

Universidade Tecnológica Federal do Paraná (UTFPR)
Departamento Acadêmico de Informática (DAINF)
Estrutura de Dados I
Professor: Rodrigo Minetto

1) Quais as vantagens de se programar com TADs?

2) Suponha um fila com n elementos. Descreva o que acontece com esses n elementos após a execução do seguinte fragmento de código:

```
Stack *pilha = criar_pilha();
while (!vazia(fila)) {
    push(pilha, dequeue(fila));
}
while (!vazia(pilha)) {
    enqueue(fila, pop(pilha));
}
```

3) Suponha a seguinte estrutura de uma lista encadeada

```
typedef struct node {
    int info;
    struct node* next;
} Lista;
```

Escreva uma função que conta o número de nós de uma lista encadeada de forma iterativa e recursiva. O protótipo das funções é dado por:

```
int conta_nos_itr (Lista *lista);
```

```
int conta_nos_rec (Lista *lista);
```

4) Considerando a mesma estrutura de lista encadeada do exercício 3, substitua as instruções que faltam na função abaixo de tal forma a implementar uma **remoção** em uma lista ligada.

```

Lista* remover (Lista *lista, int elem) {
    Lista *prv = NULL; Lista *v = lista;
    while ((v != NULL) && (v->info != elem)) {
        1. _____;
        2. _____;
    }
    if (v == NULL) { return lista; }
    if (prv == NULL) { 3. _____; }
    else { 4. _____; }
    free (5. _____);
    return lista;
}

```

5) Considerando a mesma estrutura de lista encadeada do exercício 3, escreva uma função recursiva que imprime os elementos da lista em ordem reversa.

```
void imprime_descrescente_recursivo (Lista *l);
```

6) Utilizando o conceito de TAD-Pilha, escreva um programa para avaliar expressões na forma pós-fixa. Como exemplo, considere a seguinte expressão na forma pós-fixa

• 9 0 1 + 2 3 * * +

a sua função deve retornar como resultado o cálculo do valor da expressão, que é **15** (o resultado de $9 + ((0 + 1) * (2 * 3))$).

7) Escreva uma função que, dada uma lista duplamente encadeada com números em ordem aleatória, localize o menor elemento da lista, e troque-o com a cabeça da lista. Não vale trocar os campos info dos elementos, você deverá fazer a manipulação dos apontadores para trocar os nós de posição (teste com funções para imprimir os elementos para frente e para trás para verificar se os ponteiros foram ajustados corretamente).

```

typedef struct _lista {
    int info;
    struct _lista *prev;
}

```

```

    struct _lista *next;
} Lista;

Lista* move_menor (Lista *lista);

```

Entrada:

92 86 35 93 15 77 86 83

Saída:

15 86 35 93 92 77 86 83

8) Um algoritmo de ordenação é estável se a ordem relativa dos itens com chaves iguais mantém-se inalterada após a ordenação. Quais dos algoritmos de ordenação vistos são estáveis?

9) Qual o melhor caso do insertion sort, ou seja, para qual entrada o algoritmo realiza menos trocas de elementos? E o pior caso?

10) Qual a relação que existe entre o pivô do algoritmo Quick-Sort e o pior caso do algoritmo?

11) Dada uma permutação $p[1 \dots n]$, escreva uma função em C para determinar o número de inversões em p com complexidade $\mathcal{O}(n^2)$. O par (i,j) é uma inversão quando $i < j$ e $a_i > a_j$. Por exemplo no vetor $p = \{2, 4, 1, 3, 5\}$, existem 3 inversões, já no vetor $p = \{2, 1, 3, 1, 2\}$ existem 4 inversões. Para qual vetor p podemos ter um número máximo de inversões?

12) Dê um exemplo de uma entrada mínima que mostre a não estabilidade do algoritmo de ordenação por seleção.

13) Analise a complexidade de tempo da função Particione sobre um vetor de tamanho n .

14) Modifique o algoritmo Bubble-Sort para ordenar os elementos em ordem decrescente.

15) Escreva em C uma versão do algoritmo Bubble-Sort recursivo.