```c
// Bibliotecas Utilizadas
#include <stdio.h> // Para entrada e saída padrões
#include <stdlib.h> // Biblioteca padrão

// Dados de Produção
/*
    Clientes:
        Universidade Estadual do Maranhão
        Centro de Ciências Tecnológicas
        Departamento de Engenharia da Computação
        Curso de Engenharia da Computação
        Disciplina: Estrutura de Dados Básica (ASL092N321)
        Semestre: 2022.2         Turma: 01
        Professores:
            Lúis Carlos Fonseca
            Pedro Brandão Neto

    Autores:
        Alunos:
            Alexsandro Lucena Mota   Código: 20210024710

    Propósito do Programa:

        Implementar uma função miniMax(v,n,a,b) que receba um vetor v,
        contendo n números, e devolva a e b, respectivamente, os valores
        mínimo e máximo entre aqueles armazenados em v.

    Dados de Manutenção do programa

      Data            Programador                  Descrição da Mudança
    ========    ======================      ================================
    2022/4/9    Alexsandro Lucena Mota      - Código original (versão 0.1).
    2022/5/9    Alexsandro Lucena Mota      - Acrescimento de comentário para
                                              facilitar a manutenção do programa.
    2022/5/9    Alexsandro Lucena Mota      - Adaptando o programa para alocação
                                              dinâmica.
    2022/5/9    Alexsandro Lucena Mota      - Implementado o teste de alocação de memória;
                                              Implementado a liberação de memória;
                                              Implementado, após liberação de momória alocada,
                                              a atribuição NULL aos ponteiros para evitar que
                                              fiquem soltos.
*/

// Lista de Protótipos de Funções e Procedimentos
int miniMax();

// Escopo da Função Main
int main(){
    // Apresentação do programa ao usuário
    printf("PROGRAMA miniMax\n\n");
    printf("Propósito:\n");
    printf("\tRecebe um vetor de dimensão n e retorna o me-\n");
    printf("\tnor e o maior valor entre suas componentes.\n\n");

    // Entrada de dados
    printf("Entre com a dimensão do vetor!\n");
    int dim = 0; // incializa a variável dim com valor 0.
    printf("DIM = ");
    scanf("%d", &dim); // usuário informa a dimensão do vetor (array).
    printf("Entre com as componentes (valores inteiros) do vetor!\n");
    int *vetor; // ponteiro vetor (aponta para um endereço de memória.
    vetor = (int *) malloc (dim*sizeof(int)); /* - alocando memória para vetor;
                                                 - o vetor é do tipo inteiro
                                                 - sizeof(int) retorna o quantidade
                                                   de bytes ocupados pelo tipo int: 4bytes.
                                                 - dim dar o número de componentes do vetor;
```

```c
                                            dim multiplica o sizeof(int) provendo o
                                            número de bytes necessário para alocar o
                                            vetor
                                        */
    if(vetor == NULL){ // teste de alocação de memória
        printf("Erro: Memória Insuficiente!\n");
        exit(1);
    }
    for(int i = 0; i < dim; i++){
        scanf("%d", vetor + i); // usuáro informa os valores de entrada do vetor
    }

    // Ecoando as informações entrada
    printf("As componentes informadas foram:\n");
    for(int i = 0; i < dim; i++){
        printf("v[%d] = %d\n", i + 1, *(vetor + i));
    }

    // Saída de dados
    int *return_vector; // para passagem de parâmentos de referência na função miniMax
    return_vector = (int *) malloc (2*sizeof(int)); // retornará somente duas compontentes
    if(return_vector == NULL){ // teste de alocação de memória
        printf("Erro: Memória Insuficiente!\n");
        exit(1);
    }
    miniMax(vetor,dim,return_vector);
    free(vetor); // liberando a memória alocada
    vetor = NULL; // para evitar que o ponteiro fique solto
    printf("O menor valor entre as componentes é: %d.\n", *return_vector);
    printf("O maior valor entre as componentes é: %d.\n", *(return_vector + 1));
    free(return_vector); // liberando a memória alocada
    return_vector = NULL; // para evitar que o ponteiro fique solto

    // Retorno da função
    return 0;
}

// Escopo das funções e dos procedimentos
int miniMax(int*in_vector,int num, int*out_vector){
    /*
        Função miniMax

        Propósito:
            Recebe um vetor, sua dimensão, e retornar o menor
            e menor valor (inteiros) entre suas componentes.
    */

    // Atribuições necessárias
     int minnum = *in_vector; // incializa a variável local minnum
     int maxnum = *in_vector; // incializa a variável local maxnum

     // Rotina para determinar o menor e o maior valor
     for(int i = 1; i < num; i++){
        if( *(in_vector + i) > maxnum){
            maxnum = *(in_vector + i);
        }else if( *(in_vector + i) < minnum){
            minnum = *(in_vector + i);
        }
     }

     /*
        Atribuido as saídas aos respectivos endereços de memória
        para acesso externo a função miniMax
    */
     *(out_vector + 0) = minnum;
     *(out_vector + 1) = maxnum;
```

```
133
134         // Retorno da função
135         return 0;
136 }
```