



4ª Avaliação (5/Julho) – AE22CP – 2017/1

Instruções para a avaliação:

prof.: Jean P. Martins (jeanmartins@utfpr.edu.br)

- Leia a prova com atenção e coloque o seu nome completo na mesma.
- A avaliação é individual.
- A interpretação da prova faz parte da avaliação.
- É PROIBIDA a consulta à Internet.
- O peso de cada exercício está indicado ao final do enunciado.
- Todos os códigos deverão ser submetidos em: <https://www.hackerrank.com/ae22cp-20171-sub>

NOTA:

NOME : _____

HACKERRANK-ID: _____

Uma implementação eficiente da estrutura de dados conjunto (Set) pode ser feita utilizando-se *Tabelas Hash*. Considere uma tabela hash implementada com **tratamento de colisões por ENCADEAMENTO SEPARADO**. Uma estrutura Set pode ser definida da seguinte forma:

```
#define CAPACIDADE 1021 // Utilize esta capacidade
typedef struct {
    Lista* tabela_hash[CAPACIDADE];
} Set;
```

1. TABELAS HASH

Tabelas hash: conjuntos (interseção) _____ **** (Peso 3,5) ****

Implemente uma função para interseção de dois conjuntos de inteiros.

```
Set* intersecao(Set* A, Set* B);
```

Formato de entrada

O programa receberá como entrada, primeiramente o número de elementos a serem inseridos. Seguido pelos respectivos elementos, tanto para A, quanto para B. Exemplo:

```
24
56 2 158 66 174 51 192 51 58 66 3 197 164 118 73 41 62 89 75 192 32 169 192 106
8
195 41 45 101 191 11 102 4
```

Restrições

- Elementos cujo hash colidam, devem ser inseridos **no início da lista correspondente**.

Output Format

O conjunto interseção deve conter todos elementos de A que também estejam em B. No exemplo de entrada acima, C={41}, portanto a saída será:

```
41
```

- Há um espaço ' ' após cada número impresso, inclusive no final

PILHAS

Pilhas: palíndromo (xCy) _____ **** (Peso 2,0) ****

Implemente uma função que verifique se uma string da forma "xCy" representa um palíndromo.

```
int palindromo(char* A);
```

Formato de entrada

O programa receberá como entrada uma string contendo o caractere 'C', o qual delimita o fim da primeira parte da string.

A string é um palíndromo se sua primeira parte for igual ao inverso da segunda parte.

SIM ou NÃO

(Escolha apenas uma das duas questões abaixo**)**

**** (Peso 2,5) ****

Formato de entrada

9

7

Output Format (Imprima)

1 2 5 7 10 15 20 25 30 40 50 60 70 80 90 100

**** (Peso 2,5) ****

```
Lista* B = strParaLista(str_b);
```

```
int strcmplist(Lista* A, Lista* B);
```

```
999999999999999999calksdfacdsl
```

A igual a B

se $c_1 > c_2$: A maior que B

```
999999999999999999calksdfacdsl
```

**** (Peso 2,0) ****

Em geral, lista encadeadas são classificadas como simples, ou duplas. Nas listas simples, cada item da lista tem apenas um ponteiro para o próximo item, enquanto nas duplamente encadeadas há também um ponteiro para o item anterior. Portanto, uma lista simples só pode ser percorrida do início ao fim, enquanto uma duplamente encadeada também pode ser percorrida do fim ao início. Como essa característica pode ser utilizada para diminuir o tempo gasto na obtenção do item em uma dada posição i na lista duplamente encadeada?