

Exercícios - Lista1 referente à P1

Resolva os enunciados abaixo como atividade para compor a nota N1. Cada exercício vale 0.5 (ponto).

Os **exercícios(funções)para nota** devem ser entregues impressos em dupla(identifique a matrícula, nome e turma de cada integrante da dupla), no dia da prova N1.

As funções **devem ter o código fonte comentado**. Os exercícios que não estiverem dentro das especificações acima serão desconsiderados!

As funções devem ser escritas de acordo com os protótipos anunciados em cada questão.

EXERCICIOS

1)Escreva um **procedimento** que receba um ponteiro para uma string de tamanho desconhecido. A função deve retirar todos os espaços em branco contidos na string (se existirem), modificando (reduzindo) a string original.

OBS: Não use nenhuma função da biblioteca string

```
void retira_branco(char *frase)
```

PILHAS E FILAS

Os arquivos da interface **pilha.h** e **fila.h** contem, respectivamente, as seguintes funções:

<pilha.h>

```
Pilha *pilha_cria(void);
void pilha_insere(Pilha *p,
float v);
float pilha_retira(Pilha *p);
int pilha_vazia(Pilha *p);
void pilha_libera(Pilha *p);
```

<fila.h>

```
Fila *fila_cria(void);
void fila_insere(Fila *f,
int v);
int fila_retira(Fila *f);
int fila_vazia(Fila *f);
void fila_libera(Fila *f);
```

O **typedef** usado para a definição dessas estruturas é desconhecido, assim como não se sabe se as pilhas ou filas foram implementadas por meio de vetor ou lista encadeada.

Portanto, não use qualquer aspecto ligado à implementação física, mas tão somente as funções apresentadas!

2)Usando apenas o arquivo de interface **pilha.h**, **criado para uma pilha de números reais**, escreva uma **função** que receba um ponteiro para uma pilha e retorne uma nova pilha onde cada elemento é a metade do elemento da pilha original. A pilha original **não deve** ficar vazia.

```
Pilha* dividePilha(Pilha *p)
```

3)Usando apenas o arquivo de interface **fila.h**, **criado para uma fila de números inteiros**, escreva uma função que receba ponteiros para duas filas. A função deve **criar uma nova fila** com o conteúdo intercalado

das filas f1 e f2. As filas originais **não devem** ficar vazias. A função retorna para o programa principal o endereço dessa nova fila.

Atenção: as filas podem ter tamanhos diferentes!!

```
Fila *intercala_fila (Fila *f1, Fila *f2)
```

LISTAS ENCADEADAS SIMPLES

4) Dado o typedef de lista encadeada:

```
typedef struct lista{
    char nome[20];
    char fone[9];
    struct lista *prox;
}Lista;
```

a) Crie um **procedimento** que dado um nome, mostre o fone associado ao nome ou informe que o nome não foi encontrado.

```
void mostraFone(Lista *l, char *nome)
```