

## DCA1202 - Programação Avançada (1ª avaliação - 2018.1)

Aluno: \_\_\_\_\_ Matrícula: \_\_\_\_\_

1. Um desenvolvedor precisa enviar as informações para um painel indicam se os leds de uma matriz devem estar acesos ou apagados. Para isso, criou um array de inteiros com 8 linhas e 32 colunas, do mesmo tamanho do matriz, e associou aos elementos os valores “1” ou “0” conforme os leds correspondentes devessem estar acesos ou apagados, respectivamente.

Em testes realizados, verificou-se que a taxa de transferência dos bits não era rápida o suficiente para que enviasse os elementos da matriz um a um na velocidade necessária para fazer o sistema operar corretamente.

Uma ideia surgiu: codificar grupos de oito inteiros de cada coluna em um único inteiro (unsigned char), transmiti-lo para o painel e decodificá-lo no receptor. E a ideia funcionou muito bem! O desenvolvedor criou duas funções `transmitArray()` que prepara o array para ser enviados ao destino, e `decodeArray()` que trata o array de bytes recebidos e restaura a matriz original. examine.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define NCOLUNAS 32

unsigned char sentArray[8][32] = {
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
    {0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0},
    {0,0,0,0,0,0,0,1,1,1,0,0,0,0,1,1,1,0,0,0,0,1,1,1,0,0,0,0,0,0,0},
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
    {0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0,0},
    {0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0},
    {0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0},
    {0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0},
    {0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0}};

unsigned char receivedBytes[NCOLUNAS];

unsigned char receivedArray[8][NCOLUNAS];

void transmitByte(unsigned char byte);

void encodeArray(unsigned char dados[8][32]){
    /* insira sua solucao aqui */
}

void decodeArray(unsigned char array[NCOLUNAS]){
    /* insira sua solucao aqui */
}

void transmitByte(unsigned char byte){
    static int index=0;
    receivedBytes[index]=byte;
```

```

    index++;
}

void printReceived(){
    int i, j;
    for(i=0; i<8; i++){
        for(j=0; j<NCOLUNAS; j++){
            printf("%d",receivedArray[i][j]);
        }
        printf("\n");
    }
}

int main(void){
    encodeArray(sentArray);
    decodeArray(receivedBytes);
    printReceived();
}

```

Implemente as funções `encodeArray()` e `decodeArray()` para que as etapas de codificação e decodificação da matriz associada ao painel de leds funcione corretamente.

2. Um programador criou uma função em C com o objetivo de que ela retornasse um array de inteiros previamente inicializado que pudesse ser usado pela sua função principal. Tais funções auxiliar e principal são mostradas na listagem a seguir.

```

#include <stdio.h>
#include <stdlib.h>

/* funcao auxiliar criada pelo programador:
   cria um array de 10 inteiros e retorna um ponteiro para este array
*/
int* criar_array(void){
    int x[10];
    int i;
    for(i=0; i<10; i++)
        x[i] = i;
    return x;
}

/* função principal */
int main(void){
    int *y, i;
    y = criar_array();
    for(i=0; i<10; i++)
        y[i]=10-i;
    for(i=0; i<10; i++)
        printf("%d\n", y[i]);
    free(y);
}

```

Analise a função `criar_array()` apresentada e responda o que se pede:

- a) Identifique, se existirem, erros de programação que poderão invalidar a utilidade da função criada, JUSTIFICANDO o por quê do erro identificado.
- b) Caso tenha identificado algum erro na função auxiliar proponha uma correção adequada, sem alterar seu protótipo nem a função `main()`, que resolva este problema.

As questões têm igual valor. Submeta sua solução no SIGAA. **A cópia é proibida, sob pena de nulidade do exame. A interpretação das questões faz parte do exame.**

=== MODELO DE RELATORIO DE PROVA EM TXT ===

Aluno:

== questao 1 =====

bla bla bla

== questao 2 =====

bla bla bla