

MC102W - Algoritmos e Programação de Computadores

Lab07: Guerra 4.0

Prazo: 10 de Maio de 2020

Peso da Atividade: 3

Os reinos de Snowland e Sunny Kingdom são inimigos de longa data, tão longa que até mesmo os mais antigos livros narram histórias de suas disputas. Nenhum habitante vivo conhece o real motivo que originou tal disputa, mas todos carregam dentro de si o ódio inato pelo reino adversário. Diversas guerras sangrentas foram travadas entre os dois reinos, originadas pelos mais tolos motivos como qual reino receberia primeiro o trovador Elton Johnson do reino de Lala Land. Apesar de todo o impacto social e econômico causado por esses conflitos sem sentido, nenhum dos dois reinos tinha intenção de interromper a eterna rivalidade. Após ouvir a história dos dois reinos, um sábio mago chamado Mestre Alan Turing do reino de Paddington propôs uma inovadora solução para as disputas dos dois reinos, a Guerra 4.0. A Guerra 4.0 seria realizada em um artefato recém criado pelo Mestre Alan, o computador, e serviria para decidir qualquer conflito entre os dois reinos sem os impactos negativos de uma guerra convencional. Porém, Sir Alan já tem muitas tarefas no seu atual projeto enigmático e não dispõe de mais tempo para gerar a solução proposta. Por isso, ele convoca você para desenvolver a Guerra 4.0, mas shhh... não deixe que os Alemães saibam.

Tarefa

Na Guerra 4.0, cada reino irá indicar apenas um guerreiro para o duelo. Esse guerreiro será representado no computador através de seus atributos:

- ❑ Nome: cadeia de caracteres;
- ❑ Max_vida: número inteiro positivo representando a saúde inicial do guerreiro. A vida será reduzida a cada ataque recebido e pode ser recuperada com magias, mas nunca ultrapassa o seu valor máximo;
- ❑ Dano: número inteiro positivo representando o tanto que cada ataque do guerreiro reduz da vida do oponente;
- ❑ Bloqueio: número inteiro positivo no intervalo [0,100] representando a porcentagem do dano recebido que não afetará o guerreiro;
- ❑ Max_mana: número inteiro positivo representando o número inicial de pontos para cartas mágicas. O mana é reduzido nas ativações das cartas mágicas e também pode ser recuperado, mas nunca passa do valor máximo.

Os heróis também possuem as seguintes ações básicas:

- ❑ Atacar: causa dano no inimigo de acordo com seu atributo e ativa a defesa do oponente;

- ❑ Defender: bloqueia parte do dano causado, indicando o dano real que deve ser descontado da vida.

Os atributos e ações podem ser modificados ao longo do duelo, caso o personagem encontre algumas cartas mágicas. Essas cartas podem ter custo associado e a magia será ativada somente se o herói possuir mana suficiente para isso. Caso contrário, seu programa deve gerar uma mensagem de mana insuficiente a cada tentativa. As cartas mágicas e seus efeitos estão listados na tabela a seguir:

Carta Mágica	Representação	Efeito
Mágicas de uso imediato (efeito único)		
Cura	C <custo> <pontos>	Recupera <pontos> pontos de vida a um custo <custo> de pontos de mana.
Força	F <custo> <pontos>	Aumenta o dano básico em <pontos> a um custo <custo> de pontos de mana.
Proteção	P <custo> <percentual>	Aumenta o bloqueio em <percentual> por cento, com limite superior de 100, a um custo <custo> de pontos de mana.
Éter	E <pontos>	Recupera <pontos> pontos de mana. Sem custo.
Mágicas passivas (sem custo)		
Drenagem	D <pontos>	Cada ataque desferido pelo portador da mágica reduz <pontos> pontos de mana do adversário. Sem custo.
Mágicas de ativação (custo único na ativação)		
Insano	I <custo> <n> <dano_adicional>	Após a ativação, os próximos <n> ataques do herói terão um dano adicional de <dano_adicional>. Custa <custo> pontos de mana na ativação.
Estrela	S <custo> <n>	Após a ativação, torna o herói invulnerável aos <n> próximos ataques recebidos. Custa <custo> pontos de mana na ativação.

Uma rodada é formada por um grupo de ações de um herói seguido do grupo de ações do adversário. Após iniciar seu turno na rodada, um herói executa as seguintes ações, nessa ordem: (1) pode encontrar uma magia, (2) pode ativar nenhuma ou várias mágicas encontradas e (3) obrigatoriamente ataca o adversário, ativando a defesa dele. O duelo termina assim que qualquer herói atingir vida 0. As batalhas sempre acabam em vitória.

O programa deve ser em Python e deve utilizar os conceitos de classes e objetos. O programa deve conter pelo menos a classe Heroi.

Entrada

A entrada do programa é dividida em duas partes. Primeiro, os reis de Snowland e Sunny Kingdom (nessa ordem) indicam as características do seu herói escolhido.

```
<nome_snow>
<max_vida_snow>
<dano_snow>
<bloqueio_snow>
<max_mana_snow>
<nome_sunny>
<max_vida_sunny>
<dano_sunny>
<bloqueio_sunny>
<max_mana_sunny>
```

Na segunda parte, uma sequência de ações é indicada:

- ❑ O início do grupo de ações de um herói é marcado pela linha “H 1” para o herói de Snowland e “H 2” para o herói de Sunny Kingdom.
- ❑ Após a marcação inicial, é informado se o herói encontrou alguma carta mágica nessa rodada, e é indicado na forma “M <representacao>”, onde <representacao> é a representação dada na tabela. Caso o herói não tenha encontrado nenhuma carta mágica, a linha será “M X”. As magias de efeito imediato e passivas já devem ser executadas assim que forem encontradas, se houver mana. Caso contrário, a carta será ignorada. As cartas mágicas de ativação são armazenadas até que o herói decida utilizar, mas essas mágicas não são acumuláveis, ou seja, ele não poderá ativar dois ataques insanos, por exemplo. Se a magia ativa não foi utilizada ou está em andamento e o herói encontra o mesmo tipo de magia, a nova deve ser ignorada. O mesmo vale para magias passivas, se o herói já possui a carta drenagem, por exemplo, toda nova drenagem deve ser ignorada.
- ❑ Em seguida, o herói pode ativar uma sequência de magias de ativação, desde que elas tenham sido encontradas nessa rodada ou em alguma anterior. Seu programa deve informar se a carta mágica está disponível ou não. Não existe limite de ativações, ou seja, o herói pode ativar insano e estrela no mesmo turno. A ativação é indicada pelas mesmas letras dadas na representação das mágicas sem nenhum parâmetro. Por exemplo, a linha “I” indica ativação da carta insano.

- ❑ O bloco de ações do herói termina com um ataque ao adversário, levando em conta toda modificação causada pelas mágicas nessa rodada. O ataque é indicado apenas pela letra "A".

Observação:

- ❑ Para a leitura de cada linha, utilize **`linha = input().split(" ")`**, que colocará os caracteres e os parâmetros dentro de uma lista. Exemplo: A leitura da linha "M C 20 30" resultará na lista ['M', 'C', '20', '30'].
- ❑ O dano bloqueado na defesa deve ser calculado com **`dano_bloqueado = int((dano * bloqueio) / 100)`** para evitar divergências de arredondamento.
- ❑ Tanto a vida quanto mana tem valores mínimos iguais a 0. Se um herói tem 40 de vida e leva um dano de 50, sua vida resultante é 0.

Saída

O seu programa deve reportar os eventos ocorrendo durante o duelo e ao final informar quem ganhou o duelo. As possíveis mensagens são:

Indicação de herói:

"O reino Snowland indicou o herói <nome_snow>" # obrigatória

"O reino Sunny Kingdom indicou o herói <nome_sunny>" # obrigatória

Início do turno de um herói (grupo de ações):

"Rodada <i>: vez de <nome_heroi>" # obrigatória

Encontrar carta mágica:

"<nome_heroi> encontrou a carta <magia>" # M <representacao>

"<nome_heroi> não encontrou nenhuma carta" # M X

"<nome_heroi> não possui mana suficiente para a mágica" # sem mana para mágica de uso imediato ou passiva

"<nome_heroi> já possui a carta <magia>" # não acumular passivas e de ativação

Utilizar magias de ativação:

"<nome_heroi> ativou a carta <magia>" # I ou E

"<nome_heroi> não possui mana suficiente para a mágica" # sem mana para mágica de ativação

"<nome_heroi> não possui a carta <magia>" # tentou ativar uma carta que não possui

"<nome_heroi> já ativou a carta <magia>" # tentou ativar uma carta que já estava ativa

Ataque:

"<nome_heroi> atacou <nome_inimigo>" # ataque sem insano

"<nome_heroi> deu um ataque insano em <nome_inimigo>" # ataque com insano

"<nome_inimigo> estava invulnerável" # defesa com estrela

Ao final de uma rodada (dois turnos, um de cada herói):

"<nome_snow> possui <vida> de vida, <mana> pontos mágicos, <dano> de dano e <bloqueio>% de bloqueio" # status herói de snowland

"<nome_sunny> possui <vida> de vida, <mana> pontos mágicos, <dano> de dano e <bloqueio>% de bloqueio" # status herói de sunny kingdom

Final:

"O herói <nome_snow> do reino Snowland venceu o duelo" # se snowland venceu

"O herói <nome_sunny> do reino Sunny Kingdom venceu o duelo" # se sunny kingdom venceu

"<nome_snow> possui <vida> de vida, <mana> pontos mágicos, <dano> de dano e <bloqueio>% de bloqueio" # status final do herói de snowland

"<nome_sunny> possui <vida> de vida, <mana> pontos mágicos, <dano> de dano e <bloqueio>% de bloqueio" # status final do herói de sunny kingdom

Obs:

- Os nomes das cartas mágicas seguem a mesma grafia da tabela, inclusive a primeira letra em maiúsculo.
- Cada mensagem deve ser impressa em uma linha diferente.

Exemplo

Exemplo 1:

Entrada

```
Heroi1
300
50
5
100
Heroi2
300
50
5
100
H 1
M X
A
H 2
M X
A
H 1
M I 10 1 300
I
A
```

Saída

```
O reino Snowland indicou o herói Heroi1.
```

```
O reino Sunny Kingdom indicou o herói Heroi2.  
Rodada 1: vez de Heroi1  
Heroi1 não encontrou nenhuma carta  
Heroi1 atacou Heroi2  
Rodada 1: vez de Heroi2  
Heroi2 não encontrou nenhuma carta  
Heroi2 atacou Heroi1  
Heroi1 possui 252 de vida, 100 pontos mágicos, 50 de dano e 5% de  
bloqueio  
Heroi2 possui 252 de vida, 100 pontos mágicos, 50 de dano e 5% de  
bloqueio  
Rodada 2: vez de Heroi1  
Heroi1 encontrou a carta Insano  
Heroi1 ativou a carta Insano  
Heroi1 deu um ataque insano em Heroi2  
O herói Heroi1 do reino Snowland venceu o duelo  
Heroi1 possui 252 de vida, 90 pontos mágicos, 50 de dano e 5% de  
bloqueio  
Heroi2 possui 0 de vida, 100 pontos mágicos, 50 de dano e 5% de  
bloqueio
```

Exemplo 2:

Entrada

```
Loki  
300  
30  
40  
1000  
Hulk  
300  
700  
70  
0  
H 1  
M P 20 15  
A  
H 2  
M S 5 10  
S  
A
```

Saída

```
O reino Snowland indicou o herói Loki.  
O reino Sunny Kingdom indicou o herói Hulk.  
Rodada 1: vez de Loki  
Loki encontrou a carta Proteção  
Loki atacou Hulk
```

```
Rodada 1: vez de Hulk
Hulk encontrou a carta Estrela
Hulk não possui mana suficiente para a mágica
Hulk atacou Loki
Loki possui 0 de vida, 980 pontos mágicos, 30 de dano e 55% de
bloqueio
Hulk possui 291 de vida, 0 pontos mágicos, 700 de dano e 70% de
bloqueio
O herói Hulk do reino Sunny Kingdom venceu o duelo
Loki possui 0 de vida, 980 pontos mágicos, 30 de dano e 55% de
bloqueio
Hulk possui 291 de vida, 0 pontos mágicos, 700 de dano e 70% de
bloqueio
```

Critérios específicos

Os seguintes critérios específicos sobre o envio, implementação e execução devem ser satisfeitos.

i. Submeter no SuSy o arquivo:

⇒ `lab07.py`: Arquivo contendo todo o seu programa.

ii. Não serão aceitas soluções contendo estruturas não vistas em sala, exceto as indicadas neste enunciado.

Observações gerais

No decorrer do semestre haverá 3 tipos de tarefas no SuSy (descritas logo abaixo). As tarefas possuirão os mesmos casos de testes abertos e fechados, no entanto o número de submissões permitidas e prazos são diferentes. As seguintes tarefas estão disponíveis no SuSy:

- ❑ **Lab07-AmbienteDeTeste:** Esta tarefa serve para testar seu programa no SuSy antes de submeter a versão final. Nessa tarefa, tanto o prazo quanto o número de submissões são ilimitados, porém os arquivos submetidos aqui **não serão corrigidos**.
- ❑ **Lab07-Entrega:** Esta tarefa tem limite de uma **única** submissão e serve para entregar a **versão final** dentro do prazo estabelecido para o laboratório. Não use essa tarefa para testar o seu programa e submeta aqui apenas quando não for mais fazer alterações no seu programa.
- ❑ **Lab07-ForaDoPrazo:** Esta tarefa tem limite de uma **única** submissão e serve para entregar a versão final fora prazo estabelecido para o laboratório. Esta tarefa irá substituir a nota obtida na tarefa **Lab07-Entrega** apenas se o aluno tiver

realizado as correções sugeridas no *feedback* ou caso não tenha enviado anteriormente na tarefa **Lab07-Entrega**.

Avaliação

Este laboratório será avaliado da seguinte maneira: a nota será proporcional ao número de casos **fechados** para os quais o seu programa gerou a resposta correta, **desde que os critérios indicados neste enunciado tenham sido atendidos**. Se o programa apresentou resposta correta para todos os casos, a nota será 10; caso contrário será $p \cdot 10$, onde p é o percentual de respostas corretas. A nota também poderá sofrer descontos de acordo com a qualidade do programa apresentado. Assim, mesmo que o código seja capaz de resolver todos os casos de teste fechados, a nota final ainda pode ser menor do que 10. Por isso, acrescente comentários explicativos, utilize variáveis sugestivas e faça um código claro e de acordo com o que foi solicitado.

Testando seu programa

Para testar se a solução do seu programa está correta, basta seguir o exemplo abaixo no terminal do Linux.

```
python lab07.py < arq01.in > arq01.out
diff arq01.out arq01.res
```

O `arq01.in` é a entrada e `arq01.res` é a saída esperada, ambos disponíveis no SuSy. O `arq01.out` é a saída gerada pelo seu programa. Após o prazo, os casos de teste fechados serão liberados e podem ser baixados e testados da mesma forma que os testes abertos.