



UNIVERSIDADE FEDERAL DE OURO PRETO

DEPARTAMENTO DE CIÊNCIAS EXATAS E APLICADAS

PROGRAMAÇÃO DE COMPUTADORES I – CEA030

ENUNCIADO DO TRABALHO PRÁTICO

1º Semestre/2010

Profa. Geiza Cristina da Silva

- Objetivos:
  - O objetivo desse trabalho é rever conceitos básicos de programação para a resolução de problemas comuns no âmbito da Computação.
- Critérios de avaliação:
  - A eficiência da implementação será levada em conta.
  - Clareza, indentação e comentários no programa também vão valer pontos.
  - A avaliação do trabalho será comparativa. Isto é, quanto melhor a solução adotada maior deverá ser a pontuação dada.
- Regras para entrega:
  - O trabalho deve ser realizado em duplas.
  - O trabalho deve ser enviado em anexo para o email: [geiza@decea.ufop.br](mailto:geiza@decea.ufop.br).
  - O email deve ter como assunto: Trabalho Prático CEA 030.
  - O email deve ter como conteúdo: nome e matrícula de cada um dos alunos da dupla.
  - O email deve ter como anexo um arquivo compactado contendo SOMENTE o arquivo de código fonte do programa (.c). O arquivo compactado deve ter a denominação XY.zip onde X e Y correspondem ao nome dos alunos da dupla.
  - A entrega deverá ser feita até às 23h do dia 01/06/2010 e atrasos não são tolerados. Além disso, o programa deverá ser apresentado no dia 02/07. Se não for possível a apresentação de todas as duplas, o restante dos trabalhos serão apresentados no dia 03/07.
  - Programas que apresentarem erro na compilação receberão nota zero. O programa não pode ser escrito usando a sintaxe de C++, próprias de determinados compiladores ou funções que não são do padrão de C. Programas que compilarem serão executados e avaliados numa sequência de testes.
  - Similaridades entre trabalhos poderão implicar em redução parcial (ou em totalidade) da pontuação máxima.
  - Os emails de envio de trabalho não serão respondidos.

**Atenção: A não observância das regras pode invalidar o trabalho.**

### Enunciado: Jogo Liga 4

O jogo "Liga 4" é um jogo para dois jogadores, formado por um tabuleiro de N linhas por M colunas. Cada jogador é representado por um determinado caractere ('O' para o jogador 1, e 'X' para o jogador 2). O objetivo do jogo é que cada jogador consiga arranjar seus caracteres em uma sequência de 4. Linhas horizontais, verticais ou diagonais são válidas. O jogador pode apenas escolher uma coluna para posicionar seu caractere sendo ela então colocada na sua primeira posição livre (de baixo para cima). Ou seja, caso exista um ou mais caracteres na coluna em que ele escolher, o caractere só poderá ser posto na posição livre subsequente. Se não houver espaço nesta coluna, ela não pode ser escolhida pelo jogador. As jogadas são feitas de maneira alternada entre o jogador um e o jogador dois. Ao mesmo tempo em que um jogador tenta alcançar seu objetivo, ele pode atrapalhar as jogadas do seu oponente. Vence aquele que primeiro alcançar o objetivo de colocar seus 4 caracteres em sequência.

### Implementação:

Assumiremos que o jogador 1 começa a partida. O jogo consiste basicamente nos seguintes passos:

- Mostrar uma representação do tabuleiro na tela;
- Pedir ao jogador 1 a coluna em que ele quer jogar sua peça. Se o jogador escolher uma coluna inválida (cheia ou fora das dimensões do tabuleiro) o programa deve emitir uma mensagem de erro correspondente e pedir novamente a jogada até que ela seja válida;
- Repetir o processo para o segundo jogador e assim sucessivamente.
- O programa deve ser encerrado ao encontrar um vencedor ou até que não existam mais posições livres no tabuleiro, indicando um empate.
- O trabalho deverá ser implementado utilizando necessariamente um conjunto de funções especificadas a seguir. A saída gerada pelo programa, assim como sua entrada de dados, devem também obedecer rigorosamente o padrão estabelecido nesta especificação.

A saída do programa consiste de:

- Representação do tabuleiro;
- Mensagem indicando a vez de qual jogador (1 ou 2);
- Mensagem indicando se uma jogada é inválida;
- Mensagem indicando um vencedor.

A impressão de um tabuleiro vazio deverá ser representada da seguinte maneira:

```
1 2 3 4 5 6 7
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```

Um tabuleiro preenchido após quatro rodadas poderia ter o seguinte *layout* visual:

```
1 2 3 4 5 6 7
. . . . .
. . . . .
. . . . .
. . . . .
. . X . .
. . 0 X 0 .
```

A cada rodada, o programa deve indicar de qual jogador é a vez, para que ele forneça o valor (inteiro) de uma coluna. A seguinte mensagem deverá ser gerada como saída, caso seja a vez do jogador 1:

1>

Ou então, caso seja a vez do jogador 2:

2>

No caso do jogador fornecer uma coluna inválida (fora da faixa válida) ou uma coluna já totalmente preenchida, o programa deve mostrar a seguinte mensagem:

Jogada inválida.

Lembrando que a única entrada do programa é a coluna escolhida pelo jogador a cada rodada. Ao encontrar um vencedor, o programa deve mostrar a seguinte mensagem antes de ser encerrado, caso o jogador 1 tenha ganho a partida:

Fim! 1

Ou então, caso o jogador 2 tenha ganhado:

Fim! 2

Ou ainda, no caso que não haja vencedor:

Empate!

### Especificação das Funções:

O trabalho terá que obrigatoriamente utilizar funções na solução do problema. Todas as funções devem trabalhar com um tabuleiro cujas dimensões são definidas pelas seguintes constantes, sendo N o número de linhas e M o de colunas:

```
#define N 6  
#define M 7
```

As seguintes funções, além da main, devem ser implementadas:

Protótipo	Funcionalidade
<code>void escreveTabuleiro(char tab[N][M]);</code>	Mostra o tabuleiro <i>tab</i> na tela.
<code>int ganhou(char tab[N][M], int jogador);</code>	Verifica se o <i>jogador</i> no tabuleiro <i>tab</i> ganhou. Devolve 1 se o jogador ganhou, devolve 0 caso contrário.
<code>int estaCheio(char tab[N][M]);</code>	Verifica se o tabuleiro <i>tab</i> esta cheio. Devolve 1 se verdadeiro ou 0 caso contrário.
<code>int leJogada(int jogador);</code>	Recebe como parâmetro <i>jogador</i> que pode ser 1 ou 2, lê a jogada e devolve a coluna escolhida.
<code>void executaJogada(char tab[N][M], int jogador, int coluna);</code>	Executa a jogada indicada por <i>coluna</i> do jogador <i>jogador</i> no tabuleiro <i>tab</i> . (Preenche a posição correspondente da matriz).
<code>int jogadaValida(char tab[N][M], int coluna);</code>	A partir de <i>coluna</i> escolhida pelo jogador, devolve 1 caso a jogada seja válida no tabuleiro <i>tab</i> ou devolve 0 caso seja inválida.