

Centro de Educação Superior a Distância do
Estado do Rio de Janeiro – CEDERJ

Curso de Tecnologia em Sistemas de Computação – TSC

EAD-05.009 Fundamentos de Programação

Caderno de Exercícios

Aula 12

(Persistência de Dados: Arquivos Binários)

Professores

Dante Corbucci Filho
Leandro A. F. Fernandes

Instruções

- Utilize Python 3 e a IDE PyCharm na elaboração de soluções para os problemas propostos;
- A entrada de cada problema deve ser lida da entrada padrão (teclado);
- A saída de cada problema deve ser escrita na saída padrão (tela);
- Siga o formato apresentado na descrição da saída, caso contrário não é garantido que a saída emitida será considerada correta;
- Na saída, toda linha deve terminar com o caractere `'\n'` ;
- Utilize o URI Online Judge (<http://www.urionlinejudge.com.br>) e submeta sua solução para correção automática.

Referências Autorais

Os exercícios apresentados nesta lista foram extraídos do URI Online Judge (<http://www.urionlinejudge.com.br>). Acesse a URL apresentada abaixo do título de cada problema para proceder com a correção automática de sua solução e, também, para consultar a autoria do enunciado.

Problema A: Lanche

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1038>

Observação: Após o seu programa funcionar no URI, carregue a tabela no arquivo `tabela.bin` e a sua saída deve ser escrita no arquivo `saída.bin`.

Com base na tabela abaixo, escreva um programa que leia o código de um item e a quantidade deste item. A seguir, calcule e mostre o valor da conta a pagar.

| CODIGO | ESPECIFICAÇÃO | PREÇO |
|--------|-----------------|----------|
| 1 | Cachorro Quente | R\$ 4.00 |
| 2 | X-Salada | R\$ 4.50 |
| 3 | X-Bacon | R\$ 5.00 |
| 4 | Torrada simples | R\$ 2.00 |
| 5 | Refrigerante | R\$ 1.50 |

Entrada

O arquivo de entrada contém dois valores inteiros correspondentes ao código e à quantidade de um item conforme tabela acima.

Saída

O arquivo de saída deve conter a mensagem "Total: R\$ " seguido pelo valor a ser pago, com 2 casas após o ponto decimal.

Exemplo

| Entrada | Saída |
|---------|------------------|
| 3 2 | Total: R\$ 10.00 |

| Entrada | Saída |
|---------|-----------------|
| 4 3 | Total: R\$ 6.00 |

| Entrada | Saída |
|---------|------------------|
| 2 3 | Total: R\$ 13.50 |

Problema B: Sort Simple

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1042>

Observação: Após o seu programa funcionar no URI, carregue vários valores de entrada através de um arquivo chamado `entrada.bin` e a sua saída deve ser escrita no arquivo `saída.bin`.

Leia 3 valores inteiros e ordene-os em ordem crescente no arquivo, sem jogar todos os valores na memória principal. No final, mostre os valores em ordem crescente, uma linha em branco e em seguida, os valores na sequência como foram lidos.

Entrada

A entrada contém três números inteiros.

Saída

Imprima a saída conforme foi especificado.

Exemplo

| Entrada | Saída |
|----------|--------------------------------------|
| 7 21 -14 | -14 7 21 7 21 -14 |

| Entrada | Saída |
|----------|--------------------------------------|
| -14 21 7 | -14 7 21 -14 21 7 |

Problema C: Pares entre Cinco Números

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1065>

Observação: Após o seu programa funcionar no URI, carregue vários valores de entrada através de um arquivo chamado `entrada.bin` e a sua saída deve ser escrita no arquivo `saída.bin`.

Faça um programa que leia 5 valores inteiros. Conte quantos destes valores digitados são pares e mostre esta informação.

Entrada

O arquivo de entrada contém 5 valores inteiros quaisquer.

Saída

Imprima a mensagem conforme o exemplo fornecido, indicando a quantidade de valores pares lidos.

Exemplo

| Entrada | Saída |
|--------------------------|-----------------|
| 7 -5 6 -4 12 | 3 valores pares |

Problema D: Tabuada

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1078>

Observação: Após o seu programa funcionar no URI, carregue vários valores de entrada através de um arquivo chamado `entrada.bin` e a sua saída deve ser escrita no arquivo `saída.bin`.

Leia 1 valor inteiro N ($2 < N < 1000$). A seguir, mostre a tabuada de N :
 $1 \times N = N$ $2 \times N = 2N$... $10 \times N = 10N$

Entrada

A entrada contém um valor inteiro N ($2 < N < 1000$).

Saída

Imprima a tabuada de N , conforme o exemplo fornecido.

Exemplo

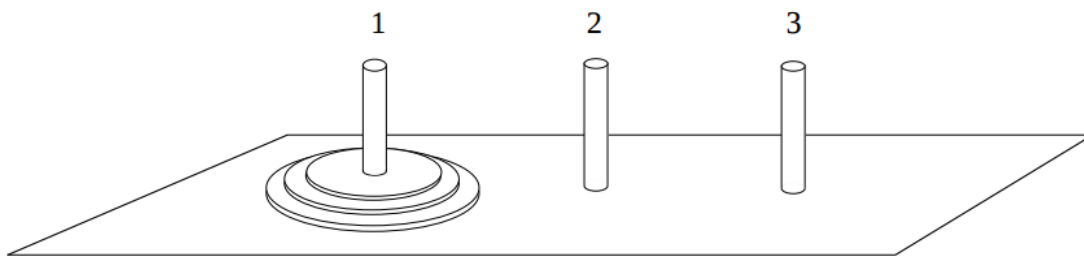
| Entrada | Saída |
|---------|--|
| 140 | $1 \times 140 = 140$ $2 \times 140 = 280$ $3 \times 140 = 420$ $4 \times 140 = 560$ $5 \times 140 = 700$ $6 \times 140 = 840$ $7 \times 140 = 980$ $8 \times 140 = 1120$ $9 \times 140 = 1260$ $10 \times 140 = 1400$ |

Problema E: Torres de Hanói

<https://www.urionlinejudge.com.br/judge/pt/problems/view/2251>

Observação: Após o seu programa funcionar no URI, carregue vários valores de entrada através de um arquivo chamado `entrada.bin` e a sua saída deve ser escrita no arquivo `saída.bin`.

O quebra-cabeças Torres de Hanoi é muito antigo e conhecido, sendo constituído de um conjunto de N discos de tamanhos diferentes e três pinos verticais, nos quais os discos podem ser encaixados.



Cada pino pode conter uma pilha com qualquer número de discos, desde que cada disco não seja colocado acima de outro disco de menor tamanho. A configuração inicial consiste de todos os discos no pino 1. O objetivo do quebra-cabeças é mover todos os discos para um dos outros pinos, sempre obedecendo à restrição de não colocar um disco sobre outro menor.

Um algoritmo para resolver este problema é o seguinte.

procedimento Hanoi(N , Orig, Dest, Temp)

se $N = 1$ **então**

mover o menor disco do pino Orig para o pino Dest;

senão

Hanoi($N-1$, Orig, Temp, Dest);

mover o N -ésimo menor disco do pino Orig para o pino Dest;

Hanoi($N-1$, Temp, Dest, Orig);

fim-se

fim

Sua tarefa é escrever um programa que determine quantos movimentos de trocar um disco de um pino para outro serão executados pelo algoritmo acima para resolver o quebra-cabeça.

Entrada

A entrada possui vários conjuntos de teste. Cada conjunto de teste é composto por uma única linha, que contém um único número inteiro N ($0 \leq N \leq 30$), indicando o número de discos. O final da entrada é indicado por $N = 0$.

Saída

Para cada conjunto de teste, o seu programa deve escrever três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n”, onde n é numerado sequencialmente a partir de 1. A segunda linha deve conter o número de movimentos que são executados pelo algoritmo dado para resolver o problema das Torres de Hanói com N discos. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

| Entrada | Saída |
|---------|---------|
| 1 | Teste 1 |
| 2 | 1 |
| 0 | Teste 2 |
| | 3 |

Problema F: Par ou Ímpar

<https://www.urionlinejudge.com.br/judge/pt/problems/view/2286>

Observação: Após o seu programa funcionar no URI, carregue vários valores de entrada através de um arquivo chamado `entrada.bin` e a sua saída deve ser escrita no arquivo `saída.bin`.

Muitas crianças gostam de decidir todas as disputas através do famoso jogo de Par ou Ímpar. Nesse jogo, um dos participantes escolhe Par e o outro Ímpar. Após a escolha, os dois jogadores mostram, simultaneamente, uma certa quantidade de dedos de uma das mãos. Se a soma dos dedos das mãos dos dois jogadores for par, vence o jogador que escolheu Par inicialmente, caso contrário vence o que escolheu Ímpar.

Dada uma sequência de informações sobre partidas de Par ou Ímpar (nomes dos jogadores e números que os jogadores escolheram), você deve escrever um programa para indicar o vencedor de cada uma das partidas.

Entrada

A entrada é composta de vários conjuntos de testes. A primeira linha de um conjunto de testes contém um inteiro N ($0 \leq N \leq 1000$), que indica o número de partidas de Par ou Ímpar que aconteceram. As duas linhas seguintes contém cada uma um nome de jogador. Um nome de jogador é uma cadeia de no mínimo um e no máximo dez letras (maiúsculas e minúsculas), sem espaços em branco. As N linhas seguintes contém cada uma dois inteiros A e B que representam o número de dedos que cada jogador mostrou em cada partida ($0 \leq A \leq 5$ e $0 \leq B \leq 5$). Em todas as partidas, o primeiro jogador sempre escolhe Par. O final da entrada é indicado por $N = 0$.

Saída

Para cada conjunto de teste da entrada, seu programa deve produzir a saída da seguinte forma. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste n ", onde n é numerado sequencialmente a partir de 1. As próximas N linhas devem indicar o nome do vencedor de cada partida. A próxima linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

| Entrada | Saída |
|---------|---------|
| 3 | Teste 1 |
| Pedro | Pedro |
| Paulo | Pedro |
| 2 4 | Paulo |
| 3 5 | |
| 1 0 | |
| 2 | Teste 2 |
| Claudio | Claudio |
| Carlos | Carlos |
| 1 5 | |
| 2 3 | |
| 0 | |