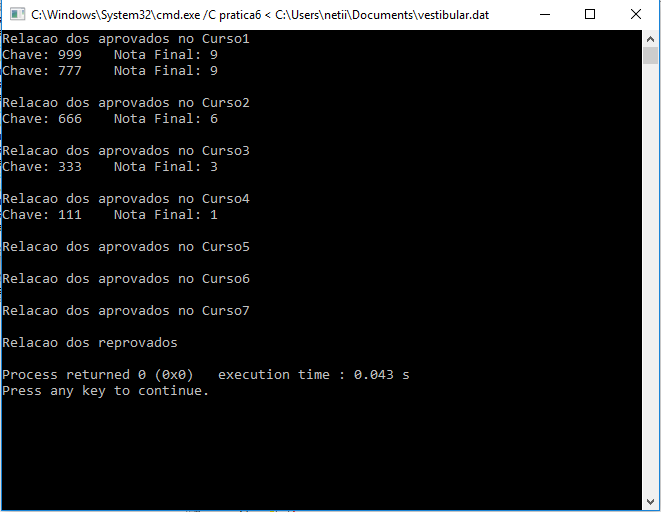
UFU

Nome: Antonio Carlos Neto

Curso: Ciência da Computação

Disciplina: GBC024

EXECUÇÃO:



main.c:

/\* Executar o programa lendo o arquivo: a.out <vestibular.dat \*/

#include <stdio.h>

#include <stdlib.h>

#include "list.h"

#define NOPCOES 3

#define NCURSOS 7

#define FALSE 0

#define TRUE 1

typedef int TipoChave;

typedef struct TipoItem

{

TipoChave Chave;

int NotaFinal;

int Opcao[NOPCOES];

} TipoItem;

ListElmt Registro;

List Classificacao[11];

List Aprovados[NCURSOS];

List Reprovados;

long Vagas[NCURSOS];

short Passou;

long i, Nota;

void print\_list(List \*list)

{

ListElmt\* element;

void\* data;

int i;

//fprintf(stdout, "List size is %d\n", list\_size(list));

i = 0;

element = list\_head(list);

TipoItem \*aux;

for (i = 0; i < list\_size(list); i++)

{

data = list\_data(element);

aux = (TipoItem\*)data;

printf("Chave: %d Nota Final: %d\n", aux->Chave, aux->NotaFinal);

element = list\_next(element);

}

return;

}

void LeRegistro(ListElmt \*Registro)

{

/\*---os valores lidos devem estar separados por brancos---\*/

int i;

TipoItem \*aux = malloc(sizeof(TipoItem\*));

scanf("%d %d", &aux->Chave, &aux->NotaFinal);

for (i = 0; i < NOPCOES; i++) scanf("%d", &aux->Opcao[i]);

Registro->data = (void\*)aux;

}

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

{

for (i = 1; i <= NCURSOS; i++) scanf("%ld", &Vagas[i-1]);

scanf("%\*[^\n]");

getchar();

for (i = 0; i <= 10; i++) list\_init(&Classificacao[i], free);

for (i = 1; i <= NCURSOS; i++) list\_init(&Aprovados[i-1], free);

list\_init(&Reprovados, free);

LeRegistro(&Registro);

void \*aux = Registro.data;

TipoItem \*aux2 = (TipoItem\*)aux;

ListElmt \*element;

while (aux2->Chave != 0)

{

list\_ins\_next(&Classificacao[aux2->NotaFinal], NULL, Registro.data);

LeRegistro(&Registro);

aux = Registro.data;

aux2 = (TipoItem\*)aux;

}

for (Nota = 10; Nota >= 0; Nota--)

{

while (list\_size(&Classificacao[Nota]) != 0)

{

element = list\_head(&Classificacao[Nota]);

if (list\_size(&Classificacao[Nota]) > 1)

list\_rem\_next(&Classificacao[Nota], element, &Registro.data);

else if (list\_size(&Classificacao[Nota]) == 1)

list\_rem\_next(&Classificacao[Nota], NULL, &Registro.data);

i = 1;

Passou = FALSE;

aux = Registro.data;

aux2 = (TipoItem\*)aux;

while (i <= NOPCOES && !Passou)

{

if (Vagas[aux2->Opcao[i-1] - 1] > 0)

{

list\_ins\_next(&Aprovados[aux2->Opcao[i-1]-1], NULL, aux);

//Insere(Registro, &Aprovados[Registro.Opcao[i-1] - 1]);

Vagas[aux2->Opcao[i-1] - 1]--;

//Vagas[Registro.Opcao[i-1] - 1]--;

Passou = TRUE;

}

i++;

}

if (!Passou) list\_ins\_next(&Reprovados, NULL, aux);

}

}

for (i = 1; i <= NCURSOS; i++)

{

printf("Relacao dos aprovados no Curso%ld\n", i);

print\_list(&Aprovados[i-1]);

printf("\n");

//Imprime(Aprovados[i - 1]);

}

printf("Relacao dos reprovados\n");

print\_list(&Reprovados);

//Imprime(Reprovados);

return 0;

}

list.c igual ao exemplo dado, e a list.h só tem os define a mais.