

# E-folio A 2017/2018

## Alínea A

O e-fólio A é constituído por 4 alíneas, valendo 1 valor cada, devendo as mesmas serem realizadas sequencialmente, e podendo ser reutilizado código entre alíneas. A cotação total do e-fólio é de 4 valores. Os critérios de correção encontram-se no espaço da UC, sendo que 50% é destinado à funcionalidade, resultante da percentagem de casos de teste corretos. A realização do e-fólio na plataforma HackerRank não dispensa a entrega do relatório no espaço da UC.

O relatório deve indicar as alíneas realizadas e resultados obtidos, e descrever o código realizado e opções tomadas, o qual não deve ultrapassar as 4 páginas. Se realizou parcialmente uma das alíneas, descreva o que fez e como planeava completar a alínea. Deve colocar o código das alíneas realizadas no anexo, mesmo as que foram realizadas parcialmente, e não colocar código no corpo do relatório.

Este e-fólio é baseado no jogo BlackJack ou jogo 21 (<https://pt.wikipedia.org/wiki/Blackjack>). O jogo é jogado com quatro baralhos de 52 cartas cada, com 4 naipes e 13 números (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, V, D, R).

Pretende-se na alínea A que receba **um valor entre 0 e 51, e mostre uma carta**. Os primeiros 13 números (de 0 a 12) correspondem às cartas do naipe P de paus (de AP a RP), os seguintes 13 números correspondem às cartas do naipe de O de ouros (de AO a RO), os seguintes 13 às cartas do naipe C de copas, e os seguintes 13 números às cartas do naipe E de espadas.

A lista de todas as cartas por ordem numérica, de 0 a 51:

AP	2P	3P	4P	5P	6P	7P	8P	9P	10P	VP	DP	RP
AO	2O	3O	4O	5O	6O	7O	8O	9O	10O	VO	DO	RO
AC	2C	3C	4C	5C	6C	7C	8C	9C	10C	VC	DC	RC
AE	2E	3E	4E	5E	6E	7E	8E	9E	10E	VE	DE	RE

**Input (entrada):**

- número de 0 a 51
- Output (saída):**
- carta correspondente ao número de entrada
- Exemplos:** ver casos de teste visíveis

## Alínea B

Pretende-se nesta alínea que junte 4 baralhos, um baralho após o outro, em que cada baralho deve estar inicialmente ordenado de 0 a 51, e os baralhe da seguinte forma: para a primeira carta na posição 0 é sorteado um valor entre 0 e o total de cartas -1 (última posição possível), sendo a carta da posição 0 trocada com a carta sorteada. De seguida para a segunda carta na posição 1 é sorteado um valor 1 e o total de cartas -1, sendo a carta na posição 1 trocada com a carta sorteada. Este processo deve repetir-se até à última carta (processo igual à atividade formativa baralhar.c).

Pretende-se que mostre as 10 primeiras cartas dos 4 baralhos, após baralhados pelo processo descrito. Confirme com as execuções de exemplo.

Utilize a função randaux para gerar valores aleatórios. O código inicial fixo, não pode ser alterado, e contém a função randaux. É lido também um número na entrada de dados, de modo a fazer algumas chamadas à função randaux e ter-se acesso a várias sequências aleatórias.

- Input (entrada):**
- número de execuções iniciais da função randaux() - procedimento já codificado, manter
- Output (saída):**
- 10 primeiras cartas do baralho
- Exemplos:** ver casos de teste visíveis

## Alínea C

Continue da alínea anterior, mas em vez de mostrar as 10 primeiras cartas, mostre apenas as cartas que a banca selecciona. Sempre que a banca tenha menos de 17 pontos, pede uma carta. As cartas com número 'A' valem 1 ou 11 pontos, nas restantes cartas, as figuras V, D, R valem 10 pontos, e as outras valem o mesmo número de pontos que o seu número. Os Ases valem sempre 11 pontos, excepto se a pontuação for superior a 21, passando portanto um dos Ases a valor 1 ponto (se ultrapassar novamente 21 e existirem 2 Ases, o

segundo Ás naturalmente passa a valor 1). Deve procurar fazer o output idêntico aos casos de exemplo.

**Input (entrada):**

- número de execuções iniciais da função randaux() - procedimento já codificado, manter

**Output (saída):**

- cartas seleccionadas pela banca, e total de pontos, com indicação no caso de rebentar

**Exemplos:** ver casos de teste visíveis

## Alínea D

Esta alínea está dividida em duas partes. A segunda parte deve ser realizada apenas após passar todos os casos de teste da primeira parte.

### Parte 1: realização de um jogo

Situação em que o número de jogos (segundo dado de entrada) tem o valor 1.

Simule um jogo de *BlackJack*, assumindo o seguinte:

1. Após os 4 baralhos estarem baralhados, dar duas cartas à banca, e de seguida duas cartas a cada um de 4 jogadores;
2. Cada jogador irá solicitar cartas até estar satisfeito, mas cada jogador tem uma estratégia distinta:
  - 2.1 Jogador 1: se não tiver um As a valer 11 (e portanto a poder passar para 1), continuar a pedir cartas sempre que tenha menos de 14. No caso de ter um ou mais Ases a valer 11, continuar a pedir sempre que tenha menos de 17.
  - 2.2 Jogador 2: idêntico ao primeiro jogador mas continua a pedir cartas sempre que tenha menos de 16 na primeira situação, e 18 na segunda situação
  - 2.3 Jogador 3: idêntico, mas com valores 12 e 15
  - 2.4 Jogador 4: idêntico, mas com valores 17 e 19
- 2.5 Banca: a banca pede sempre cartas até que tenha 17 ou superior
3. Os jogadores que perdem, ficam com -1, os jogadores que ganham ficam com +1, e no caso de empate ficam com 0.

Atenção: confirme que na sequência de cartas inicial (igual à alínea B), as duas primeiras cartas vão para a banca, as outras duas cartas para o jogador 1, e assim sucessivamente. Após as 10 primeiras cartas distribuídas, as cartas seguintes vão para o jogador 1, até que este fique satisfeito ou rebente, e assim sucessivamente.

Em termos de saída, deve mostrar o detalhe do jogo, de acordo com os casos de exemplo visíveis.

### **Parte 2:** realização de vários jogos

Situação em que o número de jogos (segundo dado de entrada) tem o valor maior que 1.

Simular o número de jogos solicitados, e contabilizar os jogos ganhos por cada um dos 4 jogadores. No final mostrar essa informação, de acordo com os caso de exemplo visíveis.

Os dois últimos casos de teste estão dependentes da eficiência do código.

#### **Input (entrada):**

- número de execuções iniciais da função randaux() - procedimento já codificado, manter
- número de jogos a realizar - procedimento já codificado, manter

#### **Output (saída):**

- caso exista um só jogo, dar detalhe do jogo, caso existam vários jogos, dar o total de jogos por cada jogador

#### **Exemplos:**

- ver casos de teste visíveis, para 1 só jogo e para múltiplos jogos (output distinto).