

1

Ambrosio's Bus Company

Iniciante

☆☆☆☆☆

Compartilhar

Tempo máximo de execução: 1s

Cadastrado por: Prof. Rodrigo em 11/07/13

Tópicos: estruturas

Atualizado 4 anos atrás

Fonte: Forbellone, A. L. V; Eberspacher H. F., Lógica de Programação, 3ed., 2005

| DESCRIÇÃO | ENVIAR RESPOSTA | SUBMISSÕES | ORÁCULO | ESTATÍSTICAS |
|-----------|-----------------|------------|---------|--------------|
|-----------|-----------------|------------|---------|--------------|

Descrição

TROCAR LINGUA

Ambrósio é sócio de muitas empresas. Uma das empresas de ambrósio é a Ambrósio's Bus Company que oferece o serviço de viagens de ônibus intermunicipais. A figura abaixo mostra a ficha de registro de cada passageiro. Os ônibus possuem 44 lugares numerados de 1 a 44.

Você vai receber uma lista de passageiros com no máximo 44 elementos.

Ao final imprima o nome de todos os passageiros que estiverem acima da média das idades e que estiverem sentados nas poltronas pares.

Ambrósio's Bus Company

Número da passagem:

Data:

De:

Para:

Horário

Poltrona:

Idade:

Nome do passageiro:

Formato de entrada



Você receberá uma lista de registro de passageiros, com os valores dos campos número da passagem, data, de, para, horário, poltrona, idade e nome do passageiro, conforme mostrado no exemplo abaixo:

```
12
15/05/2012
Maceió
Rio de Janeiro
9:00
12
19
Ana Maria Braga
```

A entrada termina quando o valor do número da passagem for -1.

Formato de saída

Em cada linha, imprima o nome de todos os passageiros que estiverem acima da média das idades e que estiverem sentados nas poltronas pares.

| Exemplos de: | |
|---|---|
| <div>Entrada</div> | <div>Saída</div> |
| <pre>12 15/05/2012 Maceió Rio de Janeiro 9:00 12 19 Ana Maria Braga 48 14/05/2012 Maceió São Paulo 9:00 7 30 Sílvio Santos 1 14/05/2012 Maceió Salvador 9:00 16 40 Pequeno Mestre Yoda -1</pre> | <pre>Pequeno Mestre Yoda</pre> |

1

Banco de Dados

Iniciante



Compartilhar

Tempo máximo de execução: 1s

Cadastrado por: [Caio Barbosa Vieira da Silva](#) em 10/12/13

Tópicos: array, estruturas

Atualizado 4 anos atrás

Fonte: theHuxley

| DESCRIÇÃO | ENVIAR RESPOSTA | SUBMISSÕES | ORÁCULO | ESTATÍSTICAS |
|-----------|-----------------|------------|---------|--------------|
|-----------|-----------------|------------|---------|--------------|

Descrição

TROCAR LINGUA

A empresa WOOF está entrando no mercado das redes sociais. Só que ela precisa de um banco de dados para armazenar os dados dos seus usuários. O dono da empresa é muito amigo seu, e sabe que você é um ótimo programador, logo, te chamou para implementar esse banco de dados. As informações dos usuários que você deve guardar são:

- Idade.
- Nome.
- Sexo.
- Estado civil.
- Quantidade de amigos.
- Quantidade de fotos.

Formato de entrada

Você vai receber os dados do usuário:

Um inteiro, representando a quantidade de usuários a serem adicionados ao banco de dados.

Um inteiro, contendo a idade.

Uma string, de até 50 caracteres, contendo o nome.

Um caractere, sendo ele M ou F, que é o sexo.

Uma caractere, simbolizando o estado civil:
S - Solteiro, C - Casado. N - Namorando. D - Divorciado.

Um inteiro, representando o número de amigos.

Um inteiro, representando a quantidade de fotos que estão no perfil.

Formato de saída

Imprimir as informações de todos os usuários, na sequencia em que eles foram adicionados, uma por linha.

Deve haver uma quebra de linha entre os casos de testes e ao fim do programa.

| Exemplos de: | |
|---|--|
| Entrada | Saída |
| <pre>2 20 Caio M S 150 300 15 Ana F S 1000 1500</pre> | <pre>Idade: 20 Nome: Caio Sexo: M Estado Civil: S Numero de amigos: 150 Numero de fotos: 300 Idade: 15 Nome: Ana Sexo: F Estado Civil: S Numero de amigos: 1000 Numero de fotos: 1500</pre> |

Tempo máximo de execução: 1s

Cadastrado por: [The Huxley](#) em 25/11/11

Tópicos: array, acumulador, caractere, string, dicionário, estruturas

Atualizado 4 anos atrás

Fonte: thehuxley

| DESCRIÇÃO | ENVIAR RESPOSTA | SUBMISSÕES | ORÁCULO | ESTATÍSTICAS |
|-----------|-----------------|------------|---------|--------------|
|-----------|-----------------|------------|---------|--------------|

Descrição

TROCAR LINGUA

Ambrosina é de lua. Quase todo dia ela quer mudar o cardápio de sua lanchonete. Ambrósio, o programador do sistema, não agüenta mais alterar o código do programa todas as vezes que Ambrosina muda o humor. Então Ambrósio resolveu mudar o software de forma que Ambrosina possa cadastrar o cardápio todas as manhãs, ao iniciar o software.

O software lê a quantidade de produtos a serem cadastrados, e depois o código de LED instalada na lanchonete seguidos pela descrição e preço do produto. Os clientes escolhem os produtos pelo código.

Se o cliente pede um produto não cadastrado ou uma quantidade negativa o pedido é considerado inválido.

O sistema calcula quantos itens o cliente escolheu de cada código e imprime o total da conta, sem descontos! Eita Ambrosina sovina !!

Formato de entrada

Consiste de um inteiro n, representando o número de produtos a serem cadastrados.

Depois, segue o cadastro dos n produtos, onde serão lidos para cada produto:

um inteiro representando o código de LED,

uma descrição do produto



e um número real representando o preço.

Depois, são lidos os pedidos.

O pedido consiste do código do produto e da quantidade de itens daquele produto que será pedido. O pedido se encerra quando o código lido é igual a 0.

Formato de saída

O valor que o cliente deve pagar, formatado com duas casas decimais. Pedidos inválidos são ignorados.

| Exemplos de: | |
|---|---|
| <div>Entrada</div> | <div><div>Saída</div><div></div></div> |
| <pre>3 501 Guarana Antartica 1.50 101 Coxinha 2 10 Cheese Frango 3.50 501 2 101 3 0</pre> | <pre>9.00</pre> |

2

Camisetas

Fácil

☆☆☆☆☆

Compartilhar

Tempo máximo de execução: 1s

Tópicos: multiplas entradas, estruturas

Cadastrado por: [The Huxley](#) em 03/04/13

Atualizado 4 anos atrás

Fonte: URI Online Judge

| DESCRIÇÃO | ENVIAR RESPOSTA | SUBMISSÕES | ORÁCULO | ESTATÍSTICAS |
|-----------|-----------------|------------|---------|--------------|
|-----------|-----------------|------------|---------|--------------|

Descrição

TROCAR LINGUA

O professor Rolien organizou junto às suas turmas de Ciência da Computação a confecção de uma camiseta polo que fosse ao mesmo tempo bonita e barata. Após algumas conversas, ficou decidido com os alunos que seriam feitas somente camisetas da cor preta, o que facilitaria a confecção. Os alunos poderiam escolher entre o logo do curso e os detalhes em branco ou vermelho. Assim sendo, Rolien precisa de sua ajuda para organizar as listas de quem quer a camiseta em cada uma das turmas, relacionando estas camisetas pela cor do logo do curso, tamanho (P, M ou G) e por último pelo nome.



Formato de entrada

A entrada contém vários casos de teste. Cada caso de teste inicia com um valor **N**, ($1 \leq N \leq 60$) inteiro e positivo, que indica a quantidade de camisetas a serem feitas para aquela turma. As próximas **N*2** linhas contém informações de cada uma das camisetas (serão duas linhas de informação para cada camiseta). A primeira linha irá conter o nome do estudante e a segunda linha irá conter a cor do logo da camiseta ("branco" ou "vermelho") seguido por um espaço e pelo tamanho da camiseta "P" "M" ou "G". A entrada termina quando o valor de **N** for igual a zero (0) e esta valor não deverá ser processado.

Formato de saída

Para cada caso de entrada deverão ser impressas as informações ordenadas pela cor dos detalhes em ordem ascendente, seguido pelos tamanhos em ordem descendente (as letras dos tamanhos e não os tamanhos em si, ou seja, P, M e G) e por último por ordem ascendente de nome, conforme o exemplo abaixo.

Obs.: Deverá ser impressa uma linha em branco entre dois casos de teste.

| Exemplos de: | |
|---|--|
| <div>Entrada</div> | <div>Saída</div> |
| <pre>9 Maria Jose branco P Mangojata Mancuda vermelho P Cezar Torres Mo branco P Baka Lhau vermelho P JuJu Mentina branco M Amaro Dinha vermelho P Adabi Finho branco G Severina Rigudinha branco G Carlos Chade Losna vermelho P 3 Maria Joao branco P Marcio Guess vermelho P Maria Jose branco P 0</pre> | <pre>branco P Cezar Torres Mo branco P Maria Jose branco M JuJu Mentina branco G Adabi Finho branco G Severina Rigudinha vermelho P Amaro Dinha vermelho P Baka Lhau vermelho P Carlos Chade Losna vermelho P Mangojata Mancuda branco P Maria Joao branco P Maria Jose vermelho P Marcio Guess</pre> |

3

A Mumba de Kamyamoto (Powered by Kinho)

Médio

☆☆☆☆☆

Compartilhar

Tempo máximo de execução: 2s

Cadastrado por: [Matheus Júlio Boncsidai de Oliveira](#) em 01/07/21

Tópicos: eof, array, decisão, lógica matemática, ciclos, busca, estruturas

Atualizado um dia atrás

Fonte: Kinho's help e Submissões

| DESCRIÇÃO | ENVIAR RESPOSTA | SUBMISSÕES | ORÁCULO | ESTATÍSTICAS |
|-----------|-----------------|------------|---------|--------------|
| | | | | |

Descrição

Fugiro Kamyamoto se tornou um ilustre entregador e, agora faz entregas interplanetárias. Porém em uma viagem, Kamyamoto passou perto de uma batalha estelar e perdeu o controle de seu carregamento.

Kamyamoto está numa posição central (0,0,0) no espaço (R3), e suas caixas estão em pontos aleatórios no R3,cada caixa tem um ID para identificá-la.

Sua função é desenvolver um programa que vai indicar quais caixas Kamyamoto deve pegar primeiro, e seu critério de preferência é sempre a mais próxima em relação a ele. Ele deve se deslocar até uma caixa para pegá-la e, a partir dessa nova coordenada, julgar novamente qual caixa é a mais próxima , até coletar todas. Se, por acaso, houverem duas caixas com a MESMA distância, você deve dar preferência pela caixa com menor ID.

Formato de entrada

A entrada consiste em vários casos teste.

Um inteiro N, representando o número de caixas.









x y z, três inteiros representando as coordenadas de **cada caixa** no R3.

O programa deve rodar enquanto N!=0.

A entrada termina quando o número de caixas N=0.

Formato de saída

O ID de cada caixa, separados por um espaço, e uma quebra de linha no final de cada busca.

| Exemplos de: | |
|---|---|
| <div>Entrada</div> <div></div> | <div><div>Saída</div><div></div></div> |
| <pre>3 10 10 10 5 5 5 10 0 0 0</pre> | <pre>2 1 3</pre> |
| <div>Entrada</div> <div></div> | <div><div>Saída</div><div></div></div> |
| <pre>5 24 15 8 22 4 6 -30 3 -29 -30 -13 -21 16 -7 -14 5 -8 30 26 -28 8 -5 -27 -7 -23 -4 10 7 26 -30 -2 5 -19 17 -22 30 -26 30 2 16 9 -29 -19 -27 1 27 -21 0</pre> | <pre>5 1 2 4 3 4 1 3 2 5 3 2 1 4 5</pre> |

3

Jaiminho, o Carteiro

Médio

★★★★★

Compartilhar

Tempo máximo de execução: 1s

Tópicos: array, decisão, repetição, caractere, string, estruturas

Cadastrado por: Murilo Barbosa Neto em 01/07/21

Atualizado um dia atrás

Fonte: ArialCM 12

DESCRIÇÃO

ENVIAR RESPOSTA

SUBMISSÕES

ORÁCULO

ESTATÍSTICAS

Descrição

Jaiminho, o Carteiro, programou um sistema que facilitava sua vida, recebendo dados de todas as pessoas que moravam na região para que ele facilmente encontrasse o endereço de todo mundo e evitasse mais a fadiga. Entretanto, enquanto finalizava o programa ao mesmo tempo que fazia as últimas entregas do dia (claro, empurrando sua bicicleta), seu notebook foi acertado por uma bola quadrada e então atropelado por um carro, destruindo todo o seu progresso. Sem pestanejar, ele liga para você, que o devia um favor, pedindo que faça o programa para que ele finalmente possa evitar a fadiga corretamente.

O programa consiste de duas operações principais: inserir um morador e encontrar um morador, e uma terceira que finaliza o programa.

Cada rua da cidade (5 ao todo) possui até 50 casas e 10 edifícios, cada edifício possui até 20 andares e cada andar tem até 2 apartamentos. Cada casa ou apartamento tem capacidade para 4 pessoas.

A enumeração das casas, edifícios, andares e apartamentos na entrada e saída começam por 1.

Formato de entrada

Um inteiro 1 <= N <= 3 indicando qual operação será executada em seguida:

Com N = 1:

- Uma string para o nome da rua;

A partir daqui, há duas possibilidades de entrada, a depender se o morador se encontra em uma casa ou prédio:

- Para casas, segue-se a leitura de, em ordem:
 - O caractere 'C', indicando que o morador vai se encontrar em uma casa;
 - Um inteiro C indicando o número da casa;
 - Uma string indicando o nome do morador;
 - Um caractere S ('M' ou 'F') indicando o sexo do morador;
 - A data de nascimento do morador no formato dd/mm/aaaa.
- Para prédios, segue-se a leitura de, em ordem:
 - Três inteiros, indicando, respectivamente, o número do edifício E, o andar A do morador e o número APT do seu apartamento;
 - Uma string indicando o nome do morador;
 - Um caractere 'M' ou 'F' indicando o sexo do morador;
 - A data de nascimento do morador no formato dd/mm/aaaa.

Com N = 2:

- Duas strings M e R, indicando, respectivamente, o nome do morador e em qual rua ele deve ser procurado.

Cada operação é separada por uma linha em branco.

Formato de saída

A cada chamada da segunda operação, devem ser retornado os dados dos X moradores encontrados que se encontram nos parâmetros de busca, da seguinte maneira:

- Caso algum morador tenha sido encontrado:
 - Caso ele more em uma casa:
"~Casa: C | Sexo: S | Nascimento: dd/mm/aaaa", seguido de uma quebra de linha.
 - Caso ele more em um prédio:
"Edifício: E | Andar: A | Numero do apt.: APT | Sexo: S | Nascimento: dd/mm/aaaa", seguido de uma quebra de linha.
 - Ao final da busca, deve ser retornado:
"Foram encontradas X instancias de pessoas chamadas "M" em "R", relatadas acima.", seguido de uma quebra de linha.
 - Os moradores encontrados são ordenados pelo número da casa, número do prédio, andar do prédio, número do apartamento e ordem de inserção, nessa ordem.
- Caso nenhum morador tenha sido encontrado:
 - "Sem dados de "M" em "R!", seguido de uma quebra de linha.

Na chamada da terceira operação, a seguinte string deve ser retornada, e o programa encerrado:

"O programa sera fechado, obrigado por fazer uso dele. Lembre-se de evitar a fadiga!"

Caso a operação inserida não seja válida, deve-se retornar a seguinte string, e o programa volta a esperar por uma entrada:

"Opcao invalida. Por favor, digite uma opcao valida.", seguido de uma quebra de linha.

Exemplos de:

Entrada

1

Rua A

c 12

Juvenal

M 01/01/2000

2

Juvenal

Rua B

2

Juvenal

Rua C

2

Juvenal

Rua D

2

Juvenal

Rua E

3

Saida

Sem dados de "Juvenal" em "Rua B"!

Sem dados de "Juvenal" em "Rua C"!

Sem dados de "Juvenal" em "Rua D"!

Sem dados de "Juvenal" em "Rua E"!

O programa sera fechado, obrigado por fazer uso dele. Lembre-se de evitar a fadiga!

Entrada

1

Rua A

c 24

Juvenal

M 01/01/2000

1

Rua A

c 24

Juvenal

M 02/01/2000

1

Rua A

c 24

Juvenal

M 03/01/2000

1

Rua A

c 24

Juvenal

M 04/01/2000

1

Rua A

c 25

Juvenal

M 05/01/2000

1

Rua A

e 10 20 2

Juvenal

M 06/01/2000

2

Juvenal

Rua A

3

Saida

Casa: 24 | Sexo: M | Nascimento: 01/01/2000

Casa: 24 | Sexo: M | Nascimento: 02/01/2000

Casa: 24 | Sexo: M | Nascimento: 03/01/2000

Casa: 24 | Sexo: M | Nascimento: 04/01/2000

Casa: 25 | Sexo: M | Nascimento: 05/01/2000

Edificio: 10 | Andar: 20 | Numero do apt.: 2 | Sexo: M | Nascimento: 06/01/2000

Foram encontradas 6 instancias de pessoas chamadas "Juvenal" em "Rua A", relatadas acima.

O programa sera fechado, obrigado por fazer uso dele. Lembre-se de evitar a fadiga!

REPORTAR PROBLEMA

TIRAR DÚVIDA

ENVIAR RESPOSTA