

Lista de Exercícios

MC102 - Algoritmos e Programação de Computadores
Instituto de Computação
Universidade Estadual de Campinas

Modularização, Vetores, Matrizes e
Registros
2012
André Santanchè

Questão 1

Escreva uma função para realizar a comparação entre dois vetores contendo nomes. Ela recebe como parâmetros os ponteiros para ambos os vetores e retorna, através de parâmetros, os seguintes dados:

Parâmetro	Descrição
status	Possui valor: 0 - se ambas as listas forem exatamente iguais (inclusive na ordem); 1 - se as listas possuírem os mesmos nomes em ordens diferentes; 2 - se as listas forem diferentes.
diferencas	Consiste em um vetor que possui os nomes que fazem parte da primeira lista e não fazem da segunda. A função receberá este parâmetro com o vetor já criado e vazio (todas as posições nulas) e deve atualizá-lo.

Considere que as listas recebidas como parâmetro já estão preenchidas com nomes e que cada lista possui no máximo 100 nomes. A função não recebe o tamanho das listas como parâmetro, porém, após o último nome de cada lista há sempre uma célula do vetor nula.

Questão 2

Considere um polinômio de grau n :

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0 x^0$$

Considere a representação computacional de tal polinômio na forma de um vetor de registros (*structs*), mantido pelas seguintes funções:

inicializa	Recebe dois parâmetros : a_i e i , e cria um polinômio em memória na forma: $P(x) = a_i x^i$. A função retorna um ponteiro para o vetor contendo o polinômio criado. Utilize <code>malloc</code> para a alocação do vetor.
termo	Recebe três parâmetros: ponteiro para o vetor contendo o polinômio, a_i e i , e adiciona o termo $a_i x^i$ ao polinômio recebido como parâmetro. O polinômio pode ter um termo $a_q x^q$ cujo valor de q seja igual a i , neste caso a função deve unificar ambos em um único termo.
calcula	Recebe um valor de x como parâmetro e retorna o valor de $P(x)$.
finaliza	Libera a área de memória alocada para o polinômio usando a função <code>free</code> .
fusao	Recebe como parâmetros dois <i>handlers</i> para polinômios e retorna o <i>handler</i> de um novo polinômio que corresponda a fusão dos polinômios de entrada.

Exercício inspirado em exemplo dos slides de prof. Tomasz Kowaltowski : "Estruturas de Dados e Técnicas de Programação", 2010.

Questão 3

Dada a seguinte estrutura que representa um item de um um vetor contendo uma lista de compras, em que cada elemento é um item da lista com seu nome e quantidade:

```
typedef struct {
    char itemName[];
    int quantity;
} ElemType;
```

Implemente uma função em C para acrescentar um item em nesta lista de compras conforme a política:

- se o item passado como parâmetro já existir na lista (mesmo nome), o método deve adicionar a sua quantidade à do item já disponível na lista;
- se o item passado como parâmetro não existir na lista, ele deve ser adicionado na lista.

A função recebe como parâmetros: o ponteiro para o vetor contendo a lista de compras e um item a ser adicionado na lista (tipo `ElemType`).

Questão 4

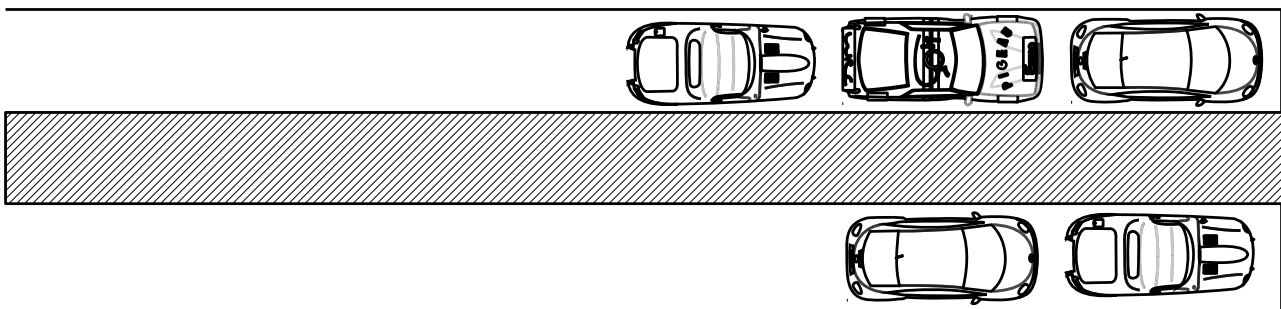
Dada uma lista de compras l (conforme especificado na questão anterior) e um valor o de orçamento, uma sublista de compras com orçamento limitado de l é uma sublista de l em que a soma dos valores gastos nos itens é menor ou igual a um valor de teto t .

Escreva uma função em C que receba como parâmetros uma lista de compras l e um valor de teto t e retorne uma possível sublista com orçamento limitado. A função deve considerar os itens pela ordem da lista, mas deve tentar alcançar o máximo de itens possível, portanto, se a soma de valor de um item da lista faz exceder o total, o método deve tentar incluir na lista um de seus subsequentes.

Considere que existe uma função já implementada que recebe como parâmetro o nome de um item de compra e retorna seu preço unitário.

Questão 5

Um estacionamento possui túneis de garagem que permitem estacionar duas filas de carros, como está ilustrado na figura abaixo:



Escreva uma função em C que implemente um gerenciador de garagem, conforme a especificação:

- a) A função recebe como parâmetro o ponteiro para um vetor representando uma fila já preenchida contendo placas de carros. Estes carros estão em uma fila externa aguardando para estacionar.
- b) A função inicia com dois vetores vazios, representando os túneis (túnel 1 e túnel 2).

- c) A função deve aguardar que o usuário digite um comando (um caractere). Ao digitar o comando, o programa executa uma das ações abaixo:

comando	ação
e	Estacionar - retira o primeiro carro da fila externa e o coloca sempre por último no túnel que tiver menos carros, ou no túnel 1 se ambos tiverem o mesmo número de carros.
d	Devolver o carro - pergunta pelo teclado a placa do carro a ser devolvido e o retira do respectivo túnel. Se houver carros atrás do que vai ser retirado, os carros que estão atrás dele precisarão ser movidos para o outro túnel antes que ele seja retirado. A mudança de túnel deve respeitar a ordem física de estacionamento (primeiro sai o último que está no fim da fila, depois o penúltimo e assim por diante). Cada vez que um carro muda de túnel no sistema, a função imprime na tela a mensagem indicando a placa do carro e o túnel para o qual ele mudou.

- d) O item (c) se repete até que a fila externa de carros esteja vazia.

Questão 6

Escreva um módulo que receba como parâmetro uma matriz MAT de 30 linhas por 20 colunas preenchida com nomes e um vetor VET1 de 30 elementos também preenchido com nomes, e gere um segundo vetor VET2 numérico (que retornará como parâmetro) onde cada elemento i seu, deve conter o número de vezes que o nome na posição i de VET1 aparece na linha i de MAT.

Por exemplo, se na primeira posição de VET1 aparece o nome ASDRUBAL, na primeira posição do vetor VET2 deve ser colocado o número de vezes que este nome aparece na primeira linha de MAT; se o nome DORIANA aparece na segunda posição de VET1, na segunda posição de VET2 será colocado o número de vezes que ele ocorre na segunda linha de MAT, e assim por diante.