LISTA I (REVISÃO)

Enos Andrade Diniz Sousa

1 Qual o valor de x, y e p no final da execução desse trecho de código?

```c
#include <stdio.h>
int main(int argc, char const *argv[])
{
    int x, y, *p;
    y = 0;
    p = &y;
    x = 4;
    (*p)++;
    x--;
    (*p) += x;

    return 0;
}
```

Solução

```c
#include <stdio.h>

int main(int argc, char const *argv[])
{
    int x, y, *p;
    y = 0;
    p = &y; // endereço de y atribuido a p
    x = 4;
    (*p)++; //Conteúdo apontado por *p é incrementado em 1, ou seja, y que valia 0
     ↪  passa a valer 1
    x--; // x é decrementado e passa a valer 3
    (*p) += x; /* valor apontado por p(y) é incrementado no valor de x (y = y + x),
                  ou seja, y valia 1 foi somado com x(3) e passa a valer 4*/

    //RESPOSTA: x = 3, y = 4, e p = endereço de y

    return 0;
}
```

2 Qual é o resultado da execução desse programa?

```c
#include <stdio.h>

void imprime_primeiro(int *vet){
    printf("Valor: %d\n", vet[0]);
}
int main(int argc, char const *argv[])
{
    int vet[5] = {1, 2, 3, 4, 5};
```

```
    imprime_primeiro(vet);
    return 0;
}
```

Solução

O programa imprime o primeiro dado armazenado no vetor vet, ou seja, 1.

Obs: Apesar de o operador não esta sendo usado, um array é sempre um ponteiro para o primeiro dado do mesmo, logo o parâmetro para a função imprime_primeiro() está sendo passado por referência.*/

3 Qual é o resultado da execução desse programa?

```c
#include <stdio.h>

void imprime_primeiro(int *vet){
    printf("Valor: %d\n", vet[0]);
}

int main(int argc, char const *argv[])
{
    int vet[5] = {1, 2, 3, 4, 5};
    imprime_primeiro(&vet[2]);
    return 0;
}
```

Solução

O programa imprime o terceiro dado armazenado no vetor vet[], ou seja, 3.

4 Qual é o resultado da execução desse programa?

```c
#include <stdio.h>

int* metade_final(int *vet, int n){
    return &vet[(int)(n/2)];
}

int main(void)
{
    int vet[6] = {1, 2, 3, 4, 5,6};
    int *v = metade_final(vet, 6);
    printf("Valor: %d\n", v[0]);
    return 0;
}
```

Solução

O programa imprimirá o primeiro número da segunda metade da lista, no caso: 4.

5 Qual é o resultado da execução desse programa?

```c
#include <stdio.h>

int main(void)
{
    int vet[6] = {1, 2, 3, 4, 5,6};
    printf("Valor1: %d\n", vet);
    printf("Valor2: %d\n", *vet);
    printf("Valor3: %d\n", *(vet+2));
    return 0;
}
```

Solução

```c
#include <stdio.h>

int main(void)
{
    int vet[6] = {1, 2, 3, 4, 5,6};
    printf("Valor1: %d\n", vet); //Será impresso o endereço de memoria apontado por
     ↪  ver
    printf("Valor2: %d\n", *vet); //Será impresso o primeiro dado armazenado no
     ↪  vetor vet: 1
    printf("Valor3: %d\n", *(vet+2)); //Será impresso o terceiro dado armazenado no
     ↪  vetor vet: 3
    return 0;
}

//RESPOSTA: Será impresso, respectivamente: endereco de vet, 1 e 3.
```

6 Implemente uma função que busca por um inteiro em um vetor de inteiros.

Solução

```c
#include <stdio.h>

int main(int argc, char const *argv[])
{
    int busca;
    int vetor[] = {12, 43, 15, 38, 1, 44, 9, 45, 34, 35, 42, 23, 3, 4, 2, 33, 17,
     ↪  20, 40, 27}; //vetor pre definido
    printf("Buscar o numero: ");
    scanf("%d", &busca);
    for(int i = 0; i < sizeof(vetor)/sizeof(int); i++){
        if (busca  == vetor[i]) {
            printf("O numero buscado esta na posicao %d do vetor", i);
            return;
        }
    }
    printf("Esse numero nao esta na lista");

```

```
    return 0;
}
```

7 Implemente uma função que busca por um inteiro em um vetor de inteiros. Retornando a última ocorrência.

Solução

```c
#include <stdio.h>

int main(int argc, char const *argv[])
{
    int busca;
    int posicao = 0;
    int contador = 0;
    int vetor[] = {12, 43, 15, 38, 1, 44, 9, 45, 34, 35, 42, 23, 3, 4, 2,
     33, 17, 20, 40, 27, 12, 45, 12, 15, 44, 42, 34}; //vetor pre definido

    printf("Buscar o numero: ");
    scanf("%d", &busca);
    int i;
    for(i = 0; i < sizeof(vetor)/sizeof(int); i++){
        if (busca  == vetor[i]) {
            contador++;
            posicao = i;
        }
    }
    if (posicao == 0) {
        printf("Esse numero nao esta na lista");
    }else{
        printf("O numero buscado aparece %d vezes na lista\nA ultima ocorrencia
          ↪  esta na posicao %d do vetor", contador,i);
    }
    return 0;
}
```

8 Criar uma função para encontrar o valor máximo em um vetor.

Solução

```c
#include <stdio.h>

int main(int argc, char const *argv[])
{
    int maximo;
    int vetor[] = {12, 43, 15, -456, 38, 1, 44, 9, 45, 34, 35, 42, 23, 3, 4, 2,
     33, 17, 20, 1042, 27, 12, 1500, 12, 15, 44, 42, 100, 127, 0, -200}; //vetor
      ↪  pre definido

    for(int i = 0; i < sizeof(vetor)/sizeof(int); i++){
        if (i  == 0) {
            maximo = vetor[i];
        }else{
```

```
            if (vetor[i] > maximo) {
                maximo = vetor[i];
            }
        }
    }
    printf("O valor maximo do vetor e %d.", maximo);

    return 0;
}
```

9 Criar uma função para encontrar o valor mínimo em um vetor.

Solução

```
#include <stdio.h>

int main(int argc, char const *argv[])
{
    int minimo;
    int vetor[] = {12, 43, 15, -456, 38, 1, 44, 9, 45, 34, 35, 42, 23, 3, 4, 2,
     33, 17, 20, 1042, 27, 12, 1500, 12, 15, 44, 42, 100, 127, 0, -200}; //vetor
      ↪  pre definido

    for(int i = 0; i < sizeof(vetor)/sizeof(int); i++){
        if (i  == 0) {
            minimo = vetor[i];
        }else{
            if (vetor[i] < minimo) {
                minimo = vetor[i];
            }
        }
    }
    printf("O valor minimo do vetor e %d.", minimo);

    return 0;
}
```

10 Faça um programa para ler e armazenar um conjunto de números em um vetor (máximo 50 números e a leitura de um número igual a zero indica fim da leitura dos dados). A seguir peça para o usuário digitar um valor inteiro e informe se o mesmo pertence ou não ao conjunto de números armazenados. O programa deve implementar e usar a função busca, que recebe como parâmetro um vetor de números inteiros (vet) de tamanho n e um valor x. A função deve retornar 1 se x pertence a esse vetor e 0 caso contrário. A função deve obedecer ao seguinte protótipo:

int busca(int vet[], int n, int x);

Solução

```
#include <stdio.h>

int main(int argc, char const *argv[])
```

```
{
    int busca(int vet[], int n, int x);
    int vetor[50];
    //ler valor do vetor
    int valorDigitado;
    for(int i = 0; i < 50; i++)
    {
        printf("\nValor %d: ", i + 1);
        scanf("%d", &valorDigitado);
        if (valorDigitado == 0) {
            break;
        }else{
            vetor[i] = valorDigitado;
        }
    }
    printf("\nBuscar o numero: ");
    scanf("%d", &valorDigitado);
    int isNumberInList = busca(vetor, 50, valorDigitado);
    if (isNumberInList == 1) {
        printf("O numero %d esta na lista", valorDigitado);
    }else{
        printf("O numero %d nao esta na lista", valorDigitado);
    }
    return 0;
}
int busca(int vet[], int n, int x){
    for(int i = 0; i < n; i++){
        if (x == vet[i]) {
            return 1;
        }

    }
    return 0;
}
```

11 Faça um programa para ler uma matriz digitada pelo usuário (tamanho 3x3) e exibir uma mensagem dizendo se a mesma é uma matriz identidade ou não. O seu programa deve implementar e utilizar a função matriz_identidade, que recebe como parâmetro uma matriz quadrada de inteiros de dimensão n, e retorna 1 se a matriz for uma matriz identidade, e 0 caso contrario. A função deve obedecer ao seguinte protótipo:

int matriz_identidade(int mat[][N], int n);

Lembre-se que uma matriz é considerada identidade quando os elementos da diagonal principal são todos iguais a 1 e os elementos restantes são iguais a zero.

Solução

```
#include <stdio.h>

int N = 3; // Dimenssão da matriz
int main(int argc, char const *argv[])
{
    int matriz[3][3];
```

```
    printf("Preencha a matriz:\n");
    for(int i = 0; i < 3; i++){
        for(int j = 0; j < 3; j++){

            printf("(%d, %d): ", i+1, j+1);
            scanf("%d", &matriz[i][j]);
        }
        printf("\n");
    }
    int matriz_identidade(int mat[][N], int n);

    int eMatrizIdentidade = matriz_identidade(matriz, 3);

    if (eMatrizIdentidade) {
        printf("\nE uma matriz identidade!");
    }else
    {
        printf("Nao e uma matriz identidade");
    }
}
int matriz_identidade(int mat[][N], int n){
    for(int i = 0; i < n; i++ ){
        for(int j =0; j < n; j++){
            if (i!=j && mat[i][j]!=0) {
                return 0;
            }
            if (i==j && mat[i][j]!=1) {
                return 0;
            }
        }
    }
    return 1;
}
```