

Jogo do bicho

Trabalho 1

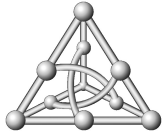
Algoritmos e Programação I

1 Descrição

Em um país muito muito distante, as pessoas são viciadas em um jogo de apostas bastante simples. O jogo é baseado em números e é chamado de *jogo do bicho*. Há um bom tempo, o jogo do bicho foi proibido pelas autoridades locais, mas ainda é amplamente difundido e as pessoas ainda o jogam bastante, mas de forma clandestina. O nome do jogo deriva do fato que os números são divididos em 25 grupos, dependendo do valor dos dois últimos dígitos (dezenas e unidades), e cada grupo recebe o nome de um animal. Cada grupo é associado a um animal da seguinte forma: o primeiro grupo (avestruz) consiste nos números 01, 02, 03 e 04; o segundo grupo (águia), consiste nos números 05, 06, 07 e 08; e assim por diante, até o último grupo (vaca) contendo os números 97, 98, 99 e 00. As regras do jogo são simples. No momento da aposta, o jogador decide o valor da aposta V e um número N , com $0 \leq N \leq 1000000$. Todos os dias, na praça principal de toda cidade do país, um número M é sorteado, onde $0 \leq M \leq 1000000$. O prêmio de cada apostador é calculado da seguinte forma:

01-02-03-04 #01 AVESTRUZ	05-06-07-08 #02 ÁGUIA	09-10-11-12 #03 BURRO	13-14-15-16 #04 BORBOLETA	17-18-19-20 #05 CACHORRO
21-22-23-24 #06 CABRA	25-26-27-28 #07 CARNEIRO	29-30-31-32 #08 CAMELO	33-34-35-36 #09 COBRA	37-38-39-40 #10 COELHO
41-42-43-44 #11 CAVALO	45-46-47-48 #12 ELEFANTE	49-50-51-52 #13 GALO	53-54-55-56 #14 GATO	57-58-59-60 #15 JACARÉ
61-62-63-64 #16 LEÃO	65-66-67-68 #17 MACACO	69-70-71-72 #18 PORCO	73-74-75-76 #19 PAVÃO	77-78-79-80 #20 PERÔ
81-82-83-84 #21 TOURO	85-86-87-88 #22 TIGRE	89-90-91-92 #23 URSO	93-94-95-96 #24 VEADO	97-98-99-00 #25 VACA

- se M e N têm os mesmos quatro últimos dígitos (milhar, centena, dezena e unidade), como por exemplo $N = 99301$ e $M = 19301$, o apostador recebe $V \times 3000$ na moeda local;
- se M e N têm os mesmos três últimos dígitos (centena, dezena e unidade), como por exemplo $N = 38944$ e $M = 83944$, o apostador recebe $V \times 500$;
- se M e N têm os mesmos dois últimos dígitos (dezena e unidade), como por exemplo $N = 111$ e $M = 552211$, o apostador recebe $V \times 50$;
- se M e N têm os dois últimos dígitos no mesmo grupo, correspondendo ao mesmo animal, como por exemplo $N = 82197$ e $M = 337600$, o apostador recebe $V \times 16$;
- se nenhum dos casos acima ocorrer, o apostador não recebe valor algum.



Obviamente, o prêmio dado a cada apostador é o máximo possível de acordo com as regras acima. No entanto, não é possível acumular prêmios, de forma que apenas um dos critérios acima deve ser aplicado no cálculo do prêmio. Se um número N ou M com menos de quatro dígitos for apostado ou sorteado, considere que dígitos 0 (zero) ocorrem na frente do número, de modo que tenha os quatro dígitos. Por exemplo, se 17 é um número apostado ou sorteado, considere 0017. Além disso, por ser um jogo clandestino, o prêmio é pago sempre em dinheiro.

Considere que a moeda local desse país é o *pila* (P\$). Pila não possui plural, sendo que uma unidade ou qualquer quantidade da moeda continua a se chamar pila. Além disso, existem apenas notas de 1, 10 e 100 pila (não há centavos). Dado um valor apostado, o número escolhido pelo apostador e um número sorteado, seu programa deve calcular qual prêmio o apostador deve receber, além de determinar como pagá-lo utilizando a menor quantidade de notas possível.

Seu programa **deve** trabalhar com números inteiros, ou seja, **não** é permitido trabalhar com dígitos em *strings*.



2 Entrada e saída

A entrada contém vários casos de teste. Cada caso de teste consiste em apenas uma linha, contendo três números inteiros V , N e M , representando respectivamente o valor da aposta V , com $1 \leq V \leq 1000$, o número escolhido para a aposta N , com $0 \leq N \leq 1000000$, e o número sorteado M , com $0 \leq M \leq 1000000$. O final da entrada é indicado por uma linha contendo $V = N = M = 0$.

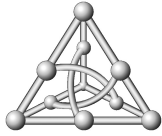
Para cada um dos casos de teste, seu programa deve imprimir uma linha contendo um número inteiro, representando o valor do prêmio correspondente à aposta dada, e a quantidade de notas utilizadas, no formato $C \times 100 \ D \times 10 \ U \times 1$, onde C , D e U são as quantidades de notas de cem, dez e um pila, respectivamente. Os dados de saída de cada caso de teste devem ser separados entre si por um espaço e conter uma quebra de linha no final.

3 Exemplo de entrada

```
32 32 213929
10 32 213032
1000 340000 0
520 874675 928567
10 1111 578311
12 1343 245841
0 0 0
```

4 Exemplo de saída

```
512 5x100 1x10 2x1
5000 50x100 0x10 0x1
3000000 30000x100 0x10 0x1
0 0x100 0x10 0x1
500 5x100 0x10 0x1
192 1x100 9x10 2x1
```



5 Entrega

Instruções para entrega do seu trabalho:

1. Cabeçalho

Seu trabalho deve ter um cabeçalho com o seguinte formato:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

#
# Nome do(a) estudante
# Trabalho 1
# Professor(a): Nome do(a) professor(a)
#
```

2. Forma de entrega

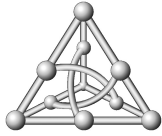
A entrega será realizada diretamente no Ambiente Virtual de Aprendizagem - UFMS (<https://ava.ufms.br>), na nossa disciplina de Algoritmos e Programação I. Um fórum de discussão deste trabalho já se encontra aberto. Para entrega do trabalho, vá até o tópico “Trabalhos”, e escolha “T1 - Entrega”. Você pode entregar o trabalho quantas vezes quiser até às **23 horas e 59 minutos** do dia **03 de maio de 2020**. A última versão entregue é aquela que será corrigida. Encerrado o prazo, não serão mais aceitos trabalhos.

3. Atrasos

Trabalhos atrasados não serão aceitos. Não deixe para entregar seu trabalho na última hora. Para prevenir imprevistos como queda de energia, problemas com o sistema, falha de conexão com a internet, sugerimos que a entrega do trabalho seja feita pelo menos um dia antes do prazo determinado.

4. Erros

Trabalhos com erros de execução (*Runtime error*) receberão nota **ZERO**. Faça todos os testes necessários para garantir que seu programa está livre de erros de execução.



5. O que entregar?

Você deve entregar um único arquivo contendo **APENAS** o seu programa fonte com o mesmo nome de seu nome e um sobrenome, como por exemplo, `fulano_silva.py`.

6. Verificação dos dados de entrada

Não se preocupe com a verificação dos dados de entrada do seu programa. Seu programa não precisa fazer consistência dos dados de entrada. Isto significa que se, por exemplo, o seu programa pede um número entre 1 e 10, você pode assumir que quem executar seu programa nunca irá digitar um valor fora desse intervalo.

7. Arquivo com o programa fonte

Seu arquivo contendo o código Python deve estar bem organizado. Um programa em qualquer linguagem tem de ser muito bem compreendido por uma pessoa. Verifique se seu programa tem a indentação adequada, se não tem linhas muito longas, se tem variáveis com nomes significativos, entre outros. Não esqueça que um programa bem descrito e bem organizado é a chave de seu sucesso.

8. Conduta Ética

O trabalho deve ser feito **INDIVIDUALMENTE**. Cada estudante tem responsabilidade sobre cópias de seu trabalho, mesmo que parciais. Não faça o trabalho em grupo e não compartilhe seu programa ou trechos de seu programa. Você pode consultar seus colegas para esclarecer dúvidas e discutir idéias sobre o trabalho, ao vivo ou no fórum de discussão da disciplina, mas **NÃO** copie o programa!

Trabalhos considerados plagiados terão nota **ZERO**. Estudante que se envolver em **DOIS CASOS DE PLÁGIO** estará automaticamente **REPROVADO** na disciplina.