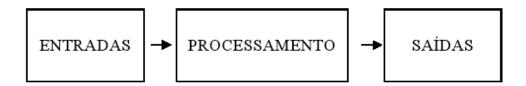
INF 01202 – ALGORITMOS E PROGRAMAÇÃO

TRABALHO PRÁTICO

1.Objetivo:

Exercitar as habilidades e conceitos de programação desenvolvidos ao longo da disciplina pela implementação de uma aplicação em C, desenvolvida por um grupo de 2 alunos. A coordenação de um trabalho em equipe faz parte da habilidade de programar.

O programa deve ser estruturado de forma a receber um conjunto de entradas (simultaneamente ou durante o uso do programa), cuja consistência deve ser verificada, processá-las, e fornecer uma ou mais saídas.



2. Produto do trabalho e datas de entrega

O trabalho será entregue em 3 etapas:

a) Relatório de Andamento: dias 09 de Junho

Apresentação na aula prática do código do trabalho onde será formalizada a dupla, e mostrada a tela de movimentação do rato funcionando e as estruturas de dados necessárias.

b) Entrega do código 06 de Julho (Quarta-feira) as 23:55 horas pelo Moodle

Upload no Moodle em tarefa própria de um ÚNICO arquivo compactado cujo nome do arquivo é o nome dos alunos contendo:

- Programa executável: o programa deve rodar nas máquinas do laboratório das aulas práticas. Verifique se executará sem exceções naquele local antes de entregar.
- Código documentado (**arquivos .c, .cpp**). Inclua o nome dos autores no cabeçalho do programa.

Se houver exceções aos requisitos listados neste enunciado, descreva como comentário no início do arquivo fonte.

Você pode fazer upload de diferentes versões e ir aperfeiçoando o programa. Faça o upload assim que tiver uma versão executável, de modo a garantir a entrega e precaver-se de problemas com servidor, redes, internet, etc.

c) Apresentação

Programa será apresentado no dia **7 Julho** na aula prática. O arquivo a ser apresentado será aquele carregado no Moodle. Nenhuma alteração será permitida. Ambos alunos devem dominar o código para explicá-lo.

3. Avaliação

O programa deve atender todos os requisitos listados neste enunciado, não deve apresentar erros de compilação e rodar normalmente, pontos serão deduzidos caso contrário.

A aplicação desenvolvida deverá demonstrar os seguintes conteúdos que serão avaliados:

- 1. (2 pontos) Habilidade em estruturar programas pela decomposição da tarefa em subtarefas, utilizando subprogramação para implementá-las.
- 2. (2 pontos) Documentação de programas (identação, utilização de nomes de variáveis, abstração dos procedimentos para obter maior clareza, uso de comentários no código).
- 3. (2 ponto) Domínio na utilização de tipos de dados simples e estruturados (arranjos, conjuntos, estruturas) e passagem de parâmetros.
- 4. (1 ponto) Formatação e controle de entrada e saída, com construção de interfaces que orientem corretamente o usuário sem instruções ou intervenção adicional do programador.
- 5. (1 ponto) Utilização de arquivos binários e de texto.
- 6. (2 pontos) Atendimento aos requisitos do enunciado do programa: modelo de estrutura de dados, de interação e de relatórios, opções do programa e salvamento intermediário de estado do sistema.

A nota do trabalho prático corresponderá a 10% da nota final. Para alcançar o conceito A na disciplina o aluno deve apresentar o trabalho final.

4. Contextualização:

OBJETIVO DO PROGRAMA

Deverá ser feito o jogo *RATO-RATOEIRA*, em modo texto, onde o usuário do programa (jogador) controlará a movimentação do rato que se move comendo alimentos e fugindo das ratoeiras que surgem a cada alimento comido. O objetivo do jogo é comer o maior número de queijos sem permitir que o rato seja pego na ratoeira. Cada queijo tem uma pontuação e, ao ser comido, aparece o número de ratoeiras correspondentes à pontuação do queijo em posições aleatórias no espaço de jogo. Se o rato encostar em uma ratoeira, o jogo se encerra.

O PROGRAMA

As duplas terão liberdade para a implementação do programa. Entretanto, os seguintes requisitos mínimos devem ser respeitados:

- a) os dados de configurações iniciais do jogo deverão ser carregados pelo usuário em um *struct*, que contem:
 - posição X inicial do rato;
 - posição Y inicial do rato;
 - direção inicial do movimento do rato (cima(0), baixo(1), esquerda(2), direita(3));
 - nome do jogador;
 - tempo inicial do jogo inicializado pelo programa com o tempo marcado pelo relógio do computador (biblioteca <time.h>).
- b) O rato será representada por 4 caracteres, as comidas pelo número correspondente as pontos adquiridos, que corresponde também ao número de ratoeiras que aparecerão. As paredes serão representadas pelo caractere "bloco" (código ASCII 219), e espaços em branco (caminháveis) pelo caractere "(espaço em branco);
- c) O espaço deverá ter um formato retangular, com dimensões máximas de 70 caracteres de largura e 24 caracteres de altura, incluindo as paredes;
- d) as seguintes teclas deverão ser utilizadas para controlar a interação no jogo:
 - q encerra o jogo
 - ↑ muda a direção de movimentação do rato para cima
 - ↓ muda a direção de movimentação do rato para baixo
 - ← muda a direção de movimentação do rato para a esquerda
 - → muda a direção de movimentação do rato para a direita

barra de espaço para pausa (o tempo do jogo deve descontar as pausas).

- e) o rato deve se movimentar sozinho em velocidade constante na direção inicialmente definida, até que o jogador mude a direção;
- f) o rato não pode atravessar as paredes (deve haver uma detecção de colisão);
- g) o queijo desaparece quando o rato encosta no queijo, a pontuação relativa é adicionada, as ratoeiras e uma nova comida aparecem;
- h) a pontuação do jogador é calculada dividindo a pontuação dos queijos pelo tempo decorrido desde o início do jogo;
- i) na área à direita da tela deve ser mostrada as seguintes informações: nível em que o jogo está, pontuação de queijos consumidos, nome do jogador e pontuação do jogador;
- j) cada jogo é composto de 3 níveis com dificuldades crescentes (a velocidade pode

aumentar ou os valores das comidas podem ser maiores a cada nível). A passagem de um nível para outro se dá ao completar com sucesso um determinado nível;

k) as posições e valores das comidas nos 3 níveis são determinados por 3 arquivos 'nivel1.txt', 'nivel2.txt' e 'nivel3.txt', que serão carregados para um arranjo. Lembrar que as posições e valores das comidas devem ser armazenadas, mas somente **uma** comida é exibida por vez.

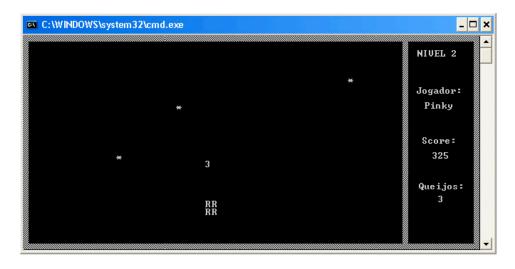
Exemplo de arquivo "nívelX.txt":

1	3	1	2	2 3	
3	2	1	4		
6	2	5	3	1	

l) o programa deve identificar quando um nível se encerra (todas as comidas acabam) e quando o jogo se encerra (todos os níveis são vencidos ou o rato cai na ratoeira)

m) Em qualquer ponto do jogo, o jogador pode pausar e salvar o estado corrente do jogo. Todos os dados do jogo serão salvos em um arquivo binário cujo nome é o do jogador. Ao iniciar o jogo, deve pedir o nome do jogador e mostrar as opções "novo jogo" ou "carregar jogo".

Exemplo de tela do programa:



Dica:

• a biblioteca conio2 contém varias funções úteis, como putchxy, kbhit e getch. Para utilizála no CodeBlocks, baixe o arquivo ZIP no Moodle (conio2.zip) e sigam os passos do arquivo LEIAME.txt contido dentro dele.