Quinta Lista de Exercícios

Estruturas de Dados 1

Prof. Paulo Henrique Ribeiro Gabriel

Alocação Dinâmica, parte 2

1. Considere o trecho de código a seguir (o tipo `bool` está disponível na biblioteca `stdbool.h`):

```
struct vetor {
        float *elem;
        int total;
        int max;
};
typedef struct vetor Vetor;

Vetor *criar(int n, bool *deuCerto) {
        Vetor *v;
        v = (Vetor *) malloc(sizeof(Vetor));
        if (v == NULL) *deuCerto = false;
        else {
                v->elem = (float *) malloc(n * sizeof(float));
                if (v->elem == NULL) *deuCerto = false;
                else {
                    *deuCerto = true;
                    v->total = 0;
                    v->max = n;
                }
        }

        return v;
}

void destruir(Vetor *v) {
        free(v->elem);
        free(v);
}
```

Entenda o funcionamento dessas funções. Note que estamos alocando, dinamicamente, um *array* dentro de um registro. Em seguida, faça os seguintes exercícios:

(a) Crie uma função `inserir` que recebe um valor do tipo `float` e insere esse valor no *array*. Essa função deve verificar se é possível inserir o valor (ou seja, se o total de elementos é menor que o máximo alocado).

(b) Peça, na `main`, que o usuário digite o tamanho do *array* e forneça os valores. Use suas funções do TAD Vetor. Não se esqueça de destruir o vetor após o uso.

2. Refaça os TADs Pilha e Fila, vistos em aula, usando alocação dinâmica de *array*.

3. Faça um programa que:

   (a) Leia dois números N e M;

   (b) Crie e leia uma matriz $N \times M$ de inteiros;

   (c) Crie e construa uma matriz transposta $M \times N$.

   (d) Mostre as duas matrizes.

   Organize suas funções, adequadamente, em um TAD. **Dica:** Organize a própria matriz em um registro, similar ao vetor do exercício (1).

Para saber mais: Programação Descomplicada, aulas 60 a 65.