

Trabalho 05 - Tabela de Símbolos Genérica

Data de entrega: 27/07/2022

Importante:

- **Não** olhe códigos de outros ou da internet. Exceto os que são fornecidos. Também não mostre ou publique o seu.
- Em caso de plágio, fraude ou tentativa de burlar o sistema será aplicado nota 0 na disciplina aos envolvidos.
- Alguns alunos podem ser solicitados para explicar com detalhes a implementação.
- Passar em todos os testes do run.codes não é garantia de tirar a nota máxima. Sua nota ainda depende do cumprimento das especificações do trabalho, qualidade do código, clareza dos comentários, boas práticas de programação e entendimento da matéria demonstrada em possível reunião.
- Você deverá submeter, até a data de entrega, o seu código na plataforma run.codes.

Este trabalho é individual e deverá ser implementado em linguagem C.

O método qsort do C aplica o QuickSort que já conhecemos para ordenar um arranjo de qualquer tipo de dados, o qsort do C tem 4 parâmetros, o primeiro é o endereço do vetor, o segundo é a quantidade de elementos, o terceiro é o tamanho em bytes de cada um dos elementos e o último é um ponteiro para uma função que compara dois desses elementos e determina qual é maior (ou se são iguais). Pesquise na internet como utilizar o qsort.

Neste trabalho iremos utilizar ponteiros para funções para criar uma tabela de símbolos que seja capaz de armazenar diferentes estruturas de dados. Você deverá implementar a TAD (tipo abstrato de dados) TABELA, que implementa pelo menos 3 funções:

`void adiciona_elemento(TABELA tab, void* elemento);`
que adiciona o elemento a tab, note que essa função não possui retorno, e por isso você não deve alterar o endereço de tab (estrutura com cabeça). Elementos iguais não devem ser colocados na tabela.

`void imprime_crescente(TABELA tab, void (*imprime)(const void*));`
que imprime os elementos em ordem crescente.

`void * busca_elemento(TABELA tab, void* elemento);`
que procura na tabela um registro igual a elemento, se não encontrar devolve NULL, se encontrar devolve o ponteiro para esse registro.

Você deverá baixar o arquivo `codigo.zip` que já tem o arquivo `Makefile`, `main.c`, o `tabela_simbolos.h` e o `tabela_simbolos.c`. Esse último é o único que você precisa implementar e submeter.

O programa lê um valor da entrada padrão do sistema que será o número de elementos sorteados para entrar na tabela.

Por exemplo:

5

Depois, a função main irá sortear 5 inteiros, 5 strings e 5 pontos (x,y) para adicionar na tabela, (os repetidos não são inseridos), por fim ele imprime os valores em ordem crescente. Com 5 a saída ficaria como abaixo (note que existem apenas 4 pontos, um ponto deve ter sido gerado repetido)

```
71876166
442951012
708592740
907283241
1483128881
FTKBJFDQL
HTFJGVMFT
KSCWMNTSB
QUCGERXOP
SLWFNKZHW
(0, 0)
(0, 3)
(3, 3)
(4, 4)
```

Obs: Para simplificar um pouco, não precisa implementar função de remover e nem de liberar memória.