KAHLIL BELLO

PROJECT 7

OCT 31 2018

PROBLEM 1

#include <iostream>

#pragma once

#ifndef LINKEDLIST

#define LINKEDLIST

using namespace std;

template <class G>

class QUEUE\_LIST

{

private:

struct node

{

G info;

node \*next;

};//end struct node

node \*front, \*rear;

public:

QUEUE\_LIST() { front = NULL; rear = NULL; }

//------------add a node to the rear of the queue

void pushq(G x)

{

node \*p = new(node);

p->info = x;

p->next = NULL;

if (front == NULL)

{

front = p;

rear = p;

}//endif

else

{

rear->next = p;

rear = p;

}//endelse

}//end pushq

//--------test whether queue is empty

bool emptyq()

{

return (front == NULL) ? true : false;

}

//----------pop the first node

G popq()

{

node \*p = front;

G x;

x = front->info;

front = front->next;

delete(p);

return x;

}//end popq

void clearq()

{

front = NULL;

}//end clearq

void makeCircular()

{

rear->next = front;

}//end makecircular

void displayCircular()

{

node \*p = front;

do {

cout << p->info << "->";

p = p->next;

} while (p != front);

cout << "->circular restart->";

}

};//end QUEUE\_LIST

#endif

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

SOURCE

#include "LINKEDLIST.h"

#include <ctime>

#include <cstdlib>

#include <iostream>

using namespace std;

int main()

{

srand(time(0));

QUEUE\_LIST <int> randQue;

int randNumbers[10];

cout << "The 10 random numbers: ";

for (int i = 0; i < 10; ++i)

{

randNumbers[i] = (rand()%50)+1;

cout << randNumbers[i] << " ";

randQue.pushq(randNumbers[i]);

}//endfor

cout << endl;

cout << "//--------------------------------\n";

cout << "Outputting in circular linked list form. . .\n";

randQue.makeCircular();

randQue.displayCircular();

cout << endl;

randQue.displayCircular();

cout << endl;

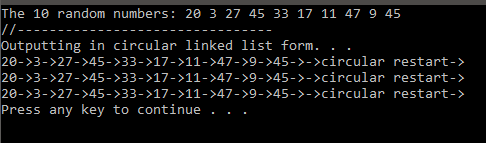
randQue.displayCircular();

cout << endl;

system("pause");

return 0;

}//end main



PROBLEM 2

HEADER

#pragma once

#include <iostream>

#include <string>

#ifndef HASH

#define HASH

using namespace std;

class HASHEN

{

private:

struct node

{

string nameInfo = "sad";

int daysInfo;

node \*next;

};//end struct node

node \*H[4];

public:

void clearHashTable()

{

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

{

H[i] = NULL;

}//endfor

}//end clearhashtable

int hashFun(int x) //hash function

{

return x % 4;

}//end hashFun

void push(string x, int y)

{

//find the insertion place first

int i = hashFun((x[0]) + (x[1]) + (x[2]));

node \*p = new node; //p->[x]

p->nameInfo = x;

p->daysInfo = y;

//inser this nwe node in front of H[i];

p->next = H[i]; H[i] = p;

}//end push

void search(string x)

{

//send to hash function for searching if it belongs to H[0] etc...

int i = hashFun((x[0]) + (x[1]) + (x[2]));

//create copy of head node

node \*p = H[i];

while (p != NULL)

{

if (p->nameInfo == x) { break; }

p = p->next;

}//endwhile

if (p == NULL)

{

cout << x << " Does not exist\n";

}

else

{

cout << x << " is found which has " << p->daysInfo << " days.\n";

}

}//end search

void displayH()

{

node \*p;

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

{

p = H[i];

cout << "H[" << i << "]->";

while (p != NULL)

{

cout << p->nameInfo << ", " << p->daysInfo << "->";

p = p->next;

}//endwhle

cout << "NULL\n";

}//endfor

}//end displayh

};//end class hash

#endif

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

source:

#include <iostream>

#include "HASH.h"

#include <string>

using namespace std;

int main()

{

struct MONTH

{

string name;

int days;

};//end struct month

//int a[9] = { 7,10,13,19,14,20,33,44,15 };

MONTH month[12] = { {"JAN",31},{"FEB",28},{"MAR", 31},{"APR",30},{"MAY",31},{"JUN",30},{"JUL",31},{"AUG",31},{"SEP",30},{"OCT",31},{"NOV",30},{"DEC",31} };

HASHEN h;

h.clearHashTable();

for (int i = 0; i < 12; ++i)

{

h.push(month[i].name, month[i].days);

}//endfor

h.displayH();

//---we want to search an item

cout << "//---------------------------------------------\n";

string item;

char again = 'y';

do {

cout << "Enter the value of an item you want to search: ";

cin >> item;

h.search(item);

cout << "Would you like to search another item? ";

cin >> again;

} while (again=='y' || again=='Y');

system("pause");

return 0;

}

OUTPUT:

