

O e-fólio A é constituído por 4 alíneas, valendo 1 valor cada. As alíneas devem ser realizadas sequencialmente e o código pode ser reutilizado entre alíneas. A cotação total do e-fólio é de 4 valores. Os critérios de correção encontram-se no espaço da UC ([aqui](#)) na versão 6, sendo que 50% é destinado à funcionalidade, 25% à qualidade e 25% ao relatório. O código do e-fólio tem de ser submetido no espaço central / VPL ([aqui](#)), e o relatório tem de ser entregue no espaço da turma (E-fólio A), sem o qual o trabalho não será avaliado.

O relatório deve indicar as alíneas realizadas e resultados obtidos. Deve descrever o código realizado e opções tomadas, o qual não deve ultrapassar as 4 páginas (não contando com anexos e capa). Se realizou parcialmente uma das alíneas, descreva o que fez e como planeava completar a alínea.

Anexos do relatório:

- Código fonte (extensão .c), das alíneas realizadas, com o seguinte nome: <número estudante><alínea>.c. Um estudante com o número 123456, que tenha feito 3 alíneas, deve entregar os ficheiros 123456A.c 123456B.c e 123456C.c.
- Resultado do pre-check de cada alínea realizada com os casos de teste do enunciado, e os testes que realizou de forma independente dos testes do enunciado, que comprovem o bom funcionamento do programa.

Propõe-se no e-fólio A que desenvolva o jogo Memoria com regras e baralho especial, sendo uma paciência a jogar por um jogador humano.



Método de avaliação: Nota mais alta

Síntese das tentativas anteriores

Tentativa	Estado	Revisão
1	Em progresso	

Pergunta **1**

Por responder

Nota: 1,00

O jogo Memória utiliza um conjunto de cartas com três características: operação, número, letra.

A operação pode ser uma das 4 operações básicas: $+$, $-$, $*$, $/$. O número pode ser um dos seguintes: **1, 2, 3, 4**. A letra pode ser uma das seguintes: **A, B, C, D**. Existem 64 cartas possíveis distintas ($4*4*4$), sendo um baralho constituído pelas 64 cartas.

Considere o seguinte código de 0 a 63 para identificar as cartas:

0: +1A	8: +3A	16: -1A	24: -3A	32: *1A	40: *3A	48: /1A	56: /3A
1: +1B	9: +3B	17: -1B	25: -3B	33: *1B	41: *3B	49: /1B	57: /3B
2: +1C	10: +3C	18: -1C	26: -3C	34: *1C	42: *3C	50: /1C	58: /3C
3: +1D	11: +3D	19: -1D	27: -3D	35: *1D	43: *3D	51: /1D	59: /3D
4: +2A	12: +4A	20: -2A	28: -4A	36: *2A	44: *4A	52: /2A	60: /4A
5: +2B	13: +4B	21: -2B	29: -4B	37: *2B	45: *4B	53: /2B	61: /4B
6: +2C	14: +4C	22: -2C	30: -4C	38: *2C	46: *4C	54: /2C	62: /4C
7: +2D	15: +4D	23: -2D	31: -4D	39: *2D	47: *4D	55: /2D	63: /4D

Pretendemos que solicite ao utilizador um valor para o código, e retorne a carta correspondente. Caso o código não seja válido, deve retornar carta inválida.

Entrada:

- Número inteiro

Saída:

- Carta correspondente ao código do número fornecido ou mensagem "Carta invalida"

Para clarificar o que é pedido, foi elaborada a seguinte tabela de casos de teste:

Entrada	Saída
-1	Carta invalida
100	Carta invalida
0	+1A
51	/1D
21	-2B
19	-1D
41	*3B

Reiniciar

1

Pergunta **2**

Por responder

Nota: 1,00

Pretendemos agora que crie um baralho com as 64 cartas, por ordem de código. De seguida baralhe o baralho, com o mesmo algoritmo solicitado em "Atividades 2" | baralhar.c | página 44. Pode reutilizar a sua resolução nesta atividade formativa, ou utilizar a resolução de referência, mantendo no código a referência relativamente à origem do código que tiver utilizado. A resolução de referência é a seguinte função:

```
int* Baralhar(int v[], int n)
{
    int i, j, aux;
    for (i = 0; i < n - 1; i++)
    {
        j = i + randaux() % (n - i);
        aux = v[i];
        v[i] = v[j];
        v[j] = aux;
    }
}
```

Pedimos que mostre o baralho dispondo as cartas em 8 x 8, com as primeiras 8 posições na primeira coluna e por ordem de linha, as segundas 8 posições na segunda coluna por ordem de linha, e assim sucessivamente. Deve mostrar o número de linha/coluna, de acordo com a seguinte formatação:

```
[1] [2] [3] [4] [5] [6] [7] [8]
[1] *3B *1C +2B *2A /1D /4D -1C /3D
[2] +3B /2B -2B *4A *2D *4D /4C +2D
[3] +4A *4B +3D *3D -4D +2A -1B /3A
[4] -4B +4B /2D +2C -2D /2A +1A -3A
[5] *1B /1B -3D /2C -1D /3B *4C -2C
[6] *1D /4B /1A +4C *2C -4A *3A +4D
[7] /3C -3B /4A *2B +3A +1C /1C *3C
[8] +3C -1A +1B -3C -2A +1D -4C *1A
```

Utilize a função randaux() para gerar números aleatórios, tal como nas atividades formativas.

Como a função randaux() tem a semente fixa, esta deve ser chamada algumas vezes no código inicial, de acordo com um valor fornecido pelo utilizador, denominado o "desperdício". Este desperdício é necessário para permitir gerar diferentes baralhos nos casos de teste.

Entrada:

- Número inteiro positivo

Saída:

- Baralho apresentado no formato indicado em cima

Para clarificar o que é pedido, foi elaborada a seguinte tabela de casos de teste:

Entrada	Saída
---------	-------

Entrada	Saída
	<div>[1] [2] [3] [4] [5] [6] [7] [8]</div> <div>[1] *3B *1C +2B *2A /1D /4D -1C /3D</div> <div>[2] +3B /2B -2B *4A *2D *4D /4C +2D</div> <div>[3] +4A *4B +3D *3D -4D +2A -1B /3A</div>
0	<div>[4] -4B +4B /2D +2C -2D /2A +1A -3A</div> <div>[5] *1B /1B -3D /2C -1D /3B *4C -2C</div> <div>[6] *1D /4B /1A +4C *2C -4A *3A +4D</div> <div>[7] /3C -3B /4A *2B +3A +1C /1C *3C</div> <div>[8] +3C -1A +1B -3C -2A +1D -4C *1A</div>
	<div>[1] [2] [3] [4] [5] [6] [7] [8]</div> <div>[1] *1D /3A /2C *4C -1B *2A *2C *4A</div> <div>[2] +1A /1B /1C /2A *3C -2A -4D *4B</div> <div>[3] -4A -2B -3C -3A +2D /3D /3C *3A</div>
1	<div>[4] -1C -3B +4C /4B +2B -1D /3B *3B</div> <div>[5] +3A *3D /4A /1D +4A /1A -3D +2C</div> <div>[6] *2B -4B /2B -2D /4C +1B *2D +1D</div> <div>[7] -1A /2D +3B -4C +3D +4D *4D *1A</div> <div>[8] +2A +3C +1C +4B *1B /4D -2C *1C</div>

No primeiro caso, os códigos iniciais resultantes da função baralhar são: 41 9 12. Estes códigos correspondem às cartas: *3B +3B +4A, que correspondem às primeiras posições no primeiro caso. Já no segundo caso os códigos iniciais resultantes são: 35 0 28. Estes códigos correspondem às cartas: *1D +1A -4A, correspondendo às cartas nas primeiras posições do segundo caso.

Reiniciar

```

1  #include <stdio.h>
2
3  unsigned int randaux()
4  {
5      static long seed=1;
6      return(((seed = seed * 214013L + 2531011L) >> 16) & 0x7fff);
7  }
8
9  void main()
10 {
11     int desperdicio, i;
12     scanf("%d", &desperdicio);
13     for(i=0; i<desperdicio; i++)
14         randaux();

```

Run 

Pre-check 

Pergunta **3**

Por responder

Nota: 1,00

Vamos agora simular uma jogada do jogo da Memória. Para tal o utilizador deve escrever uma sequência de 1 a 4 números inteiros, que correspondam a posições do baralho. Uma posição é fornecida pela sua coluna nas dezenas, e linha nas unidades. Exemplo utilizando o baralho da alínea B:

```

[1] [2] [3] [4] [5] [6] [7] [8]
[1] *3B *1C +2B *2A /1D /4D -1C /3D
[2] +3B /2B -2B *4A *2D *4D /4C +2D
[3] +4A *4B +3D *3D -4D +2A -1B /3A
[4] -4B +4B /2D +2C -2D /2A +1A -3A
[5] *1B /1B -3D /2C -1D /3B *4C -2C
[6] *1D /4B /1A +4C *2C -4A *3A +4D
[7] /3C -3B /4A *2B +3A +1C /1C *3C
[8] +3C -1A +1B -3C -2A +1D -4C *1A

```

Na posição 24 está a carta +4B. Se o jogador escrever a sequência 55 71 16 33, está a selecionar as cartas -1D -1C *1D +3D. Assim, o baralho deve ser mostrado apenas com as cartas selecionadas, ficando as restantes ocultas, da seguinte forma:

```

[1] [2] [3] [4] [5] [6] [7] [8]
[1] ### ### ### ### ### ### -1C ###
[2] ### ### ### ### ### ### ### ###
[3] ### ### +3D ### ### ### ### ###
[4] ### ### ### ### ### ### ### ###
[5] ### ### ### ### -1D ### ### ###
[6] *1D ### ### ### ### ### ### ###
[7] ### ### ### ### ### ### ### ###
[8] ### ### ### ### ### ### ### ###

```

Caso a posição não seja válida, ou seja, não tenha correspondência na matriz de 8x8, essa posição e seguintes deve ser desconsiderada, terminando a jogada. O mesmo ocorre na situação em que há mais que 4 números introduzidos.

Entrada:

- Número inteiro positivo
- Sequência de números inteiros na mesma linha, separados por espaços - simulação de uma jogada

Saída:

- Baralho no formato indicado em cima, mostrando apenas cartas nas posições da jogada

Entrada	Saída
	<pre> [1] [2] [3] [4] [5] [6] [7] [8] [1] ### ### ### ### ### ### -1C ### [2] ### ### ### ### ### ### ### ### [3] ### ### +3D ### ### ### ### ### [4] ### ### ### ### ### ### ### ### [5] ### ### ### ### -1D ### ### ### [6] *1D ### ### ### ### ### ### ### [7] ### ### ### ### ### ### ### ### [8] ### ### ### ### ### ### ### ### </pre>
0	
55 71 16 33	

Entrada	Saída
	<div>[1] [2] [3] [4] [5] [6] [7] [8]</div> <div>[1] ### ##</div> <div>[2] ### ##</div> <div>[3] ### ##</div> <div>[4] ### ##</div> <div>[5] ### ## /1D</div> <div>[6] ### ##</div> <div>[7] ### ##</div> <div>[8] +2A ### ##</div>
1	
18 45 91 53	

O primeiro caso já estava dado no enunciado, sendo o resultado previsto. O segundo caso temos o baralho do segundo exemplo da alínea B, tendo sido escolhidas as posições 18 e 45. A posição 91 sai fora da matriz 8 x 8 não sendo processado mais nada.

Reiniciar

```

1  #include <stdio.h>
2
3  unsigned int randaux()
4  {
5      static long seed=1;
6      return(((seed = seed * 214013L + 2531011L) >> 16) & 0x7fff);
7  }
8
9  void main()
10 {
11     int desperdicio, i;
12     scanf("%d", &desperdicio);
13     for(i=0; i<desperdicio; i++)
14         randaux();
15
16
17
18 }
```

Run

Pre-check

Pergunta **4**

Por responder

Nota: 1,00

Vamos agora terminar o jogo, efetuando uma sequência de jogadas, de acordo com as seguintes regras:

1. O baralho inicia-se totalmente virado para baixo, sem indicação de qualquer posição das cartas;
2. O jogador faz uma jogada nos moldes da alínea C (indica entre 1 e 4 posições);
3. As cartas jogadas são visualizadas. No caso de terem todas o mesmo operador, ou o mesmo número, ou o mesmo naipe, são retiradas do baralho. Caso contrário mantêm-se no baralho.
4. As cartas retiradas do baralho, são mostradas com três espaços na sua posição.
5. Caso seja escolhida uma posição com uma carta retirada, o jogo termina imediatamente.
6. Caso não existam mais cartas, termina o jogo, caso contrário ir para a regra 2.
7. Ao terminar o jogo, indicar o número de jogadas e o número de cartas que restam.

De modo a reduzir o número de dados de entrada, caso o jogador introduza uma jogada com 0 posições válidas, parar de solicitar jogadas ao jogador humano. Em vez disso, adoptar o seguinte procedimento:

- Em cada jogada selecionar apenas 1 posição, a primeira posição que tiver uma carta (valor mais baixo de coluna*8+linha).
- Executar jogadas desta forma até que o jogo termine

Entrada:

- Número inteiro positivo
- Sequência de números inteiros na mesma linha, separados por espaços, colocando tantas linhas quantas as jogadas a efetuar

Saída:

- Mostrar os 5 baralhos iniciais, executar as jogadas mas não mostrar mais baralhos, exceto o final.
- Indicar no final do jogo o número de jogadas e o número de cartas que restam, de acordo com os exemplos

Para clarificar o que é pedido, foi elaborada a seguinte tabela de casos de teste:

Exemplo 1 - Entrada	Exemplo 1 - Saída	Exemplo 2 - Entrada	Exemplo 2 - Saída
------------------------	-------------------	------------------------	-------------------

Exemplo 1 - Entrada	Exemplo 1 - Saída	Exemplo 2 - Entrada	Exemplo 2 - Saída
	<div><div>[1][2][3][4][5][6][7][8]</div><div>[1]########################</div><div>[2]########################</div><div>[3]########################</div><div>[4]########################</div><div>[5]########################</div><div>[6]########################</div><div>[7]########################</div><div>[8]########################</div><div>[1][2][3][4][5][6][7][8]</div><div>[1]##################-1C###</div><div>[2]########################</div><div>[3]######+3D############</div><div>[4]########################</div><div>[5]############-1D#########</div><div>[6]*1D##################</div><div>[7]########################</div><div>[8]########################</div><div>[1][2][3][4][5][6][7][8]</div><div>[1]*3B##################</div><div>[2]+3B##################</div><div>[3]########################</div><div>[4]###+4B###############</div><div>[5]###/1B###############</div><div>[6]########################</div><div>[7]########################</div><div>[8]########################</div><div>[1][2][3][4][5][6][7][8]</div><div>[1]##################/3D</div><div>[2]#####################</div><div>[3]######+3D############</div><div>[4]#####################</div><div>[5]#####################</div><div>[6]#####################</div><div>[7]##################*3C</div><div>[8]#####################</div><div>[1][2][3][4][5][6][7][8]</div><div>[1]############/4D###</div><div>[2]##################</div><div>[3]##################</div><div>[4]###/2D######/2A###</div><div>[5]##################</div><div>[6]######/1A#########</div><div>[7]##################</div><div>[8]##################</div><div>[1][2][3][4][5][6][7][8]</div><div>[1]############</div><div>[2]##################</div><div>[3]##################</div><div>[4]###############</div><div>[5]##################</div><div>[6]##################</div><div>[7]##################</div><div>[8]############</div><div>Jogo com 6 jogadas. Restam 49 cartas.</div></div>	<div><div>[1][2][3][4][5][6][7][8]</div><div>[1]########################</div><div>[2]########################</div><div>[3]########################</div><div>[4]########################</div><div>[5]########################</div><div>[6]########################</div><div>[7]########################</div><div>[8]########################</div><div>[1][2][3][4][5][6][7][8]</div><div>[1]#####################</div><div>[2]#####################</div><div>[3]#####################</div><div>[4]#####################</div><div>[5]#########/1D#########</div><div>[6]#####################</div><div>[7]#####################</div><div>[8]+2A##################</div><div>[1][2][3][4][5][6][7][8]</div><div>[1]#####################</div><div>[2]###/1B###############</div><div>[3]###############/3D###</div><div>[4]#####################</div><div>[5]#####################</div><div>[6]#####################</div><div>[7]###############*4D###</div><div>[8]#####################</div><div>[1][2][3][4][5][6][7][8]</div><div>[1]#########*4C#########</div><div>[2]#####################</div><div>[3]#####################</div><div>[4]#####################</div><div>[5]#####################</div><div>[6]#####################</div><div>[7]#####################</div><div>[8]######+1C######-2C###</div><div>[1][2][3][4][5][6][7][8]</div><div>[1]#####################</div><div>[2]#####################</div><div>[3]#####################</div><div>[4]#####################</div><div>[5]##################+2C</div><div>[6]############/4C###*2D###</div><div>[7]#####################</div><div>[8]##################</div><div>[1][2][3][4][5][6][7][8]</div><div>[1]</div><div>[2]</div><div>[3]</div><div>[4]</div><div>[5]</div><div>[6]</div><div>[7]</div><div>[8]</div><div>Jogo com 77 jogadas. Restam 0 cartas.</div></div>	
0		1	
55 71 16 33		18 45 91 53	
11 12 24 25		22 77 63	
81 87 33 1		41 78 38	
61 64 34 36		85 76 56	
88 28 38 68		64 73 68	
14 11 41 13		84 74 24	
33 85 56 25		53 66 13	
71 16 57 26		61 48 14	
86 35 38 73		58 34 12	
78 88 77 44		11 33 21	
63 45 62 43		65 55 87	
27 32 54 46		32 82 83	
23 22 36 65		26 16 15	
52 81 66 42		46 81 52	
67 34 48 76		35 18 75	
21 58 53 18		51 25 37	
68 12 64 87		28 57 88	
51 84 24 75		31 43 27	
74 61 47 28		54 36 86	
83 31 17 55		23 17 71	
82 15 37 72		67 47 45	
0		42 44 62	
		72	
		0	

Podemos ver que o número de baralhos é igual, embora o número de jogadas não seja. Essa situação deve-se a que apenas os 5 primeiros baralhos, e o último, são mostrados. No exemplo 1, o jogo termina, já que a posição 33 foi solicitada na jogada que seria a 7ª, já que a posição 11 foi solicitada na 6ª jogada, mas essa posição já não tinha uma carta. No exemplo 2 após os 5 primeiros baralhos, seguiram-se ainda muitas jogadas, terminando o jogo apenas com 77 jogadas. Os dados de entrada têm no entanto muito menos que 77 jogadas. Assim que ocorreu a primeira jogada sem posições válidas, o procedimento de escolha de uma só carta foi adoptado para as restantes jogadas.

[Reiniciar](#)

```
1  #include <stdio.h>
2
3  unsigned int randaux()
4  {
5      static long seed=1;
6      return(((seed = seed * 214013L + 2531011L) >> 16) & 0x7fff);
7  }
8
9  void main()
10 {
11     int desperdicio, i;
12     scanf("%d", &desperdicio);
13     for(i=0; i<desperdicio; i++)
14         randaux();
```

Run Pre-check ☒