

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO - DEPARTAMENTO DE INFORMÁTICA
2º Trabalho Computacional de PD II e Prog II - EC/CC
Data da entrega: 28/06/2009

A empresa **LITTLE-GAMES** deseja criar um jogo de computador onde o usuário poderá escolher entre quatro tipos de jogos:

- a) O famoso **JOGO DA MEMÓRIA**
- b) Escolher uma figura e gerar aleatoriamente uma figura o mais semelhante possível da original - **UMA GERAÇÃO DE SORTE**
- c) Gerar aleatoriamente duas figuras o mais próximas uma da outra - **DUAS GERAÇÕES DE SORTE**
- d) A famosa **FORCA**

Seguem as descrições de cada um dos jogos:

JOGO DA MEMÓRIA

Neste jogo serão fornecidos dois grupos de arquivos. O grupo **A** possui cinco arquivos denominados **a1.txt**, **a2.txt**, **a3.txt**, **a4.txt** e **a5.txt** e o grupo **B** possui os mesmos arquivos porém enumerados em ordem distintas, denominados **b1.txt**, **b2.txt**, **b3.txt**, **b4.txt** e **b5.txt**. Tais arquivos armazenam matrizes 10x10 com elementos 0 e 1 que formam algumas figuras geométricas. Exemplo de um triângulo:

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0
0 0 0 0 1 1 1 0 0 0
0 0 0 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

Quando o jogo iniciar, todos os arquivos do grupo B são exibidos para o jogador. Depois de ver (observar bem) a ordem das figuras, o jogador deve escolher um arquivo do grupo A (que será exibido para o jogador) e em seguida, escolher um arquivo do grupo B com a intenção de casar as figuras iguais. Esse procedimento se repete cinco vezes ou até o jogador errar o primeiro casamento. Neste jogo pode participar apenas um jogador. Caso haja mais participantes, vence aquele que acertar todos os casamentos na mesma rodada. Se dois ou mais jogadores acertarem na mesma rodada é declarado empate. Imprimir na tela o(s) nome(s) do(s) vencedor(es).

UMA GERAÇÃO DE SORTE

Neste jogo será considerado o grupo A de arquivos do jogo acima. O jogador deve escolher um arquivo deste grupo (figura) e o computador deve gerar aleatoriamente uma nova figura (equivale a gerar uma matriz de 0 e 1 aleatoriamente). Vence o jogador que tiver a sorte do computador gerar a figura mais semelhante. Se dois ou mais jogadores tiverem figuras com a mesma semelhança é declarado empate. Imprimir na tela o(s) nome(s) do(s) vencedor(es). Considere **figuras semelhantes** aquelas que possuem mais de 50% de elementos iguais na figura. A função de geração de números aleatórios é fornecida no final deste documento.

DUAS GERAÇÕES DE SORTE

Neste jogo serão geradas aleatoriamente duas figuras para cada jogador e aquele que tiver a sorte das figuras serem mais semelhantes é declarado o vencedor deste jogo. Se dois ou mais jogadores tiverem figuras com a mesma semelhança é declarado empate. Imprimir na tela o(s) nome(s) do(s) vencedor(es). Utilizar a mesma função de geração de números aleatórios.

FORCA

Existem 10 palavras armazenadas em um arquivo denominado **WORDS.TXT**. O jogo deve acessar esse arquivo e armazenar as palavras em um vetor de *strings*. O jogador deve escolher um número entre 0 e 9 (refere-se a alguma palavra) e tem no máximo 10 tentativas para escolher letras e completar a palavra. O jogo deve informar para o jogador o número de letras da palavra escolhida e exibir as partes da palavra que está sendo criada para auxiliar o jogador a descobrir a palavra. Neste jogo pode participar apenas um jogador. Caso haja mais participantes, vence aquele que acertar a palavra escolhida com menor número de tentativas na mesma rodada. Os jogadores, na mesma rodada, podem escolher palavras diferentes. Se dois ou mais jogadores acertarem a palavra escolhida com mesmo número de tentativas na mesma rodada é declarado empate. Imprimir na tela o(s) nome(s) do(s) vencedor(es).

Uma dica: inicializar o *string* que armazena a palavra que está sendo criada com *underscore*. Exemplo,

```
char nova[15];  
  
nova="_____";
```

Organização do programa: o seu programa deverá ler o tipo de jogo que se deseja brincar (um dos quatro jogos apresentados acima), o número de jogadores e o nome de cada jogador. Deverá oferecer a oportunidade dos mesmos jogadores jogarem novamente. A saída do programa depende do jogo foi escolhido. Cada um foi descrito detalhadamente acima.

Data de Entrega: O trabalho deverá ser entregue até as 23h59min do dia 28/06/2008.

Forma de Entrega: O assunto do e-mail deverá ser (somente o que está entre aspas duplas): "\pd2prog2:trab2:nome:". Substitua nome pelo seu primeiro nome e último sobrenome, separados por espaços. O nome do arquivo enviado deverá ser os primeiros nomes dos componentes do grupo. NÃO escreva o seu nome com caracteres estendidos (ã, ç, etc). Veja abaixo um exemplo de um e-mail de envio do trabalho de João da Silva e José Carlos de Abreu: (apenas 1 do grupo deve mandar o trabalho)

Para: crangel@inf.ufes.br

De: Joao da Silva

Assunto: \pd2prog2:trab2:Joao Silva:Jose Abreu:

Anexo: joao-jose.c

Observações importantes:

- 1) Enviar o trabalho estritamente para o e-mail especificado acima. Não serão aceitos trabalhos enviados fora do prazo.
- 2) Em caso de detecção de cópia (parcial ou integral), todos os envolvidos recebem nota ZERO.
- 3) Trabalho que não compila recebe nota ZERO. Não adianta nem submete.
- 4) Os programas serão avaliados pela sua correção durante a execução e pelo estilo de programação. Serão observados se os programas possuem comentários apropriados, nomes significativos para as variáveis, se o código está corretamente indentado, etc.

A função aleatória: a função *rand()* gera um número inteiro maior ou igual a zero. Para gerar números menores que M, toma-se *rand()%M*. Deve-se iniciar o processo de geração aleatória com a função *srand(semente)*, executada uma única vez. Esta semente é qualquer valor numérico. Neste

trabalho deve ser utilizada a semente do “*clock* do relógio do computador” para que seja tomada uma semente diferente a cada execução do programa. Abaixo, um programa de geração de número entre 0 e 60 (incluse os limites).

```
#include<stdio.h>
#include<stdlib.h> /*rand() e srand() estão em stdlib.h*/
main()
{
    float semente;
    int seg;
    semente = time(&seg)%100;
    srand(semente); /*inicializar o gerador de números
aleatórios com a semente*/
    printf("semente do clock do relógio = %f \n",semente);
    printf("numero aleatorio entre 0 e 60 = %d \n",rand()%61);
}
```

Conselhos da professora que se preocupa com os alunos

- 1) Não deixem para fazer o trabalho em cima da hora.
- 2) Façam testes prevendo situações adversas para verificar se o programa está funcionando direitinho.
- 3) Certifiquem-se de que a versão enviada é a final.
- 4) A idéia de se trabalhar em dupla é para haver comunicação entre as partes. Deve haver reunião presencial.