# Programação I

1º Semestre
2018/2019

# 2º Trabalho
# Data de Entrega: 25 de Novembro de 2018

Neste trabalho, os alunos vão adquirir a prática de utilização de instruções de ciclo, vectores(*arrays*) e funções. Pretende-se que os alunos desenvolvam uma abordagem estruturada (*top-down*) na construção de programas, através da especificação de funções, e organizando-os em módulos (múltiplos ficheiros fonte) quando necessário.

Cada grupo entregará no *site* da sua turma os ficheiros fonte das respetivas soluções, devidamente comentados e comprimidos num ficheiro (com extensão `zip`, `rar` ou `gz`).

## Recomendações

É valorizada a simplicidade das soluções que, no entanto, devem cumprir os requisitos do enunciado. A compilação dos programas não deve gerar *warnings*, os nomes das variáveis devem descrever o seu propósito e todos os valores lidos do teclado devem ser validados. A utilização de "números mágicos" no código devem ser evitados e recomenda-se a utilização de constantes através da definição de `macros`. O aluno deve citar os *sites* que consultou aquando da elaboração dos programas que implementou na realização do trabalho.

## Questões

**1.** Realize um programa para jogar o "jogo do 31" entre o computador e um jogador humano. Cada jogador joga alternadamente um valor inteiro de entre três possíveis (1, 2 ou 3) que vai acumulando com o total das jogadas anteriores. Ganha o primeiro jogador que primeiro atingir (ou ultrapassar) o valor 31.

Deve ser escolhido aleatoriamente quem joga da primeira vez. O jogo tem uma estratégia vencedora que deve ser seguida na programação da jogada do computador, mas no caso de ser o computador o primeiro a jogar a sua jogada deve ser aleatória.

Durante uma sessão de utilização do programa podem ser realizados vários jogos, jogando em primeiro lugar o vencedor do jogo anterior.

Ao lado exemplifica-se o funcionamento do jogo num cenário em que o computador é o primeiro a jogar, para servir de referência na sua implementação.

```
isel@isel-virtual-machine: ~/disciplinas/pg1/i2017-2018/trabs/trab2
File  Edit  Tabs  Help
isel@isel-virtual-machine:~/disciplinas/pg1/i2017-2018/trabs/trab2$ jogo31
Primeiro eu
eu jogo 3, vai em 3
sua jogada: 2
eu jogo 2, vai em 7
sua jogada: 3
eu jogo 1, vai em 11
sua jogada: 3
eu jogo 1, vai em 15
sua jogada: 1
eu jogo 3, vai em 19
sua jogada: 1
eu jogo 3, vai em 23
sua jogada: 2
eu jogo 2, vai em 27
sua jogada: 1
eu jogo 3, vai em 31
desta vez ganhei eu. Boa sorte para a próxima
novo jogo (s/n)? n
adeus, até à próxima!
isel@isel-virtual-machine:~/disciplinas/pg1/i2017-2018/trabs/trab2$
```

Fig.1 Output gerado na execução do jogo do 31

Nota: na geração de valores (pseudo) aleatórios utilize as funções `srand(int)` para especificar a semente do gerador e `int rand()` para obter sucessivos valores aleatórios.

**2.** De modo a representar e efetuar operações com conjuntos de inteiros (até 256 elementos) foi definido o tipo SET apresentado abaixo.

```
#define MAX_SET 256             /// capacidade (máxima cardinalidade) do conjunto
typedef int Set[MAX_SET+1];     /// array que contem dimensão e elementos de conjunto
```

O tipo de dados **`Set`** é representado como indicado na fig. 2: O valor presente no índice 0 indica o número de elementos do conjunto e a partir do índice 1 encontram-se os elementos contidos nesse conjunto.

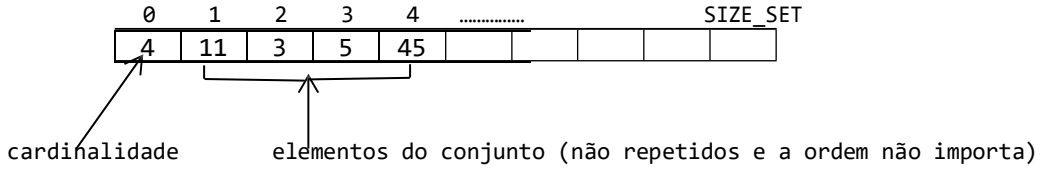



Fig.2 – Organização de um conjunto SET: exemplo para o conjunto `{11, 3, 5, 45}`

Em anexo são disponibilizados ficheiros com os protótipos das funções a realizar (`set.h`), o ficheiro com as definições das funções (`set.c`, a completar) e um ficheiro com uma série de testes (`set_testes.c`) para verificar a correção das vossas soluções. Notem que é legítimo adicionar mais funções ao ficheiro `set.c` caso o considerem útil.

**3.** Pretende-se realizar uma versão simplificada do jogo do gamão para dois jogadores humanos.
Cada jogador tem 15 peças (brancas ou vermelhas) distribuídas inicialmente num tabuleiro de 24 posições (casas). A Fig. 1 apresenta o tabuleiro tradicional de Gamão no seu estado inicial (antes de qualquer jogada).

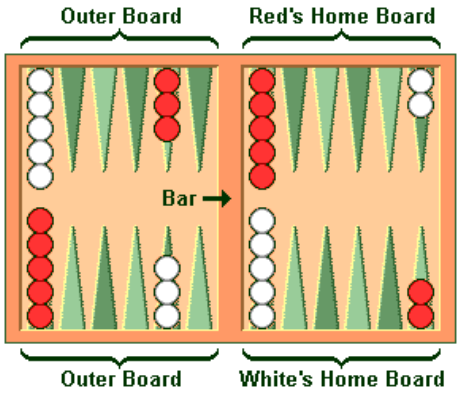



**Fig. 1 – Aspeto inicial do tabuleiro**

A lista seguinte apresenta as regras que iremos seguir na nossa versão do jogo.

- As 12 casas da parte de cima e as 12 da parte de baixo do tabuleiro são numeradas (1 a 12), contando da direita para a esquerda. Assim, as casas *Home* das vermelhas e das brancas são identificadas por valores de 1 a 6.
- Cada casa só pode ter peças de uma cor, e não há limite ao número de peças presentes em cada casa.
- As peças brancas deslocam-se no sentido contrário aos ponteiros do relógio e as vermelhas no sentido dos ponteiros do relógio.
- Cada jogador lança os dois dados e executa os movimentos (lances) correspondentes, sobre duas peças distintas ou sobre a mesma peça. Ganha o jogo quem primeiro levar todas as suas peças para a sua zona *Home*.
- Só é possível efetuar a jogada se a casa destino tiver no máximo uma peça do adversário (diz-se que a casa está livre). No caso de haver uma única peça do adversário, esta é capturada.
- Um jogador que tenha peças capturadas tem de as libertar antes de movimentar outras quaisquer peças. Para que isso seja possível, um dos dados da jogada terá de indicar um valor de casa livre (1 a 6) na *Home* do adversário. Se a casa do adversário tiver uma peça deste, então a peça é também capturada. No caso de ainda sobrar um lance na jogada após a libertação de peça capturada, este pode ser aplicada a qualquer peça (mesmo a que foi libertada).
- É obrigatório executar os lances correspondentes ao lançamento dos dados, a menos que não sejam possíveis de executar. Neste caso o jogador dá a vez ao seu adversário.

Em anexo é fornecida uma versão incompleta do jogo do gamão para jogar na consola do computador. O código está distribuído pelos seguintes ficheiros:

**gamoncons.c** – contém o código de apresentação e de execução das jogadas. Está completo.

**model.c** – contém a representação interna do tabuleiro e as funções que validam jogadas e efetuam movimentos de peças. Este módulo está por implementar.

**model.h** – contém as assinaturas (protótipos) das funções presentes em `model.c` (completo)

**utils.c** - funções utilitárias, por exemplo para a geração de valores aleatórios (completo).

**utils.h** – protótipos das funções definidas em `utils.c` (completo)

Uma jogada é composta por 3 componentes: <ação> <dado escolhido> <casa da peça a mover>:

**ação** – **M** para mover e **L** para libertar
**dado escolhido** – 1 ou 2
**casa da peça a mover** – o número onde a peça se encontra juntamente com a sua posição no tabuleiro (**c** para cima e **b** para baixo)

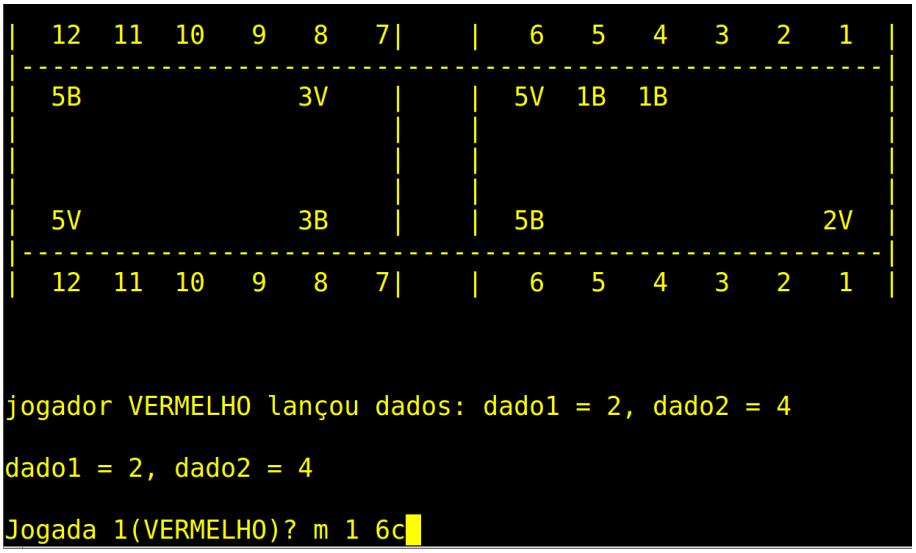
A fig. 3 apresenta a jogada o aspeto do tabuleiro pretendido e a primeira jogada do jogador das peças brancas. O comando (`m 1 1c`) significa "mover peça (branca) usando o dado1 a partir da casa 1 da parte de cima.

```
> gamoncons

|  12  11  10   9   8   7|    |   6   5   4   3   2   1  |
|-------------------------------------------------------|
|  5B             3V    |    |  5V                 2B  |
|                       |    |                         |
|                       |    |                         |
|                       |    |                         |
|  5V             3B    |    |  5B                 2V  |
|-------------------------------------------------------|
|  12  11  10   9   8   7|    |   6   5   4   3   2   1  |

jogador BRANCO lançou dados: dado1 = 5, dado2 = 3

dado1 = 5, dado2 = 3

Jogada 1(BRANCO)? m 2 1c
```

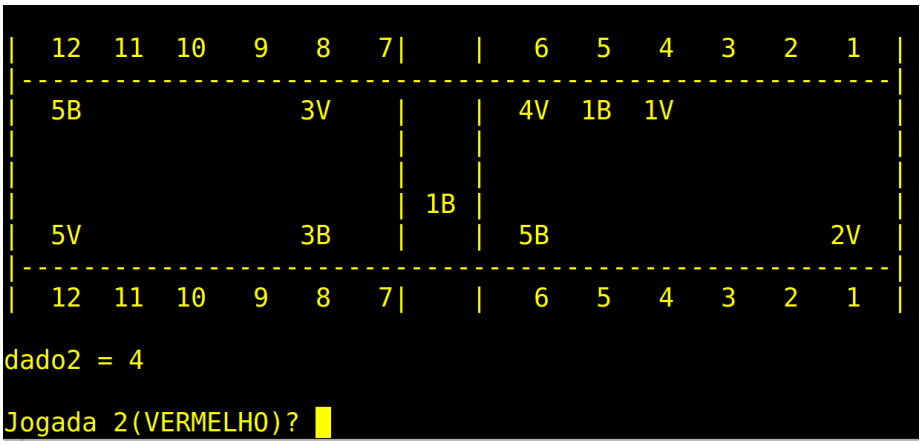
O resultado da jogada é:

```
|  12  11  10   9   8   7|    |   6   5   4   3   2   1  |
|-------------------------------------------------------|
|  5B             3V    |    |  5V      1B           1B  |
|                       |    |                         |
|                       |    |                         |
|                       |    |                         |
|  5V             3B    |    |  5B                 2V  |
|-------------------------------------------------------|
|  12  11  10   9   8   7|    |   6   5   4   3   2   1  |

dado1 = 5

Jogada 2(BRANCO)?
```

A seguir apresentamos um exemplo de jogada que envolve captura de peça:




No lance da peça vermelha a peça branca presente na casa 4 da parte de cima é capturada!




Bom trabalho.

Diogo Cardoso,
Jorge Martins,
Manuel Carvalho