#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include <string.h>

#include <locale.h>

#define TAMANHO 6

//----------------------------------------------

typedef struct

{

int chave;

char nome[20];

char poder[30];

} HashItem;

// tabela hash

typedef struct

{

HashItem \*vetor[TAMANHO]; // vetor de ponteiros para struct

} TabelaHash;

//----------------------------------------------

void iniciaTabelaHash(TabelaHash \*tabela)

{

for (size\_t i = 0; i < TAMANHO; i++)

{

// PEGAR TODAS AS posições

tabela->vetor[i] = NULL;

}

}

//----------------------------------------------

void imprimirItem(HashItem \*item)

{

if (item == NULL)

{

printf("NULL\n");

}

else

{

printf(" %s | %s\n", item->nome, item->poder);

}

}

void imprimirTabelaHash(TabelaHash \*tabela)

{

printf("----------TABLE----------\n");

for (size\_t i = 0; i < TAMANHO; i++)

{

printf("%d |", i);

imprimirItem(tabela->vetor[i]);

}

}

//----------------------------------------------

void inserirTabelaHash(TabelaHash \*tabela, HashItem \*item)

{

if (item->chave>=TAMANHO || item->chave<0)

{

printf("Warning: posicao nao existe");

return;

}

if (tabela->vetor[item->chave] == NULL)

{

tabela->vetor[item->chave] = item;

}

else

{

printf("Posição ocupada\n");

}

}

//----------------------------------------------

void destruirTabelaHash(TabelaHash \*tabela)

{

iniciaTabelaHash(tabela);

}

//----------------------------------------------

void removeTabelaHash(TabelaHash \*tabela, int chave)

{

if (chave>=TAMANHO || chave<0)

{

printf("Warning: posicao nao existe");

return;

}

else{

if (tabela->vetor[chave] != NULL)

{

tabela->vetor[chave] = NULL;

}

}

}

//----------------------------------------------

HashItem\* consultaTabelaHashEndDireto(TabelaHash \*tabela, int chave){

if (chave>=TAMANHO || chave<0)

{

printf("Warning: posicao nao existe");

return NULL;

}

return(tabela->vetor[chave]);

}

//----------------------------------------------

int main(int argc, char const \*argv[])

{

setlocale(LC\_ALL, "portuguese");

TabelaHash table;

iniciaTabelaHash(&table);

imprimirTabelaHash(&table);

HashItem bat = {1, "batiman", "rico"};

inserirTabelaHash(&table, &bat);

HashItem sup = {3, "super homem", "ser apelão"};

inserirTabelaHash(&table, &sup);

HashItem miran = {4, "miranha", "ser aranha"};

inserirTabelaHash(&table, &miran);

imprimirTabelaHash(&table);

HashItem \*retorno = consultaTabelaHashEndDireto(&table,1);

removeTabelaHash(&table, 1);

imprimirTabelaHash(&table);

printf("=====================\n");

destruirTabelaHash(&table);

imprimirTabelaHash(&table);

imprimirItem(retorno);

return 0;

}