



## Lista de Exercícios de Fixação 1

### Algoritmos e Estruturas de Dados I

Prof. Rafael Fernandes Lopes

#### Ponteiros

---

1. Considere o seguinte programa

```
int main () {  
    int a, b,*p1, *p2;  
    a = 4;  
    b = 3;  
    p1 = &a;  
    p2 = p1;  
    *p2 = *p1 + 3;  
    b = b * (*p1);  
    (*p2)++;  
    p1 = &b;  
    printf("%d %d\n", *p1, *p2);  
    printf("%d %d\n", a, b);  
}
```

Usando a tabela abaixo, escreva os valores das variáveis **para cada instrução** do programa acima. Considere que as variáveis **a** e **b** estão armazenadas nos endereços **1000** e **2000**, respectivamente. Qual a saída do programa? [2,00]

Valores das variáveis			
a	b	p1	p2

2. O professor Rafael resolveu montar uma *Software House* e o seu primeiro projeto foi o desenvolvimento de um jogo de RPG online chamado BICT (*Bárbaros Indestrutíveis e Cavaleiros Triunfantes*). Para o desenvolvimento desse jogo ele resolveu recrutar você para ajudar no desenvolvimento do jogo! Vamos lá? :) [4,00]
- a) Sua primeira tarefa é modelar e implementar em C o TAD (Tipo Abstrato de Dados) “*Personagem*”. Um personagem deverá armazenar as informações relativas ao seu nome, nível, classe (um personagem pode ser um bárbaro ou um cavaleiro), sua posição em um mapa (considere que o mapa é bidimensional, distribuído em linhas e colunas) e quantidade de dinheiro. Além disso, esse TAD também deve armazenar os atributos do personagem: força, inteligência, sabedoria, destreza, constituição e carisma.
  - b) Para o jogo será necessário que o TAD forneça operações para criar, liberar (o jogo precisará que os personagens sejam alocados dinamicamente!), aumentar o nível, movimentar o personagem, definir a quantidade de dinheiro, consultar os atributos (não alterá-los!), consultar todos os seus atributos (nome, classe, posição dinheiro, atributos, etc.)
  - c) Implemente um programa que utilize o TAD.
3. Ainda no contexto do jogo BICT, foi solicitado que você implemente em C o TAD coleção “*mochila*”. A mochila vai ser usada para armazenar para armazenar itens do jogo. [4,00]
- a) Modele e implemente um TAD coleção “*mochila*”. Cada item da mochila é basicamente uma string que armazena o nome do item. A mochila deve ser capaz de armazenar até 10 itens. Se você achar mais fácil, pode implementar um tipo estruturado “*item*” para armazenar na mochila, mas é opcional.
  - b) Implemente as operações da mochila. Ela deverá fornecer operações para criar, liberar (o jogo precisará que as mochilas sejam alocadas dinamicamente!), inserir um item, consultar quantos itens a mochila tem e retirar/retornar um item da mochila. No entanto, tem um detalhe importante na mochila: sempre será retirado/retornado o item que foi inserido por último nela! Por exemplo, se forem inseridos, nessa ordem, os elementos “*espada*”, “*flauta*” e “*adaga*”, a operação de retirar os itens deve retornar os valores “*adaga*”, “*flauta*” e “*espada*”.
  - c) Implemente um programa que utilize o TAD.