Nome: João Paulo de Oliveira 11611BCC046

10° Aula prática

Uberlândia

2016

1.Código fonte:

1.1 Parte 1:

* Main.c:

#include <stdio.h>

#include <stdlib.h>

#include <sys/time.h>

#define MAXTAM 4

typedef int TipoApontador;

typedef int TipoChave;

typedef struct {

TipoChave Chave;

/\* outros componentes \*/

} TipoItem;

typedef struct {

TipoItem Item[10];

TipoApontador Frente, Tras;

} TipoFila;

void FFVazia(TipoFila \*Fila)

{ Fila->Frente = 1;

Fila->Tras = Fila->Frente;

}

int Vazia(TipoFila Fila)

{ return (Fila.Frente == Fila.Tras); }

void Enfileira(TipoItem x, TipoFila \*Fila)

{ if (Fila->Tras % 10 + 1 == Fila->Frente)

printf(" Erro fila est a cheia\n");

else { Fila->Item[Fila->Tras - 1] = x;

Fila->Tras = Fila->Tras % 10 + 1;

}

}

void Desenfileira(TipoFila \*Fila, TipoItem \*Item)

{ if (Vazia(\*Fila))

printf("Erro fila esta vazia\n");

else { \*Item = Fila->Item[Fila->Frente - 1];

Fila->Frente = Fila->Frente % 10 + 1;

}

}

void Imprime(TipoFila Fila)

{ int Aux;

for (Aux = Fila.Frente - 1; Aux <= (Fila.Tras - 2); Aux++)

printf("%12d\n", Fila.Item[Aux].Chave);

} /\* Imprime \*/

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

{ struct timeval t;

TipoFila fila;

TipoItem item;

int vetor[10];

int i, j, k, n, max;

gettimeofday(&t,NULL);

srand((unsigned int)t.tv\_usec);

max = 9;

FFVazia(&fila);

/\*Gera uma permutacao aleatoria de chaves entre 1 e max\*/

for(i = 0; i < max+1; i++) vetor[i] = i + 1;

for(i = 0; i < max; i++){

k = (int) (10.0 \* rand()/(RAND\_MAX + 1.0));

j = (int) (10.0 \* rand()/(RAND\_MAX + 1.0));

n = vetor[k];

vetor[k] = vetor[j];

vetor[j] = n;

}

/\*Enfileira cada chave \*/

for (i = 0; i < max; i++){

item.Chave = vetor[i];

Enfileira(item, &fila);

printf("Enfileirou: %d \n", item.Chave);

}

/\*Desenfileira cada chave \*/

for(i = 0; i < max; i++)

{ Desenfileira(&fila, &item);

printf("Desenfileirou: %d\n", item.Chave);

}

return(0);

}

1.2 Parte 2:

* Main.c:

#include <stdio.h>

#include <stdlib.h>

#include "lista.h"

int main(){

Fila\* fila;

fila = cria\_Fila();

int el=0,i;

for(i=0;i<5;i++){

scanf("%d",&el);

if (i==0)fila->inicio = el;

else insere\_Fila(fila,el);

printf("inserido: %d\n",el);

if(i==3){

printf("removido: %d\n",fila->inicio);

remove\_Fila(fila);

}

}

for(i=0;i<4;i++){

printf("removido: %d\n",fila->inicio);

remove\_Fila(fila);

}

libera\_Fila(fila);

return 0;

}

* fila.c:

#include "lista.h"

#include <stdio.h>

#include <stdlib.h>

Fila\* cria\_Fila(){

Fila \*fi = (Fila\*) malloc(sizeof(struct fila));

if(fi!=NULL){

fi->inicio = 0;

fi->final = 0;

fi->qtd = 0;

}

return fi;

}

void libera\_Fila(Fila\* fi){

free(fi);

}

int tamanho\_Fila(Fila\* fi){

if(fi==NULL)

return 1;

return fi->qtd;

}

int Fila\_cheia(Fila\*fi){

if(fi == NULL)return -1;

if(fi->qtd==MAX)

return 1;

else

return 0;

}

int Fila\_vazia(Fila\*fi){

if(fi == NULL)return -1;

if(fi->qtd==0)

return 1;

else

return 0;

}

int insere\_Fila(Fila\* fi, int elemento){

if(fi == NULL)return 0;

if(Fila\_cheia(fi))return 0;

fi->elemento = elemento;

fi ->final = (fi->final+1)%MAX;

fi->qtd++;

return 1;

}

int remove\_Fila(Fila \*fi){

if(fi == NULL||Fila\_vazia(fi))return 0;

fi->inicio = (fi->inicio+1)%MAX;

fi->qtd--;

return 1;

}

int consulta\_Fila(Fila\* fi, int elemento){

if(fi == NULL||Fila\_vazia(fi))return 0;

elemento = fi->elemento;

return 1;

}

* fila.h:

#ifndef LISTA\_H\_INCLUDED

#define LISTA\_H\_INCLUDED

#define MAX 100

struct fila{

int inicio,final,qtd;

int elemento;

};

typedef struct fila Fila;

Fila\* cria\_Fila();

void libera\_Fila(Fila\* fi);

int tamanho\_Fila(Fila\* fi);

int Fila\_cheia(Fila\*fi);

int Fila\_vazia(Fila\*fi);

int insere\_Fila(Fila\* fi, int elemento);

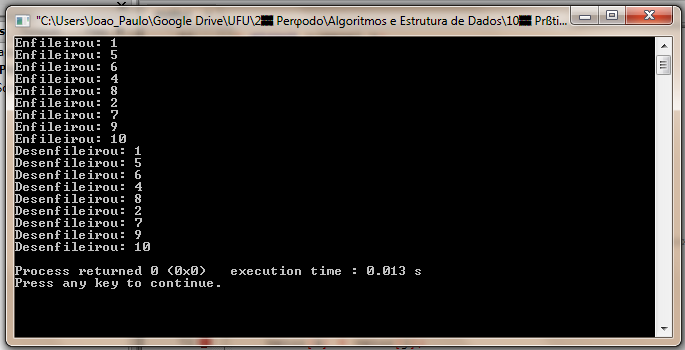
int remove\_Fila(Fila \*fi);

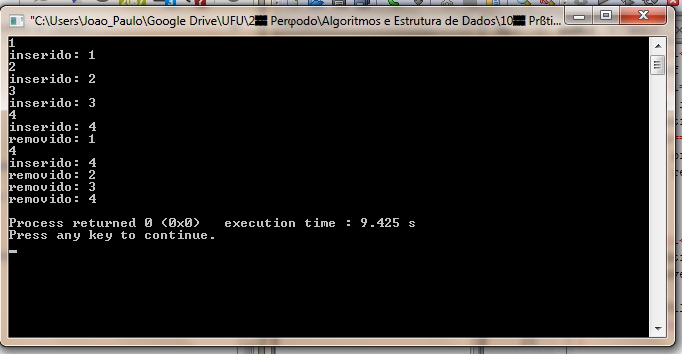
int consulta\_Fila(Fila\* fi, int elemento);

#endif // LISTA\_H\_INCLUDED

2.Print do funcionamento:

2.1 Parte 1:



 2.2 Parte 2: