

Reimplemente o programa da questão anterior utilizando a função `qsort()` do C. Comente o seu código, explicando o que faz cada uma das linhas.

### Programa (Input):

```
#include <stdio.h>      // para o printf
#include <stdlib.h>     // para o qsort e malloc

int compare (const void * a, const void * b) // função para comparar e
ordenar os números de forma crescente
{
    return ( (int)a - (int)b ); /* foi criado com a seguinte lógica:
    - se a subtração de (int)a e (int)b for menor que zero, significa que o
    'b' é maior que o a, e que 'a' irá primeiro que o 'b' na sequência

    - se a subtração de (int)a e (int)b for igual a zero, significa que 'a'
    e 'b' são iguais, não importando a ordem deles, ou seja, não muda a
    sequência entre um e outro

    - se a subtração de (int)a e (int)b for maior que zero, significa que o
    'a' é maior do que o 'b', e que o 'a' irá depois que o 'b' na sequência
    */
}

int main (){ //inicializando a função principal
    float *x; // criando e definindo o tipo do ponteiro x
    int n, i; // criando variáveis de auxílio
    printf("Digite a quantidade de números que deseja ordenar: \n");
    //pedindo que o usuário digite a quantidade desejada de números que quer
    ordenar
    scanf("%d", &n); // coletando a entrada fornecida pelo usuário
    x = malloc(n*sizeof(float)); //alocando o vetor/ponteiro e
    especificando o tamanho dele dentro da função malloc()
    for(i=0; i<n; i++){ // um for() para o programa repetir que o usuário
    escreva os números que quer ordenar pela quantidade de vezes necessária,
    informado anteriormente quando pedimos a primeira entrada, armazenada na
    variável 'n'
        printf("Escreva o número %d ", i+1); // explicando que o usuário deve
        digitar o número (1, 2, 3, ..., n) para a comparação (também poderia ser:
        escreva o primeiro número, escreva o segundo número, mas a forma
        utilizada ficou mais facilmente automatizada)
        printf("\n"); //apenas utilidade visual, pular linha
        scanf("%f", &x[i]); // pedindo uma entrada ao usuário (os números
        desejados)
    }
    qsort (x, n, sizeof(float), compare); // dentro da função qsort, foi
    indicado, respectivamente o vetor (que usamos de forma alocada), o número
```

```
de elementos que serão ordenados, o tamanho de cada elemento e a função
de comparação definida antes do código principal (compare)
    for (i=0; i<n; i++){ // um for() para repetir o processo de impressão
        até que todos os elementos sejam mostrados
        printf ("%f ",x[i]); // aqui solitimos a impressão de todos os
        elementos do vetor com apenas duas casas decimais (lembrando que já
        estarão todos ordenados em ordem crescente, por consequência da função
        qsort() utilizada acima)
    }
    free(x); //foi usado para liberar o ponteiro x, sendo necessário após o
    uso do malloc() que está localizado próximo ao início do código
    return 0; //retornar 0 caso o programa seja executado normalmente
}
```

### Output (Exemplo):

```
❏ clang-7 -pthread -lm -o main main.c
❏ ./main
Digite a quantidade de números que deseja ordenar:
4
Escreva o número 1
3.14
Escreva o número 2
2.67
Escreva o número 3
1.602
Escreva o número 4
6.02
1.602000, 2.670000, 3.140000, 6.020000, ❏
```