Atividade 4 – Ordenação de Pesos - Daniel de Oliveira Gomes

Código Fonte:

#include <stdio.h>

#include <locale.h>

#define maxpacotes 10 //define o número máximo de pacotes que podem ser inseridos.

//Protótipos das funções

void exibirarray(float *arr*[], int *n*);

void ordenarcrescente(float *arr*[], int *n*);

void ordenardecrescente(float *arr*[], int *n*);

int main() {

    setlocale(LC\_ALL, "pt\_BR.UTF-8"); //Possibilita o uso de acentos para língua portuguesa.

    float pesos[maxpacotes];

    int i;

    // Leitura dos pesos

    printf("Por favor, insira os pesos para os pacotes:\n");

    for (i = 0; i < maxpacotes; i = i + 1) {

        printf("Dê entrada no valor do peso do pacote %d em kg: ", i + 1);

        scanf("%f", &pesos[i]);

    }

    printf("\nSegue a lista de pesos para os pacotes:\n");

    exibirarray(pesos, maxpacotes);

    // cópias dos dados do array original

    float pesoscrescente[maxpacotes];

    float pesosdecrescente[maxpacotes];

    for (i = 0; i < maxpacotes; i = i + 1) { //passagem de valores do array original para cada um dos arrays cópias.

        pesoscrescente[i] = pesos[i];

        pesosdecrescente[i] = pesos[i];

    }

    ordenarcrescente(pesoscrescente, maxpacotes); //chama a função de ordenação crescente dos valores digitados usando a cópia do array original.

    printf("\n Pesos em Ordem Crescente \n");

    exibirarray(pesoscrescente, maxpacotes); //chama a função de exibição do array ordenado em ordem crescente.

    ordenardecrescente(pesosdecrescente, maxpacotes); //chama a função de ordenação decrescente dos valores digitados usando a cópia do array original.

    printf("\n Pesos em Ordem Decrescente \n");

    exibirarray(pesosdecrescente, maxpacotes); //chama a função de exibição do array ordenado em ordem decrescente.

    printf("\n");

    printf("Ordenações finalizadas");

return 0;

}

// Função para exibir o array

void exibirarray(float *arr*[], int *n*) {

    int i;

    printf("[ ");

    for (i = 0; i < *n*; i = i + 1) {

        printf("%.2f ", *arr*[i]);

        if (i < *n* - 1) {

            printf(", ");

        }

    }

    printf("]\n");

}

// Função para ordenar o array em ordem crescente

void ordenarcrescente(float *arr*[], int *n*) {

    int i, j;

    for (i = 0; i < *n* - 1; i++) { //loop externo para contolar o número de passagens

        for (j = 0; j < *n* - i - 1; j++) { // loop interno para fazer as comparações e trocas

            if (*arr*[j] > *arr*[j + 1]) { // se o elemento atual for maior que o próximo, troca-os de posição.

                float temp = *arr*[j]; // variável temporária para auxiliar na troca de valores

*arr*[j] = *arr*[j + 1];

*arr*[j + 1] = temp;

            }

        }

    }

}

// Função para ordenar o array em ordem decrescente

void ordenardecrescente(float *arr*[], int *n*) {

    int i, j;

    for (i = 0; i < *n* - 1; i++) { //loop externo para contolar o número de passagens

        for (j = 0; j < *n* - i - 1; j++) { // loop interno para fazer as comparações e trocas

            if (*arr*[j] < *arr*[j + 1]) { // se o elemento atual for menor que o próximo, troca-os de posição.

                float temp = *arr*[j]; // variável temporária para auxiliar na troca de valores

*arr*[j] = *arr*[j + 1];

*arr*[j + 1] = temp;

            }

        }

    }

}

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.