P1 - Algoritmos e Estruturas de Dados I - 15/04/2021

Nome Completo:

NLN = nome todo = 22

NLUS = último sobrenome = 7

A = |NLN|, B = |NLUS|

Questão 1. (2 pontos)

Dada a expressão aritmética abaixo, totalmente parentizada, mostrar a **situação das pilhas** para:

a) **conversão para notação polonesa reversa** e

b) cálculo da expressão.

Expressão - Conversão:

E:

((((9 - 2) - 8) * 7) - 2)

2 2

Pilha:

- - - - * * - -

Saída:

9

9 2

2

9 2 -

2

9	2	-	8					
	2							
9	2	-	8	-				
	2							
9	2	-	8	-	7			
	2							
9	2	-	8	-	7	*		
	2							
9	2	-	8	-	7	*	2	
	2						2	
9	2	-	8	-	7	*	2	-
	2						2	

Expressão - Cálculo:

E:

9	22	-	8	-	7	*	22	-

Pilha:

	22		8		7		22	
9	9	-13	-13	-21	-21	-147	-147	-169

Questão 2. (3 pontos) - Embaralhamento

Suponha que um baralho esteja representado em um **vetor B[1..52]**, cada elemento do vetor contendo um número entre 1 e 52. Inicialmente, o vetor está ordenado. Escrever um algoritmo para embaralhar esse baralho usando a seguinte ideia. Serão feitos **51 sorteios**, usando a função **Sorteio(k),** que devolve um número aleatório entre 1 e k. Suponha essa função pronta, não sendo necessário descrevê-la. Cada sorteio é feito para preencher a posição k, da direita para a esquerda. A cada sorteio, a carta da posição sorteada é **trocada** com a carta da posição k.

No exemplo a seguir, por questões de espaço, o baralho conteria apenas 10 cartas.

Baralho inicial:

1	2	3	4	5	6	7	8	9	10

O primeiro sorteio é feito através da chamada Sorteio(10) e suponhamos que retorne 5. O baralho fica:

1	2	3	4	10	6	7	8	9	5

O segundo sorteio é feito através da chamada Sorteio(9) e suponhamos que retorne 3. O baralho fica:

1	2	9	4	10	6	7	8	3	5

O terceiro sorteio é feito através da chamada Sorteio(8) e suponhamos que retorne 3. O baralho fica:

1	2	8	4	10	6	7	9	3	5

etc.

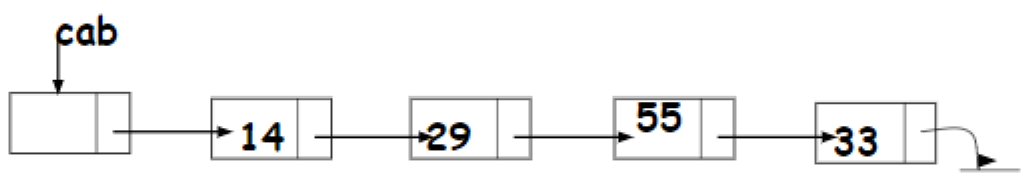
Resposta:

```
Embaralha(ref inteiro V[]):
    inteiro contador

    para contador <- 52 até 2 passo -1:
        sorteio(k)
```

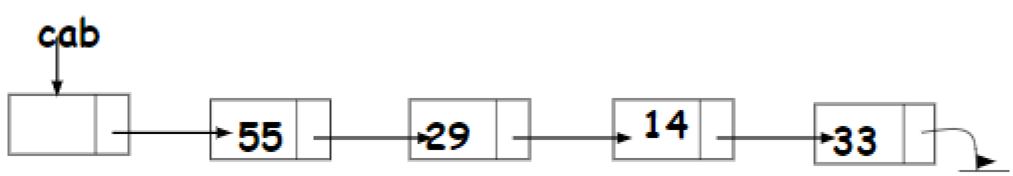
```
            V[contador], V[k] <- V[k], V[contador]

        retornar V


inteiro B[*] <- [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44,
45, 46, 47, 48, 49, 50, 51, 52]

escrever(Embaralha())
```

Questão 3. (3 pontos) – Troca de elementos em uma Lista Encadeada
Suponha uma lista encadeada circular com nó cabeça **(cab)** e **n>1** nós . Cada nó contém uma chave **k,** que é um inteiro (todos os inteiros são distintos) e um ponteiro **prox** para o próximo elemento. Escreva um algoritmo **Troca(cab)** para trocar de posição os elementos **k1** (o maior da lista) com **k2**, (o menor). Dizer qual a complexidade do algoritmo e justificar.

Ex: Suponha a lista abaixo.



Após a execução de Troca(cab), teríamos:



Questão 4. (2 pontos) – Notação assintótica/Ferramentas Matemáticas
Responda **C(Certo)** ou **E(Errado)** para as afirmativas abaixo. Cada resposta errada anula uma certa.

(C) $f(n) = 25.2^{n+2}$ é $O(2^n)$

(E) $f(n) = 25n^3$ é $\Theta(n^2)$

(E) A complexidade de um algoritmo indica se é fácil compreendê-lo ou não.

(C) É possível aplicar Indução Matemática para descobrir propriedades de algoritmos.

Boa sorte.