

# Análise e Desenvolvimento de Sistemas

# ADSMA3

# Felix Petiz Bonilho RA: 1680481911002

# Estrutura de Dados

# Lista referente à P1

# São Caetano do Sul – SP

# 2020

# Exercícios

## 1.

**void retira\_brancos(char \*frase) {**

**for (int cont = 0; \*frase != '\0'; frase++) { */\* repete o loop até o fim da string (enquanto não chegar no ‘\0’) \*/***

**\*frase = \*(frase + cont); */\* sobrescreve o conteúdo do caractere que está na posição atual com o conteúdo da posição atualizada (posição atualizada com base na quantidade de caracteres em brancos já percorridos) \*/***

**if (\*frase == ' ') { */\* verifica se o caractere atual está em branco \*/***

**cont++; */\* atualiza o contador com a quantidade de caracteres em branco já percorridos \*/***

**\*frase = \*(frase + cont); */\* substitui o caractere em branco pelo seguinte \*/***

**}**

**}**

**}**

## 2.

**Pilha\* dividePilha(Pilha \*p) {**

**Pilha \*metade = pilha\_cria(), \*aux = pilha\_cria(); */\* Criação de pilhas auxiliares \*/***

**float auxNum; *// Criação de auxiliar numérico***

**while (!pilha\_vazia(p)) */\* repete o loop até que a pilha original fique vazia \*/***

**pilha\_insere(aux, pilha\_retira(p)); */\* insere elementos da pilha original na auxiliar \*/***

**while (!pilha\_vazia(aux)) { */\* repete o loop até que a pilha auxiliar fique vazia \*/***

**auxNum = pilha\_retira(aux); */\* auxiliar numérico recebe o elemento retirado da pilha auxiliar \*/***

**pilha\_insere(metade, auxNum/2); */\* insere elementos na 2° pilha auxiliar com metade de seu valor \*/***

**pilha\_insere(p, auxNum); */\* insere elementos de volta na pilha original \*/***

**}**

**pilha\_libera(aux); *// libera primeira pilha auxiliar da memória***

**return metade; *// retorna pilha com os elementos valendo metade***

**}**

## 3.

**Fila \*intercala\_fila (Fila \*f1, Fila \*f2) {**

**Fila \*aux1 = fila\_cria(), \*aux2 = fila\_cria(); *// Criação de filas***

**Fila \*intercalada = fila\_cria(); *// auxiliares***

**int auxNum; *// Criação de auxiliar numérico***

**while (!fila\_vazia(f1) || !fila\_vazia(f2)) { */\* repete o loop até que as duas filas originais fiquem vazias \*/***

**if (!fila\_vazia(f1)) {*//Verifica se a 1° fila original está vazia***

**auxNum = fila\_retira(f1); */\* auxiliar numérico recebe o elemento retirado da 1° fila original \*/***

**fila\_insere(intercalada, auxNum); */\* insere elementos da 1° fila original na intercalada \*/***

**fila\_insere(aux1, auxNum); */\* insere elementos da 1° fila original na 1° auxiliar \*/***

**}**

**if (!fila\_vazia(f2)) {*//Verifica se a 2° fila original está vazia***

**auxNum = fila\_retira(f2); */\* auxiliar numérico recebe o elemento retirado da 2° fila original \*/***

**fila\_insere(intercalada, auxNum); */\* insere elementos da 2° fila original na intercalada \*/***

**fila\_insere(aux2, auxNum); */\* insere elementos da 2° fila original na 2° auxiliar \*/***

**}**

**}**

**while (!fila\_vazia(aux1)) */\* repete o loop até que a 1° fila auxiliar fique vazia \*/***

**fila\_insere(f1, fila\_retira(aux1)); */\* insere elementos de volta na 1° fila original \*/***

**fila\_libera(aux1); *// libera 1° fila auxiliar da memória***

**while (!fila\_vazia(aux2)) */\* repete o loop até que a 2° fila auxiliar fique vazia \*/***

**fila\_insere(f2, fila\_retira(aux2)); */\* insere elementos de volta na 2° fila original \*/***

**fila\_libera(aux2); *// libera 2° fila auxiliar da memória***

**return intercalada; *// retorna fila com os elementos intercalados***

**}**

## 4. a.

**void mostraFone(Lista \*l, char \*nome) {**

**Lista \*aux; *// Criação de lista auxiliar***

**for (aux = l; aux != NULL; aux = aux->prox) { */\* repete o loop até o final da lista \*/***

**if (strcmp(strlwr(aux->nome), strlwr(nome)) == 0) { */ verifica se o nome do elemento que está na posição atual é igual ao nome informado \*/***

**printf("\n\tFone associado ao nome %s:", nome); *// imprime no***

**printf("\n\t%s\n", aux->fone); */\* terminal o telefone referente ao nome informado \*/***

**return; *// finaliza a função***

**}**

**}**

**printf("\n\tO nome %s nao foi encontrado\n", nome); */\* imprime no terminal que o nome não foi encontrado \*/***

**}**