



Exercício – Ponteiros

Profa. Simone Aires – Prof. Saulo

1. Para cada problema listado a seguir, apresente um código fonte que os ilustra. Explique brevemente cada problema e como eles ocorrem no código apresentado:

- (a) leitura ilegal;
- (b) escrita ilegal;
- (c) falha de segmentação com ponteiros;
- (d) vazamento de memória;

2. No contexto da temática de ponteiros na linguagem C, quais os possíveis significados para a ocorrência do caractere * em um código fonte? Exemplifique.

3. Qual o tipo mais adequado para a variável p2 no código abaixo?

```
#include <stdlib.h>

int main() {

int *p = (int *) malloc(2*sizeof(int));

??? p2 = &p;

return 0;

}
```

4. Sejam duas variáveis inteiras int a e int b declaradas na função int main(). Apresente um procedimento em C capaz de trocar o valor de a pelo seu quadrado e o valor de b pelo seu fatorial. Apresente as suposições que julgar necessário.

5. Supondo que na linha 04 do código abaixo seja atribuído o valor 0x949b008 a p, qual o valor resultante das impressões das linhas 05 e 06?

Explique como a execução se comportará em cada caso.

```
#include<stdlib.h>

int main()

{

*p=(int *) malloc(2*sizeof(int));

printf("%p\n",p+1);

printf("%p\n",(char*)p+1);

return 0; }
```



6. Dado que variáveis do tipo *unsigned char* comportam um numero natural entre 0 e 255 (cada um dos quais representa um dos 256 caracteres da tabela ASCII basica) e que variáveis do tipo *unsigned short int* empregam 2 bytes para representar números naturais {0, 1, 2, . . .}, é possível prever o resultado da impressão abaixo? Em caso negativo justifique, do contrário apresente o valor no formato que desejar.

```
#include <stdlib.h>

int main() {

    unsigned char a[2] = {255,255};

    unsigned short int *b = (unsigned short int *) a;

    printf("%d\n", *b);

    return 0;

}
```

7. Durante o desenvolvimento de um projeto um colega seu constatou a frequente necessidade de se liberar memória a partir de variáveis do tipo ponteiro para int. Por esse motivo ele resolveu propor o procedimento abaixo e afirmou ter tomado os cuidados básicos necessários para resolver o problema mencionado. A solução proposta alcança o objetivo? Em caso afirmativo explique o porque de cada comando. Em caso negativo reescreva o procedimento incluindo as modificações necessárias para que ele funcione adequadamente.

```
#include

void Desaloca(int *p)

{

    free(p);

    p = NULL;

}
```

8. Escreva uma função mm que receba um ponteiro para um vetor inteiro v[0..n-1], o número de elementos do vetor (n) e os endereços de duas variáveis inteiras, *min* e *Max*. Deposite nessas variáveis o valor de um elemento mínimo e o valor de um elemento máximo do vetor. Escreva também uma função *main* que use a função mm.

9. Crie uma função que receba uma string como parâmetro (de tamanho desconhecido) e retorne uma cópia da mesma. A assinatura da função deve ser: char *strcpy(char *str);

10. Reescreva as funções **strlen**, **strcpy**, **strlwr**, **strupr**, **strchr** e **strcat** da biblioteca <string.h> usando ponteiros.

11. Escreva um programa que a partir da leitura de duas Strings qualquer, informe o número de caracteres iguais contidos nas duas Strings.

12. Escrever um procedimento que faça a União de dois vetores dinâmicos ($C = A \cup B$). Lembre-se de **liberar a memória dos vetores A e B após a concatenação**.



13. Fazer um programa contendo as seguintes funções:

- Leitura: lê um vetor de números inteiros positivos
- Separa: **separa os elementos do vetor em dois vetores**, sendo que um deles conterá os números **pares** e o outro os números **ímpares**. Todos os vetores dinâmicos. Não esqueça de **liberar memória quando possível**.
- Imprime: **exibe na tela os elementos contidos no vetor passado por parâmetro**.

14. Para o código abaixo comente o que cada linha de código faz:

```
#include<stdio.h>
#include <alloc.h>
int main ()
{
    int y, x=3, k=5, *p,*q;
    p = &x;
    q = (int *)malloc(6 * sizeof(int));
    for(y = 0; y< 3; y++)
    {
        p--;
        scanf("%d", p);
        p++;
        scanf("%d", p);
        *q = k;
        *(q+1) = x;
        q = q + 2;
    }
}
```