

# Linguagens de Programação I

---

## Trabalho Prático

Este trabalho destina-se a todos os estudantes inscritos na unidade curricular de Linguagens de Programação I. Pretende-se que, usando a linguagem C, crie um programa que resolva o problema apresentado em Anexo.

Os estudantes deverão juntar-se em grupos de 2 elementos de modo a dividir, da melhor forma, as tarefas definidas neste trabalho. Excepcionalmente, e quando tal se justifique, poderão ser considerados grupos com outro número de elementos.

Cada grupo deve indicar quais os elementos que o constituem (nº aluno + nome completo + turma). Isto deve ser feito até ao final do dia 17 de Dezembro de 2008 através da página da disciplina em <http://www.estgf.ipp.pt/moodle/>. Esta operação deve ser efectuada por apenas um membro do grupo.

A entrega do trabalho deve ser feita até ao final do dia 18 de Janeiro de 2008 também na página da disciplina. A entrega inclui, além do código-fonte do programa, também um relatório cujas características são descritas mais abaixo. A apresentação e defesa do trabalho realizar-se-á em data ainda a anunciar.

Este trabalho servirá como elemento de avaliação, sendo objecto de classificação entre 0 e 20 valores e entrará no cálculo da nota final. Os elementos de um mesmo grupo podem ter classificações diferentes.

## Objectivos

Com este trabalho, pretende-se:

- Desenhar e implementar, eficaz e eficientemente, um programa de resolução do problema proposto;
- Promover o estudo e utilização dos conceitos sobre construções algorítmicas;
- Promover o estudo e utilização da linguagem de programação C;
- Promover o estudo e utilização de ferramentas de desenvolvimento de programas.

## **Cr terios de Avalia  o**

Para efeitos de classifica  o do trabalho, s o considerados os seguintes factores:

- Efic cia e efici ncia da implementa  o do programa;
- Clareza e legibilidade, traduzida numa escolha criteriosa dos nomes das vari veis e fun  es e na correcta identifica  o dos programas;
- Apoio   compreens o do c digo, traduzida em coment rios explicativos, inclu dos no ficheiro fonte;
- Utiliza  o de conceitos apresentados na disciplina, para a resolu  o de problemas usando o computador, nomeadamente subdividindo o problema em problemas de menor dimens o;
- Conte do do relat rio;
- Entrega atempada dos elementos de avalia  o;
- Apresenta  o e defesa do trabalho.

Qualquer tentativa de fraude, sob qualquer forma, ser  punida com a anula  o do trabalho.

## **Relat rio**

Dever  entregar um pequeno relat rio que descreva a forma como abordou o problema, que descreva todo o trabalho realizado, que mostre e comente os resultados obtidos e que apresente as respectivas conclus es. Mais concretamente, esse relat rio deve conter o seguinte:

- Funcionalidades implementadas/n o implementadas:
  -   Exemplo: Al m das funcionalidades pedidas, implementamos uma outra que nos permite obter informa  es estat sticas sobre as salas.
  -   Exemplo: N o conseguimos implementar as restri  es X,Y,Z.
- Explica  o e justifica  o do trabalho desenvolvido:
  -   Exemplo: Fizemos esta fun  o porque nos dava jeito para implementar a funcionalidade XPTO. Implementamo-la usando um vector de caracteres...

- Trabalho que ficou por desenvolver e como o poderiam fazer:
  - Exemplo: Apesar de não termos conseguido implementar a funcionalidade K, pensamos que usando uma estrutura XPTO, e ordenando-a de forma decrescente...
- Dificuldades encontradas:
  - Exemplo: Tivemos dificuldade em implementar a função XPTO porque não sabíamos como fazer a parte de...
- Não precisam de colocar algoritmos a não ser que vos seja útil para explicar algum raciocínio ou conceito.

## **Datas importantes**

- Definição de grupos de trabalho: até ao final do dia 17 de Dezembro de 2008
- Entrega: até ao final do dia 18 de Janeiro de 2008
- Defesa: a definir

# ANEXO

## Problema

Pretende-se que implemente em linguagem C o famoso jogo de cartas “Paciência”, também conhecido por *Solitaire* e popularizado pela sua inclusão no sistema operativo *Windows*.

### Objectivo do jogo

O objectivo do jogo é empilhar separadamente as cartas dos quatro naipes, cada uma na casa do seu naipe. As cartas devem ser empilhadas por ordem ascendente, isto é, Ás, 2, 3, 4, 5, 6, 7, 8, 9, 10, Valete, Dama, Rei.

### Preparação do jogo

Utiliza-se um baralho completo de 52 cartas. Deve-se fazer um monte com 7 cartas com as faces viradas para baixo, depois um monte com seis cartas colocado da mesma forma à esquerda do primeiro, e assim sucessivamente até um último monte de apenas uma carta. Vira-se a primeira carta de cada monte, totalizando sete cartas reveladas. As restantes cartas são colocadas na casa do baralho. Todas as cartas devem ser distribuídas de forma aleatória.

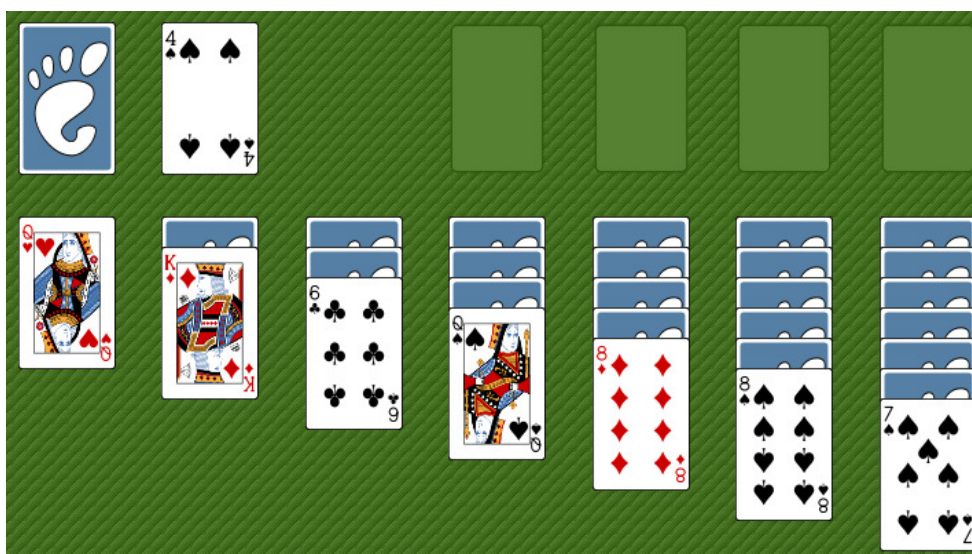


Figura 1 - Exemplo do início de um jogo

## Regras

Pode movimentar-se qualquer carta que esteja empilhada nos montes. Nesses montes, as cartas devem ser empilhadas da maior para a menor, alternando as cores. A movimentação das cartas deve respeitar essa sequência.

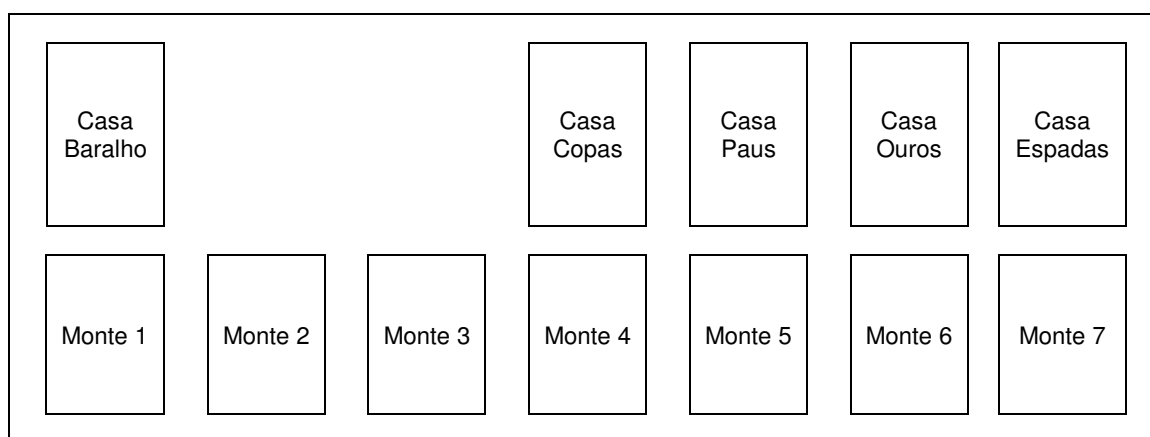
Num monte, podem movimentar-se quantas cartas se desejar, desde que estas sejam movidas para um outro monte com a carta superior adequada. Por exemplo, se um monte tem 7 de Espadas, 6 de Copas e 5 de Paus, essas três cartas podem ser colocadas sobre um monte cuja carta superior seja um 8 de um naipe vermelho.

Quando um monte estiver vazio (sem cartas), nele pode ser colocado um Rei. Sempre que uma das cartas cobertas ficar livre, ou seja, sem nenhuma carta em cima, o utilizador pode ou não escolher virá-la. Não é permitido ao jogador colocar cartas sobre cartas não viradas.

Do baralho pode ser retirada a carta do topo para ajudar a construir as sequências necessárias. Não é possível transferir cartas dos montes ou das casas para o baralho.

O jogo acaba quando conseguirmos colocar todas as cartas na casa do respectivo naipe, pela ordem correcta.

Qualquer dúvida que haja sobre o funcionamento do jogo ou suas regras deve ser esclarecida com o docente Nuno Branco.



**Figura 2 - Diagrama de disposição dos montes e casas**

## Funcionamento do programa

### Representação das cartas

Cada carta deve conter as seguintes informações: Valor, Naipes e Cor. Deve ainda ser apresentada usando o seu valor seguido do naipe e usando a cor correspondente à cor do naipe (exemplo: o 4 de Copas fica 4♥). As cartas cobertas são representadas simplesmente por "XX" e apresentadas na cor branca.

Valor	Representação
Rei	R
Dama	D
Valete	V
2 a 10	O próprio número
Ás	A
Carta coberta	XX

Naipes	Representação	Representação alternativa
Copas	♥ (código 3)	C
Ouros	♦ (código 4)	O
Paus	♣ (código 5)	P
Espadas	♠ (código 6)	E

Cor	Código
Vermelho	LIGHTRED
Preto	BLACK
Branco	WHITE

Figura 3 - Diferentes valores e códigos para apresentar cartas

Exemplo de como imprimir uma carta com o respectivo naipe e sua cor (Ás de Copas):

```
#include <stdio.h>

/* Seguem-se as bibliotecas necessárias para utilização de cores.
Caso não tenha no seu computador instalada alguma delas, estão
disponíveis na página da disciplina. */

#include <conio.h>
#include <conio.c>

main()
{
    system("color 20"); // altera a cor de fundo do ecrã para verde
    textcolor(LIGHTRED); // altera a cor do texto para vermelho
    printf("A\3"); // imprime o A (de ás) e o naipe de copas
}
```

Resultado: A♥

### Geração de números aleatórios

```
#include <time.h> // biblioteca necessária para correcta
                  // geração de números aleatórios

int aleatorio() // função que devolve números aleatórios
               // entre 1 e 10
{
    srand( (unsigned int) time(NULL) );
    return 1 + rand() % 10;
}
```

## Interface

O programa deve ser criado usando o Ambiente de Programação Dev-C++, para ser executado numa janela MS-DOS, em modo character.

O interface pode, por exemplo, seguir a disposição e apresentação ilustrada na figura abaixo.

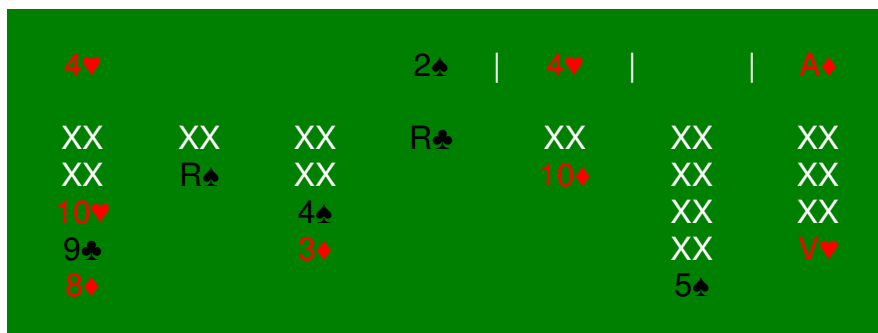


Figura 4 - Exemplo de um instante do jogo

## Pontuação e Recordes

Durante o jogo deve ser registada e apresentada a pontuação do jogador, com o objectivo de armazenar em ficheiro uma lista ordenada (TOP 5) das pontuações mais altas juntamente com o nome de quem as obteve. Essa lista pode ser consultada quando se inicia o programa ou quando se termina um jogo.

A pontuação funciona da seguinte forma:

- São atribuídos 10 pontos por cada carta correctamente empilhada na casa do seu naipe;
- São atribuídos 5 pontos por cada carta do baralho empilhada num dos montes;
- São atribuídos 5 pontos por cada carta virada nos montes, isto é, cartas com a sua face voltada para baixo;
- São retirados 15 pontos por cada carta retirada da casa do seu naipe e empilhada num dos montes;
- São retirados 50 pontos por cada vez que se percorrem todas cartas do baralho;
- Quando todas as cartas tiverem sido empilhadas nas respectivas casas de naipe, o jogo termina e a pontuação final é a obtida até esse momento.