

MC102QR - Algoritmos e Programação de Computadores

Lab 5 - Funções

Prazo da atividade: 01 de Maio de 2022

Peso na nota: 2 (4,88%)

Você é um desenvolvedor de jogos, e cabe agora a você programar a batalha final do mais novo projeto de sua empresa. No clímax dessa aventura, a nobre heroína Sarah adentrou o castelo das trevas e derrotou o tirano que assolava seu reino. Porém, resta vencer mais um desafio antes que a paz possa retornar: ela mesma. O tirano criou uma cópia maligna de Sarah como seu último recurso. Sarah e seu clone possuem atributos distintos (pontos de vida, ataque e defesa), porém suas habilidades são as mesmas, descritas a seguir:

- **Soco:** Um simples ataque que remove pontos de vida do oponente. O valor base de dano causado é o valor de ataque do atacante menos o valor de defesa do oponente, e esse dano base é então multiplicado por um número aleatório. Caso o dano base seja negativo, o ataque causa 0 pontos de dano.
- **Arremesso de facas:** Um ataque de n golpes sucessivos, onde cada i -ésimo golpe causa dano no oponente igual à divisão inteira do ataque do atacante por 3^i . A quantidade n de golpes é um número gerado aleatoriamente.
- **Invocação de fada:** Durante sua jornada, Sarah conseguiu formar aliança com fadas, e agora pode invocá-las em seu auxílio. Fadas usam magia para curar o invocador, porém, como são criaturas brincalhonas e voláteis, a quantidade de pontos de vida ganhos também é um número aleatório, e outros efeitos aleatórios podem acontecer, como aumentar o ataque ou a defesa do invocador. Se Sarah tiver muita sorte mesmo, a batalha é encerrada de forma bizarra, invocando um monstro de uma tonelada que destrói todo o castelo.

A batalha se dá conforme as eternas tradições do reino, com os combatentes alternando turnos em que usam uma de suas habilidades, sendo o primeiro turno sempre da heroína (o reino de Sarah foi fundado por jogadores de xadrez).

Para manter o código organizado, sua empresa quer que você programe cada uma das habilidades das personagens como uma função. O código principal deve apenas imprimir uma mensagem após o retorno de cada função, descrevendo o resultado da ação. Você também deve criar uma função que roda ao fim de cada turno, checando se algum personagem foi derrotado, e encerrando o programa caso isso aconteça.

Tarefa

Você deve implementar as funções `soco`, `arremesso_de_facas`, e `invocacao_de_fada` conforme descritas no enunciado.

Para essa tarefa, você precisará de uma função auxiliar geradora de números pseudo-aleatórios¹. Ela deve ser implementada da seguinte forma: cada número gerado depende do número gerado anterior, seguindo a seguinte fórmula: $x_n = (7 * x_{n-1} + 111) \% 1024$ onde x_{n-1} é o último número gerado e x_n é o novo número a ser gerado. O número x_0 (chamado de semente) necessário para gerar o primeiro número pseudo-aleatório x_1 é dado na entrada. Como o jogo ainda está em fase de testes, a função também deve imprimir cada número gerado na seguinte mensagem: "MENSAGEM DEBUG - número gerado: x_n ". Note que essa função gera números entre 0 e 1023. Para obter um número inteiro em $[0, k)$ você deve pegar o resto da divisão do número gerado por k . Note também que, como a sequência de números gerados depende dos anteriores, você deve gerar exatamente a quantidade de números pedidos por cada função, e nenhum número a mais, para que a sequência seja conforme esperado.

A função `soco` deve gerar um único número pseudo-aleatório para gerar o parâmetro multiplicador m do dano base, sendo $0 \leq m \leq 2$. A função `arremesso_de_facas` também deve gerar um único número pseudo-aleatório para gerar a quantidade n de golpes, sendo $0 \leq n \leq 5$.

A função `invocacao_de_fada` deve gerar dois números pseudo-aleatórios, um para gerar a quantidade p de pontos de vida a serem ganhos ($0 \leq p \leq 99$), e outra para gerar um valor q ($0 \leq q \leq 1023$) que decidirá o efeito pseudo-aleatório secundário da ajuda das fadas. Se q for ímpar e menor que 100, o usuário da magia ganha q pontos de ataque a mais. Se q for par, menor que 100 e diferente de 0, o usuário ganha q pontos de defesa a mais. Se q for maior ou igual a 1019, a fada invoca o monstro que destrói o castelo e a batalha se encerra.

Após o retorno das funções `soco` ou `arremesso_de_facas`, seu programa deve imprimir a mensagem "[nome] sofreu [dano] pontos de dano!", onde [nome] é o nome da personagem oponente ("Sarah" ou "O clone"), e [dano] é a quantidade de dano causada. Após o retorno da função `invocacao_de_fada`, seu programa deve exibir a mensagem "[nome] ganhou [p] pontos de vida!", além da mensagem extra "[nome] ganhou [q] pontos de ataque!" ou "[nome] ganhou [q] pontos de defesa!" se o efeito pseudo-aleatório correspondente ocorreu. Se o efeito de invocação do monstro ocorreu no turno de Sarah, seu programa deve encerrar a execução após imprimir a seguinte mensagem:

¹ https://en.wikipedia.org/wiki/Pseudorandom_number_generator

“O quê? A fada trouxe um monstro gigante com ela!
O Clone e o castelo foram destruídos,
e Sarah agora tem um novo pet.
FINAL SECRETO - PARABENS???”

Se a invocação do monstro ocorre no turno do clone, seu programa também deve encerrar a execução e a mensagem é a seguinte:

“O quê? A fada trouxe um monstro gigante com ela!
Sarah foi derrotada...”

Você também deve implementar uma função que verifica se um personagem foi derrotado (pontos de vida reduzidos a zero), e chamar essa função no fim de cada turno em que um personagem sofre dano. Após o retorno da função, se Sarah foi derrotada, seu código deve imprimir a mensagem “Sarah foi derrotada...”, e se o clone foi derrotado, deve imprimir a mensagem:

“O clone foi derrotado! Sarah venceu!
FIM - PARABENS”

Em ambos esses casos, o programa deve então encerrar a execução após a mensagem.

(Nota: se quiser, seu código pode ter mais funções além das exigidas acima.)

Entrada

A primeira linha da entrada contém três números inteiros, descrevendo os pontos de vida, ataque e defesa de Sarah, respectivamente. A segunda linha descreve os atributos do clone, na mesma ordem. A terceira linha contém a semente a ser usada pela função geradora de números pseudo-aleatórios. As linhas seguintes contém uma única palavra, “soco”, “facas” ou “fada”, indicando qual habilidade foi usada a cada turno da batalha.

Saída

A saída do programa deve ser as mensagens após o retorno de cada função, além das mensagens contendo cada número pseudo-aleatório gerado, conforme descrito no enunciado. Atente-se ao enunciado e aos exemplos e siga exatamente a formatação dada.

Exemplos

Exemplo 1:

Entrada

```
41 56 25
48 86 25
29
facas
soco
soco
fada
```

Saída

```
MENSAGEM DEBUG - número gerado: 314
O clone sofreu 24 pontos de dano!
MENSAGEM DEBUG - número gerado: 261
Sarah sofreu 0 pontos de dano!
MENSAGEM DEBUG - número gerado: 914
O clone sofreu 62 pontos de dano!
O clone foi derrotado! Sarah venceu!
FIM - PARABENS
```

Exemplo 2:

Entrada

```
84 70 59
84 102 69
61
facas
facas
fada
soco
```

Saída

```
MENSAGEM DEBUG - número gerado: 538
O clone sofreu 32 pontos de dano!
MENSAGEM DEBUG - número gerado: 805
Sarah sofreu 34 pontos de dano!
MENSAGEM DEBUG - número gerado: 626
MENSAGEM DEBUG - número gerado: 397
Sarah ganhou 26 pontos de vida!
MENSAGEM DEBUG - número gerado: 842
Sarah sofreu 86 pontos de dano!
Sarah foi derrotada...
```

Exemplo 3:

Entrada

```
1000 100 100
5000 999 55
```

```
548
fada
facas
fada
soco
```

Saída

```
MENSAGEM DEBUG - número gerado: 875
MENSAGEM DEBUG - número gerado: 92
Sarah ganhou 75 pontos de vida!
Sarah ganhou 92 pontos de defesa!
MENSAGEM DEBUG - número gerado: 755
Sarah sofreu 497 pontos de dano!
MENSAGEM DEBUG - número gerado: 276
MENSAGEM DEBUG - número gerado: 1019
Sarah ganhou 76 pontos de vida!
O quê? A fada trouxe um monstro gigante com ela!
O Clone e o castelo foram destruídos,
e Sarah agora tem um novo pet.
FINAL SECRETO - PARABENS???
```

Submissão

Você deverá submeter no CodePost, na tarefa Lab 5, um arquivo com o nome `lab5.py`, contendo todo o seu programa.

Após o prazo estabelecido para a atividade, será aberta uma tarefa Lab 5 - Segunda Chance, com prazo de entrega até o fim do semestre.