Segundo Exercício-Programa: Las Begas

O dono de um cassino de "Las Begas", em "ABUSA", pretente fazer uma máquina jogadora de "Seven & Half" para jogar contra seus clients, digo, seus clientes. O dono do cassino, Mr. H. R. Whole, deseja que o jogador aposte contra a máquina (que faz as vezes da banca) como descreveremos abaixo.

Das regras do jogo

Daremos primeiramente a descrição do jogo como normalmente é jogado em ABUSA.

- Antes de qualquer sorteio, o apostador aposta x dólas (pronuncia-se dó-las) e a banca "banca" a aposta, ou seja, faz uma aposta de mesmo valor. Ao final do jogo, o vencedor leva todo o dinheiro: $2x$ dólas.
- Tanto a banca quanto o apostador são genericamente chamados de *jogador*.
- Dizemos que a *pontuação* de um jogador é o total dos valores das cartas que foram sorteadas para este jogador. Vence aquele jogador cuja pontuação for maior. Em caso de empate, vence aquele que fez a pontuação com o menor número de cartas. Caso ainda haja empate a banca embolsa o dinheiro.
- A banca administra o baralho e sorteia, suponha que honestamente, as cartas. Primeiro, tantas cartas quanto o apostador quiser; em seguida, tantas cartas quanto a banca desejar para si.
- A cada novo sorteio, a carta sorteada é virada sobre a mesa de modo que o jogador e a banca a vejam.
- O baralho possui quarenta cartas — é um baralho comum em que foram retiradas as cartas oito, nove e dez de qualquer dos quatro náipes usuais[1].
- O ás vale 1, as figuras (rei, dama e valete) valem $\frac{1}{2}$, as demais cartas valem o número correspondente.
- Caso a pontuação de um jogador supere $7\frac{1}{2}$, situação em que se diz que o jogador *estourou*, o jogador perde automaticamente.
- Após um jogo, o vencedor leva os $2x$ dólas apostados e os jogadores podem reiniciar outro jogo caso o apostador assim o deseje.

Veremos agora um exemplo de quatro jogos:

jogo	jogador	cartas	pontuação	situação final
jogo 1	apostador	ás, 3, dama, rei, 2	$1+3+\frac{1}{2}+\frac{1}{2}+2=7$	apostador vence
	banca	4, 2, rei, 5	$4+2+\frac{1}{2}+5=11\frac{1}{2}$	
jogo 2	apostador	ás, 3, 6	$1+3+6=10$	banca vence
	banca			
jogo 3	apostador	ás, rei, 6	$1+\frac{1}{2}+6=7\frac{1}{2}$	banca vence
	banca	7, valete	$7+\frac{1}{2}$	
jogo 4	apostador	7	7	apostador vence
	banca	7, ás	$7+1=8$	

[1]Dizem que o fabricante de baralhos já fabrica o baralho só com as 40 cartas para não ter que jogar fora as 12 restantes, contribuindo assim com a preservação das florestas de onde ecologicamente se extrai a celulose necessária.

Das estratégias de cada jogador

Cada jogador possui um objetivo e uma estratégia. O objetivo de cada jogador é vencer. Quanto às estratégias não são muito mais complicadas.

A estratégia do apostador é pedir que a banca sorteie uma nova carta para ele enquanto achar necessário. O apostador sabe que precisa ter a maior pontuação possível, sem no entanto estourar. Assim, adotaremos uma estratégia simples: adotaremos um `teto` para o apostador. Se sua pontação corrente for menor que o `teto`, ele pede mais uma carta; caso contrário, ele diz à banca que não quer mais nenhuma carta e, caso o jogador não tenha estourado, ela passa a sortear cartas para si própria.

Quanto à estratégia da banca, é mais simples ainda. Se o apostador não tiver estourado, pois caso contrário a banca já terá ganho, ela vai sorteando cartas para si enquanto a pontuação obtida não lhe garantir a vitória sobre o apostador e houver ainda alguma chance de obter uma pontuação vencedora com um novo sorteio. Ao final, a banca terá feito uma pontuação que lhe garanta a vitória sobre o apostador, ou terá estourado.

Da simulação

O dono do cassino, Mr. Whole, decidiu contratar vocês para construir um programa que simule o jogo de suas máquinas de "Seven n Half". Como vimos antes, o apostador joga contra a máquina que por sua vez faz o papel da banca.

Por simplicidade[2], após sortear uma carta qualquer, o jogador em questão contabiliza os pontos da carta que foi sorteada para si e *a carta é devolvida ao baralho*. O mesmo é então honestamente embaralhado antes de um novo sorteio de uma carta, caso seja necessário. Assim o sorteio de uma segunda carta é completamente independente da carta sorteada na vez passada, digo, na última vez. Pode inclusive repetir-se a mesma carta.

Primeiro, o programa deve inicializar uma variável de nome `passado` com o valor real:

```
passado = 0.<seu número USP>             /* exemplo:  passado = 0.5262199 */
```

Depois deve ser feito o seguinte cálculo:

```
sorteio = ( 9821.0 * fabs( seno(passado) ) + 0.211327 ) ...................... (1)
passado = sorteio - floor( sorteio ) .......................................... (2)
```

onde `fabs(x)` é uma função que devolve o módulo de `x`,

`floor(x)` é uma função que devolve o maior inteiro não maior que `x`,

seno`(x)` denota o valor calculado para o seno de `x` **USANDO** o método descrito ao final deste exercício-programa.

As fórmulas em (1) e (2) (nesta ordem) nos fornecem um número no intervalo $[0, 1[$. Para obtermos um número inteiro entre 1 e 10 basta fazer a seguinte conta:

```
carta = floor( passado*10 + 1 ); .............................................. (3)
```

Isto fornece-nos uma carta, sendo que 8, 9 e 10 representam respectivamente uma *dama*, *valete* e *rei*. Observe que os náipes não interessam. Sempre que for desejado um sorteio de uma carta, o programa **DEVE** fazer as operações descritas nos passos (1), (2) e (3).

[2]Algumas línguas dizem que foi preguiça do programador.

Observe que durante as repetições da (1) e (2) o valor da variável `passado` **vai sendo alterado** e portanto os valores sorteados para as cartas vão se alterando.

Mr. Hole, digo Mr. Whole, deseja saber se a sua máquina de Seven & Half alferirá bons lucros, qualquer que seja a estratégia adotada pelo apostador. Por isto, o dono do casino[3] contratou o supra-sumo das turmas supra-citadas para fazer um programa em C que simule os jogos de suas máquinas. Seu programa deve testar as estratégias do apostador e da banca **CONFORME** descrito anteriormente, para todos os valores possíveis que `teto` possa assumir. Para cada valor de `teto`, de 0 a $7\frac{1}{2}$, o programa deve simular 100 jogos e computar em quantos jogos o apostador venceu. Chamemos de `derrotas` o número de vezes que o apostador venceu (portanto o número de vezes que a máquina de Mr. Hole Rule Whole perdeu). Para cada um destes testes com `teto`, o programa deve imprimir uma linha dizendo qual valor de `teto` está sendo considerado, quantas vezes o apostador venceu (o valor de `derrotas`) e em seguida tantos caracteres `'*'` quanto for o valor de `derrotas`. Se para `teto=4.5` e `teto=5` os valores encontrados de `derrotas` forem respectivamente 20 e 29, deverão ser impressas linhas como as abaixo:

```
4.5  20 ********************
5.0  29 *****************************
```

IMPORTANTE: Todo exercício-programa deve seguir as observações contidas nos folhetos distribuídos a parte. Nestes folhetos estão descritas as diretrizes para a forma de entrega do exercício, aspectos importantes na avaliação, etc. Para calcular os valores do *seno*(`x`) **USE OBRIGATORIAMENTE** a seguinte aproximação:

$$\text{seno}(x) = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots + \frac{(-1)^k x^{2k+1}}{(2k+1)!} + \cdots$$

incluindo na soma todos os termos enquanto

$$\frac{|x^{2k+1}|}{(2k+1)!} \geq 10^{-8}.$$

[3] A mãe de Mr. Rule é italiana e ele viveu um tempo em Nápoli.