

## MC102—Algoritmos e Programação de Computadores

Turmas M e N

Segundo semestre de 2009

## Lista de exercícios 05

1. Considere o trecho de programa abaixo. Depois de executado, quais são os valores associados aos itens de (a) a (g). Suponha que os endereços das variáveis *u* e *v* são 1000 e 1004 respectivamente.

```
int v, u;
int pv, pu;
v = 45;
pv = &v;
*pv = v + 1;
u = *pv + 1;
pu = &u;
```

(a) &v  
(b) pv  
(c) \*pv  
(d) u  
(e) &u  
(f) pu  
(g) \*pu

2. Considere o trecho de programa abaixo. Depois de executado, quais são os valores associados aos itens de (a) a (i). Suponha que os endereços das variáveis *a*, *b* e *c* são 1000, 1004 e 1008 respectivamente.

```
float a, b;
float c, *pa, *pb;
a = 0.001;
b = 0.003;
pa = &a;
*pa = 2 * a;
pb = &b;
c = 3 * (*pa + *pb);
```

(a) &a  
(b) &b  
(c) &c  
(d) pa  
(e) \*pa  
(f) &>(\*pa)  
(g) pb  
(h) &>(\*pb)  
(i) c

3. Dado *p* um ponteiro qualquer, qual a diferença entre *p++*, *(\*p)++* e *\*(p++)*?

4. Implemente as funções abaixo, considerando a definição da estrutura e o protótipo das funções. *cria\_lista* deve alocar um vetor do tipo `struct Aluno` com *num\_alunos* elementos, e retornar um apontador para este vetor. *preenche\_dados* recebe um apontador para uma lista de alunos, e deve preencher o elemento *i* com o RA e nome dado.

```
struct Aluno { int RA; char nome[80]; }
```

```
struct Aluno *cria_lista(int num_alunos);
```

```
void Preenche_dados(struct Alunos *lista, int i, int novoRa, char novoNome[80]);
```

5. Escreva a função `le_dados_do_teclado()` que lê um conjunto de *n* inteiros da entrada padrão e escreve em um arquivo texto. O número de inteiros lidos da entrada padrão deve ser informado pelo usuário, mas não deve ser gravado no arquivo.

6. Escreva a função `escreve_dados_na_tela()` que lê um conjunto de inteiros de um arquivo texto e escreve na saída padrão. O número de inteiros lidos do arquivo deve ser informado pelo usuário.
7. Altere os dois exercícios anteriores para que eles operem sobre arquivos binários, ao invés de arquivos texto.
8. Escreva uma função que, dado um arquivo binário composto de vários registros do tipo `Registro`, de tamanho `sizeof(Registro)`, faça a leitura de todos os registros desse tipo para um vetor `v`, do tipo `Registro`, passado como parâmetro e que possui um tamanho grande o suficiente para que caibam todos os registros existentes naquele arquivo. É necessário conhecer os campos do tipo `Registro` para escrever esta função?
9. Escreva uma função que, dado um arquivo binário composto de vários registros do tipo `Registro`, de tamanho `sizeof(Registro)`, e um inteiro `n` passado como parâmetro, faça a leitura do `n`-ésimo registro existente no arquivo binário, ou retorne um registro vazio caso essa posição não exista no arquivo.
10. Escreva a função `imprime_grafico_no_arquivo`, que dado um vetor, passado como parâmetro, armazena em um arquivo um gráfico de barras com o seu conteúdo. O cabeçalho da função deve ser `void imprime_grafico(char nome_arquivo[], int vetor[], int tamanho);`. Cada linha do arquivo contém o número `n` a ser impresso e uma barra que contém `n` vezes a letra "X". Por exemplo, para o vetor 5,6,2,4, teríamos o seguinte gráfico armazenado no arquivo:

```
5|XXXXX
6|XXXXXX
2|XX
4|XXXX
```

11. Escreva uma função que leia `n` inteiros da entrada padrão e armazene-os em uma lista ligada simples de inteiros.
12. Escreva uma função que escreva todo o conteúdo de uma lista ligada simples de inteiros na saída padrão.
13. Escreva um programa que ordene os elementos de uma lista ligada simples de inteiros. Utilize o método de ordenação que você considerar mais conveniente.
14. **Pilhas:** Pilhas são listas de elementos onde o elemento removido é sempre aquele que foi inserido mais recentemente. Escreva duas funções que manipulam pilhas
  - `int insere_pilha(int n)`: insere o elemento `n` no topo da pilha, retornando 1, ou retorna 0 se a pilha estiver cheia.
  - `int remove_pilha()`, Remove o elemento do topo da pilha, ou retorna -1 se a pilha estiver vazia.

Você deve fazer duas implementações: uma utilizando um vetor para armazenar os dados da pilha (utilize um inteiro para armazenar a posição do topo da pilha) e outra utilizando listas ligadas (utilize um ponteiro para armazenar o topo da pilha. Como seriam realizadas as operações de inserção e remoção?)

15. Um polinômio pode ser representado por uma lista ligada, onde cada nó representa um termo do polinômio, com seu coeficiente e seu expoente, além do apontador para o próximo elemento do polinômio. A estrutura de cada nó segue abaixo:

```
struct polinomio {float coef; float expo; struct polinomio* prox;};
typedef struct polinomio Polinomio;
```

Escreva o corpo da função `imprime_polinomio`, que imprime o conteúdo do polinômio na tela. Considere que não há termos nulos (ou seja, que o coeficiente é sempre diferente de zero) e que os expoentes não se repetem ao longo da lista ligada. O cabeçalho da função é:

```
void imprime_polinomio(Polinomio *p)
```