Instituto Federal de Educação, Ciência e Tecnologia de Brasília – Câmpus Taguatinga
Ciência da Computação – Programação de Computadores I
Lista de Exercícios – Ponteiros
Prof. Daniel Saad Nogueira Nunes

Aluno: _____
Matrícula: _____

Exercício 1

Quais os valores de x, y e p ao final do seguinte código:

```
int x, y, *p;
y = 0;
p = &y;
x = *p;
x = 4;
(*p)++;
--x;
(*p) += x;
```

Exercício 2

Os programas a seguir apresentam erros. Faça alterações de modo a corrigi-los.

(a)
```
int main(void){
        int x,*p;
        x = 100;
        p = x;
        printf("O valor de p: %d.\n",*p);
    }
```

(b)
```
void troca(int* i, int * j){
        int* tmp;
        *tmp = *i;
        *i = *j;
        *j = *tmp;
    }
```

Exercício 3

Faça um programa que leia um inteiro n e crie um vetor de n números reais. O vetor deve ser alocado de maneira dinâmica.

Exercício 4

Detalhe a organização da memória de um programa.

Exercício 5

Se `C` não possui passagem por referência, explique detalhadamente como é possível emulá-la através de ponteiros.

Exercício 6

Como uma variável do tipo ponteiro em `C` é passada por referência?

Exercício 7

Crie uma função `min_max` que recebe um vetor de inteiros de n elementos e retorne as posições dos valores mínimo e máximo do vetor através de dois parâmetros inteiros `l` e `r` passados por "referência" para função. Em caso de empate, a função deverá considerar a posição mais à esquerda possível. Sua função deverá possuir a seguinte assinatura:

```
void min_max(int* v,int n, int* l,int* r);
```

Exercício 8

Crie uma função que receba n notas de um aluno e retorne a sua situação. A função também deverá calcular a média e armazená-la em uma das variáveis passadas por "referência". A situação do aluno deverá ser 'A' em caso de aprovação, 'R' em caso de reprovação. Considere que a média para aprovação é 6.0. Sua função deverá possuir a seguinte assinatura:

```
char calcula_situacao(double* v, int n, double* media);
```

Exercício 9

Implemente uma função `getline` que leia uma linha inteira e retorne uma string com os caracteres lidos independente do número de caracteres. A string deverá ser alocada dinâmicamente de modo a não desperdiçar espaço. Sua função deverá possuir a seguinte assinatura:

```
char* getline(void);
```

Dica: para esta função é interessante utilizar as funções `getchar()` e `realloc()`.

Exercício 10

Crie um vetor de inteiros redimensionável. As seguintes operações devem atuar sobre o vetor:

- `void push_back(int* v,int* n,int* capacidade,int valor)`: insere o conteúdo de `valor` em `v` e atualiza o seu tamanho `n`. Caso o vetor chegue na sua `capacidade`, ela deve ser dobrada.
- `void print(int* v,int n)`: imprime o vetor `v`.
- `int pop_back(int* v,int* n,int* capacidade)`: retorna o último elemento do vetor e o retira do mesmo, decrementando o tamanho dele. No caso de o vetor possuir $\frac{1}{4}$ da sua capacidade máxima, ela deve ser reduzida pela metade.

Exercício 11

Faça um programa que leia dois inteiros n e m e aloque uma matriz $A_{n \times m}$ de maneira dinâmica.