo do padrão CNAB:

Tamanho	Coluna Inicial	Coluna Final	Informação
6	5	10	Vencimento (DDMMYY)
6	11	16	Valor de Pagamento inteiro
2	17	18	Valor de Pagamento Decimal
4	19	22	Número Documento
6	23	28	Data de Pagamento

No momento, a empresa está desenvolvendo o processamento do arquivo de retorno, o qual contém as informações dos boletos pagos. Sua função será fazer um programa para extrair das cadeias de caracteres do arquivo texto, o somatório recebido com adimplência e inadimplência, baseando-se na data de pagamento.

Input

A entrada consiste em um único caso de teste por vez, informando-se várias linhas, cada qual contendo uma cadeia com 30 caracteres com informações sobre o pagamento de um boleto.

Output

Seu programa deverá imprimir o valor total, com duas casas decimais, seguido do termo "ADIMPLENTE" para todo os boletos pagos em dia e, na outra linha, o valor total pago de todos os boletos com atraso seguido do termo "INADIMPLENTE". Esta última linha deve terminar com uma quebra de linha.

Exemplo de entrada 1

Exemplo de saída 1

006712101800023230012317051815	904.64-ADIMPLENTE
003808111800044000065217051816	794.40-INADIMPLENTE
008907061700023150235617051817	
006329051800023234672317051812	
003220041800056290341117051814	

Exemplo de entrada 2

Exemplo de saída 2

006731101800123230012310011815	13362.30-ADIMPLENTE
003816111801213000065227051816	103126.74-INADIMPLENTE
008917061700233150235614051817	
006329051809923234672323071812	
003230041800156290341107051814	

Problema E

Galáxia 42: Missão Conversão de Dados

Arquivo fonte: conversao.{ c | cpp | java | py }

Autor: Prof. Itor Isaías da Silva (Fatec Taquaritinga)

Em uma missão espacial rumo ao futuro, a nave Galáxia 42 se encontra repleta de dados digitais coletados de inúmeras galáxias distantes. Para catalogar essas informações de forma organizada e evitar o caos digital a bordo, os cientistas precisam converter os dados brutos (medidos em bytes) para unidades mais familiares, como KB, MB, GB, e até TB. Sua tarefa é ajudar a tripulação da Galáxia 42 a transformar cada valor de bytes em um formato legível, para que possam rapidamente identificar a quantidade de dados armazenados em cada parte da nave.

Dado um ou mais valores inteiros **N** ($0 \leq N \leq 10^{12}$) representando bytes, converta cada um para uma combinação das unidades de armazenamento no sistema binário (múltiplos de 1024), nos formatos: B (bytes), KB (kilobytes), MB (megabytes), GB (gigabytes) e TB (terabytes). Para garantir a segurança de processamento na nave.

Cada unidade é convertida em múltiplos de 1024, seguindo a ordem: TB (1024^3 bytes), GB (1024^2 bytes), MB (1024^1 bytes), KB (1024^0 bytes), e B (bytes restantes).

Input

O programa recebe várias linhas, cada uma contendo um único valor inteiro **N** em bytes. A entrada é encerrada com o fim da entrada (EOL).

Output

Para cada valor de entrada **N**, imprima o valor convertido no formato TB:GB:MB:KB:B, exibindo o valor de cada unidade, mesmo que seja zero. Cada valor deve ser um número inteiro.

Exemplo de entrada 1

Exemplo de saída 1

5368709120	0:5:0:0:0
1048576	0:0:1:0:0
1234567890	0:1:177:813:146
1099511627776	1:0:0:0:0
0	0:0:0:0:0

Problema F

Pacman (InterFatecs 2015 Fase 2)

Arquivo fonte: pacman.{ c | cpp | java | py }

Autor: Leandro Luque (Fatec Mogi das Cruzes)

Pacman é um famoso jogo, criado pela Namco para fliperamas, que teve versões para diversos consoles, como Game Boy e Nintendo DS. O jogo consiste em um labirinto onde se encontra o jogador, representado por uma pizza sem uma fatia (essa foi a inspiração real para a personagem), fantasmas, que não podem ser tocados, e itens que devem ser coletados (círculo escuro), conforme Figura 1. Nesta figura, existem dois fantasmas e um item a ser coletado.

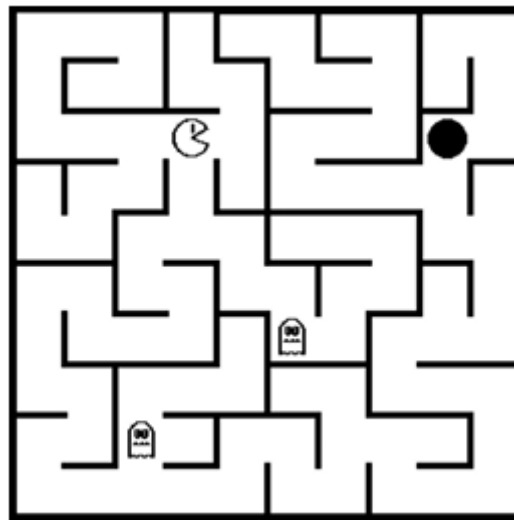


Figura 1. Exemplo de labirinto com o jogador, dois fantasmas e um item a ser coletado.

Você foi convidado para implementar uma nova versão do jogo onde os labirintos são gerados em tempo de execução (no jogo original, o labirinto era sempre o mesmo) e os fantasmas ficam parados (no jogo original, eles se moviam). Em resumo, o jogo consiste em mover o jogador pelo labirinto, sem tocar nos fantasmas, até coletar o item. Para tanto, você precisa saber se, dado um certo labirinto gerado pelo seu código, existe ao menos um caminho que leva o jogador até o item sem passar por um fantasma.

Input

A entrada contém vários casos de teste. A primeira linha de cada caso de teste contém um inteiro N ($7 \leq N \leq 20$), indicando o número N de linhas e colunas do labirinto (incluindo as paredes externas). As próximas N linhas contêm N caracteres C ($C='*'$, $C='I'$, $C='T'$, $C='#'$, $C=' '$) cada representando o labirinto e o seu conteúdo. Um caractere $C='*'$ indica uma parede do labirinto. Um caractere $C='I'$ indica o local onde se encontra o jogador. Um caractere $C='T'$ indica o local onde está o item a ser coletado. Um caractere $C='#'$ indica o local onde está um fantasma. Um espaço em branco ($C=' '$), por sua vez, representa um caminho livre no labirinto. Em um mesmo labirinto, existe apenas um jogador e um item a ser coletado, mas podem existir vários fantasmas. A entrada termina com o fim do arquivo.

Output

Para cada caso de teste, imprima um caractere 'S' (maiúsculo e sem aspas), caso exista ao menos um caminho entre o jogador e o item a ser coletado sem passar por fantasmas. Caso contrário, imprima um caractere 'N' (maiúsculo e sem aspas).

Exemplo de entrada 1

7 ***** * T*I* * *** * * # * * *** * * * * *****	S
---	---

Exemplo de saída 1