Departamento de Ciência da Computação – IME-USP

MAC 115 – Introdução à Computação para Ciências Exatas e Tecnologia

Escola Politécnica – Primeiro Semestre de 1998

Segundo Exercício-Programa | Data de entrega: até **22/04/98.**

Turmas: **Civil C, Elétrica C-D, Mecânica**

URKASKY 2

Uma máquina de apostas "caça-níqueis" funciona da seguinte forma: a máquina tem três rolos (R_1, R_2 e R_3), sendo que cada rolo tem cinco figuras (f_1, f_2, f_3, f_4 e f_5). Cada vez que a máquina é acionada, os rolos giram até parar em uma posição aleatória, apresentando uma das figuras.

1. quando dois rolos apresentam figuras iguais e apenas um apresenta figura diferente, a máquina premia o apostador com uma quantia X;
2. quando os três rolos apresentam figuras iguais mas essas figuras não são a figura f_1, a máquina premia com uma quantia $2 \times X$;
3. quando os três rolos apresentam a figura f_1, o prêmio é $4 \times X$;
4. nos casos restantes, o apostador fica a ver navios.

Construa um programa para simular o funcionamento dessa máquina. Para tanto, voce precisará de um gerador de números aleatórios entre 1 e 5. Utilize o seguinte trecho de programa para essa tarefa:

Primeiro, o programa deve inicializar uma variável y com um valor real, por exemplo

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

Depois deve ser feito o seguinte cálculo:

`x = (9821.0 * fabs (`*seno* (y)`) + 0.211327)` .... (1)

`y = x - floor (x)` .... (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 nas observações que acompanham este exercício-programa.

As fórmulas em (1) e (2) (nesta ordem) fornecem um número no intervalo $[0, 1[$. Para obter um número inteiro no intervalo [1,5] basta fazer o seguinte cálculo:

`figura = (y*5) + 1;` .... (3)

Observe que a variável "figura" fornece uma figura sorteada. Para obtermos novas figuras, os cálculos (1), (2) e (3) devem ser repetidos considerando o valor mais recente de "y". Como o valor de "y" é alterado a cada novo cálculo, obtemos sempre uma nova figura como resultado.

Se a distribuição das figuras produzidas dessa maneira for uniforme, as seguintes probabilidades devem ser obtidas (experimente deduzir para saber se isso está correto):

1. prob$(X) = \frac{14}{25}$
2. prob$(2 \times X) = \frac{4}{125}$
3. prob$(4 \times X) = \frac{1}{125}$
4. prob(ver navios) $= \frac{2}{5}$

Seu programa deve permitir escolher dentre quatro alternativas de uso:

1. apostar apenas em f_1: qualquer outro prêmio será desconsiderado;
2. apostar somente em obter três figuras iguais: prêmio correspondente a apenas duas figuras iguais será desconsiderado;
3. apostar em qualquer prêmio: o que vier será lucro;
4. modo de aferição: nesse modo o programa deverá fazer o seguinte:
   (a) Ler e imprimir um número inteiro $N \geq 1$;
   (b) Apostar N vezes e contar quantas vezes cada prêmio é gerado;
   (c) Imprimir um histograma na posição horizontal (como mostrado no exemplo abaixo) com 180 asteriscos (*), onde o número de asteriscos na linha com o **Prêmio** i é proporcional ao número de ocorrências do prêmio i durante as N apostas;
   (d) Imprimir a diferença entre a quantidade de prêmios gerados e a quantidade teórica que deveria ter sido obtida conforme acima, para cada prêmio.

Exemplo:

Valor de N: 60				
Prêmio	1	2	3	4
Ocorrência	15	14	12	19

Histograma Correspondente:

```
Prêmio
  1      *********************************************
  2      ******************************************
  3      ************************************
  4      *********************************************************
```

esperado para X: 33.6 obtido para X: 15 diferenca: -18.6

esperado para 2X: 1.9 obtido para 2X: 14 diferenca: 12.1

esperado para 4X: 0.5 obtido para 4X: 12 diferenca: 4.3

esperado para 0: 24.0 obtido para 0: 19 diferenca: -5.0

IMPORTANTE: Todo exercício-programa deve seguir as observações contidas nos folhetos distribuídos a parte. Nestes folhetos estão descritas as diretrizes para forma de entrega do exercício, aspectos importantes na avaliação, etc.

OBSERVAÇÕES SOBRE O EXERCÍCIO-PROGRAMA URKASKY

1. Seu programa deve permitir ao jogador continuar jogando, se assim o desejar. Para isso, seu programa pode ter a seguinte estrutura:

```
#include <stdio.h>
#include <math.h> /* Nao deixe de colocar esta linha! */

main() {
  char resposta;

  [...]

  do {

    <corpo do programa>;

    printf ("Deseja jogar novamente? (S/N) ");
    resposta = getc(stdin); /* leitura do teclado, nao precisa bater ENTER */
  } while ((resposta != 'N') && (resposta != 'n'));

  [...]

} /* fim do programa */
```

2. 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 saída do programa não precisa ser exatamente como a descrita no enunciado do exercício-programa. Você pode deixar a saída mais bonita, se o desejar, colocando, por exemplo, o histograma na vertical (não é complicado, pense a respeito).