

Iniciado em	quinta-feira, 9 jan. 2025, 10:06
Estado	Finalizada
Concluída em	quinta-feira, 9 jan. 2025, 10:29
Tempo empregado	23 minutos 10 segundos
Avaliar	100,00 de um máximo de 100,00



Questão 1

Correto

Atingiu 100,00 de 100,00

Cálculo de esforço do método QuickSort

Abaixo é apresentado o código do método de ordenação QuickSort. Adapte as funções quicksort e separa abaixo para:

1. Ordenar um vetor de caracteres;
2. Extrair quantas trocas de elementos do vetor são realizadas até que a ordenação termine.

CÓDIGOS

```
int separa (int v[], int p, int r) {
    int c = v[r]; // pivô
    int t, j = p;
    for (int k = p; k < r; ++k)
        if (v[k] <= c) {
            t = v[j], v[j] = v[k], v[k] = t;
            ++j;
        }
    t = v[j], v[j] = v[r], v[r] = t;
    return j;
}
```

```
void quicksort (int v[], int p, int r){
    if (p < r) {
        int j = separa (v, p, r);
        quicksort (v, p, j-1);
        quicksort (v, j+1, r);
    }
}
```

#####

Você está encarregado de adaptar as funções **quicksort** e **separa**.

Exemplo de aplicação para a ordenação de um vetor de char:

```
int main() {
    char v[1024];
    long int qt_trocas = 0;
    int n;
    if (fgets(v, sizeof(v), stdin) != NULL) {
        v[strcspn(v, "\n")] = '\0'; // Remove a nova linha, se existir
    }
    n = strlen(v); // Obtém o tamanho do vetor de caracteres
    printf("Antes : %s\n", v);
    quicksort(v, 0, n - 1, &qt_trocas);
    printf("Depois: %s\n", v);
    printf("Trocas: %ld\n", qt_trocas);
    return 0;
}
```

Em suma, sua tarefa é postar os códigos das funções **int separa** e **void quicksort** (**nesta ordem**) adaptados para o que se pede acima. Uma vez que você concluir esta função, as saídas deverão ser geradas conforme os exemplos apresentados.

POSTE APENAS OS CÓDIGOS DAS FUNÇÕES REQUERIDAS. NÃO POSTE NADA A MAIS (main, includes ou outro código qualquer).

For example:

Input	Result
uhjcba	Antes : uhjcba Depois: abchju Trocas: 9
estrutura	Antes : estrutura Depois: aerrsttuu Trocas: 13



Answer: (penalty regime: 0 %)

```
1 int separa(char v[], int p, int r, long int *qt_trocas) {
2     char c = v[r];
3     char t;
4     int j = p;
5     for (int k = p; k < r; ++k) {
6         if (v[k] <= c) {
7             t = v[j]; v[j] = v[k]; v[k] = t;
8             ++(*qt_trocas);
9             ++j;
10        }
11    }
12    t = v[j]; v[j] = v[r]; v[r] = t;
13    ++(*qt_trocas);
14    return j;
15 }
16
17 void quicksort(char v[], int p, int r, long int *qt_trocas) {
18     if (p < r) {
19         int j = separa(v, p, r, qt_trocas);
20         quicksort(v, p, j - 1, qt_trocas);
21         quicksort(v, j + 1, r, qt_trocas);
22     }
23 }
24
```

	Input	Expected	Got	
✓	uhjcba	Antes : uhjcba Depois: abchju Trocas: 9	Antes : uhjcba Depois: abchju Trocas: 9	✓
✓	estrutura	Antes : estrutura Depois: aerrsttuu Trocas: 13	Antes : estrutura Depois: aerrsttuu Trocas: 13	✓

Passou em todos os teste! ✓

[► Show/hide question author's solution \(C\)](#)**Correto**

Notas para este envio: 100,00/100,00.