Programação e Algoritmos
Lista de Exercícios – Ponteiros

Prof. Ana Luiza Bessa de Paula Barros
Ciência da Computação – UECE

1) Você foi contratado para realizar a consultoria de um sistema bastante antigo e se deparou com o código:

```
#include <stdio.h>
int main() {
   int y, *p, x;
   y = 0;
   p = &y;
   x = *p;
   x = 4;
   (*p)++;
   x--;
   (*p) += x;
   printf("y = %d\n", y);
   return 0;
}
```

Seus contratantes não sabem se a resposta que o programa está retornando é correta, pois desconhecem qual é o retorno esperado. Sendo assim, eles pediram que você analise o programa e informe qual seria o retorno esperado.

2) Na documentação desse código, há o uso dos termos "passagem de parâmetros por referência/por valor" para algumas funções. Seu contratante pediu que você explique o que são ambas e qual a diferença, além de pedir uma demonstração prática por meio da criação de uma função simples de potenciação usando ambas as abordagens.

3) Na documentação são feitas várias referências aos termos na tabela abaixo. Os contratantes pediram que você faça uma breve explicação de cada um deles:

- Um ponteiro
- Um ponteiro para ponteiro
- O endereço de um ponteiro
- O conteúdo de um ponteiro
- O endereço de uma variável

4) Nesse sistema, há uma função que está apresentando problemas e os contratantes pediram que você a refaça. Essa função deve receber por parâmetro dois inteiros A e B e realizar a soma de ambos, armazenando o resultado na variável A. No entanto, devido às limitações do sistema, essa função não pode ter nenhum retorno, sendo necessário o uso de lógica de ponteiros. Além disso, na função 'main', deve-se ler A e B do teclado e imprimir o valor atualizado de A e B após o uso da função criada, a fim de testá-la.

5) O sistema está enfrentando problemas com a memória e precisa de um programa para ajudar a resolver. Após a leitura de 2 variáveis do teclado, a variável com o maior endereço de memória sempre está incorreta. Sendo assim, foi pedido que você faça um programa que, ao comparar essas 2 variáveis após a leitura, informe qual delas possui o maior endereço e qual é esse endereço.

6) A fim de testes do sistema, foi pedido que você faça 2 vetores de pontos flutuantes de tamanho 10, utilizando 2 abordagens: alocação estática e alocação dinâmica. Após a criação, imprima o endereço de memória de cada elemento de ambos os vetores. Não se esqueça de que a memória dinâmica deve ser liberada após o uso.

7) Desejando verificar se o resultado será diferente para uma matriz em vez de um vetor, realize o mesmo procedimento feito no problema anterior, mas desta vez utilizando uma matriz 3x3 de pontos flutuantes com as 2 abordagens diferentes.

8) Para lidar com o bug estranho no sistema, onde apenas os números pares de um vetor lido pelo teclado estão incorretos, o contratante pediu que você faça um pequeno programa que tenha um vetor de 5 inteiros com números lidos pelo usuário. Após a leitura, imprima apenas os endereços de memória dos números pares.

9) Para atender à demanda de modificar uma função no sistema para que ela receba os parâmetros por referência, foi solicitado que você refaça essa função que inverte um número dado (231 vira 132) com essa nova exigência.

10) O sistema roda em uma máquina antiga e com diversos outros programas rodando ao mesmo tempo. Sendo assim, toda eficiência é importante. No entanto, uma função não está sendo a mais eficiente possível. Foi pedido que você refaça essa função utilizando alocação dinâmica. Essa função deve ler um número N, criar dinamicamente um vetor com esse tamanho N, preenchê-lo com dados do usuário e, em seguida, ordená-lo e imprimi-lo.

11) Uma outra função no sistema que deve ser atualizada para utilizar ponteiros é aquela que transforma letras minúsculas em um vetor em letras maiúsculas. Foi pedido que você pesquise e estude sobre a função em C que transforma um caractere minúsculo em maiúsculo.