**TRƯỜNG ĐẠI HỌC MỞ HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

****

**BÀI TẬP LỚN**

**MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**QUẢN LÝ CÔNG TRÌNH XÂY DỰNG**

*Giáo viên hướng dẫn :* **ThS. Nguyễn Thị Tâm**

*Họ và tên :* **Hà Văn Cảnh – 2110A03**

**Mai Thị Thảo Chi – 2110A03**

**HÀ NỘI - 2022**

MENU

[**I. XÁC ĐỊNH BÀI TOÁN** 3](#_Toc121689867)

[**II. XÁC ĐỊNH CTDL BIỂU DIỄN BÀI TOÁN** 3](#_Toc121689868)

[**III. XÁC ĐỊNH CÁC THUẬT TOÁN** 3](#_Toc121689869)

[**IV. CHỨC NĂNG BÀI TOÁN QUẢN LÝ NHẬP VẬT TƯ TẠI CÔNG TY XÂY DỰNG** 9](#_Toc121689870)

[**V. LẬP TRÌNH - CÀI ĐẶT - KIỂM THỬ - TEST** 11](#_Toc121689871)

[I. DSLK DON QUAN LI KHACH HANG 11](#_Toc121689872)

[II. DSLK ĐÔI QUẢN LÍ ĐƠN VỊ THI CÔNG 11](#_Toc121689873)

[III. STACK QUẢN LÝ NHÂN VIÊN 11](#_Toc121689874)

[IV. QUEUE QUẢN LÝ HỢP ĐỒNG 11](#_Toc121689875)

# **I. XÁC ĐỊNH BÀI TOÁN**

Bài toán Quản lý công trình xây dựng nhằm hỗ trợ nắm rõ thông tin công trình để có thể quản lý các hoạt động của công trình thuận tiện và nhanh chóng hơn. Đồng thời giúp quản lý, sắp xếp các công trình ở nhiều hạng mục.

# **II. XÁC ĐỊNH CTDL BIỂU DIỄN BÀI TOÁN**

Đối tượng quản lý bài toán bao gồm: **khachHang,** **donviThiCong, hopDong** và **nhanVien** trong đó:

\***khachHang**: cccdKH, tenKH, ngaySinhKH, sdtKH, diaChiKH, maHopDong

\***donViThiCong**: maDonVi, maHopDong, slTho, thoiGianBatDau, tenKTS, sdtKTS

\***hopDong**: maHopDong, ngayKi, soTienDuTru, coc, hangMuc, maNhanVien

**\*nhanVien:** maNV, tenNV, ngaySinhNV, sdtNV, maHopDong, luongCB, hoaHong

# **III. XÁC ĐỊNH CÁC THUẬT TOÁN**

**\* *Các thuật toán của hàng đợi***

1. **Thêm vào cuối danh sách**

Bước 1: Bắt đầu hàm

Bước 2: Nhận vào hàm &Q thuộc Queue và &x chứa thông tin vật liệu

Bước 3: Khởi tạo con trỏ p thuộc kiểu Node

Bước 4: p->data = x, p->next = NULL (vì thêm vào cuối danh sách nên prev sẽ null);

Bước 5:

* Nếu Q.Head = NULL (danh sách rỗng) thì p->prev = NULL và gán Q.Head = Q.Tail = p
* Ngược lại: p->prev = Q.Tail, Q.Tail->next = p, Q.Tail = p;

Bước 6: Kết thúc hàm

*Sơ đồ thuật toán*

Node \*p = new Node()

Nhận vào Queue &Q, Sanpham &x

p->data = x;

p->next = NULL

Q.Head == NULL

p->prev = NULL

Q.Head = Q.tail = p

p->prev = Q.Tail

Q.Tail->next = p

Q.Tail = p;

Đúng

Sai

1. **Tìm kiếm**

Bước 1: Bắt đầu

Bước 2: Nếu Q.head == NULL thì thông báo ds rỗng và thoát hàm

Bước 3: Nhập vào thông tin cần tìm (có thể là mã vật liệu, tên, chất liệu)

Bước 4: Khởi tạo Node \*p = Q.head

Bước 5: So sánh dữ liệu trong p với thông tin nhập vào

* Nếu trùng khớp thì in ra thông tin vật liệu và thoát hàm
* Ngược lại gán p = p->next

Bước 6:

* Nếu p == NULL chuyển đến bước 7
* Ngược lại, quay về bước 5

Bước 7: Kết thúc

*Ví dụ sơ đồ thuật toán tìm theo* ***mã vật liệu***

Node \*p = Q.Head

Nhập mã v\_lieu cần tìm

Q.Head == NULL

Đúng

**p->data.maVatLieu = ma\_vlieu**

Sai

Sai

In ra thông tin vật liệu

p == NULL

p = p->next

Đúng

Đúng

Sai

Ds rỗng

1. **Hàm sắp xếp**

Ở đây ta lấy ví dụ trực tiếp vào hàm sắp xếp theo mã v.liệu tăng dần (kiểu int)

Bước 1: Bắt đầu

Bước 2: Nếu Q.Head == NULL thì đưa ra thông báo ds rỗng và kết thúc

Bước 3: Khởi tạo Node \*p = Q.Head

Bước 4: Khởi tạo Node \*q = p->next

Bước 5:

* Nếu q == NULL thì chuyển đến bước 7
* Ngược lại mã v.liệu của p > của q thì thực hiện thao tác hoán đổi data

Bước 6: q = q->next, quay lại bước 5

Bước 7: p = p->next

Bước 8:

* Nếu p == NULL chuyển đến bước 9
* Ngược lại quay về bước 4

Bước 9: Kết thúc

Node \*p = Q.Head

Q.Head == NULL

Sai

Node \*q = p->next

q != NULL

p = p->next

p != NULL

Hoán đổi data giữa hai Node

p->data.mavl > q->data.mavl

Đúng

Đúng

Đúng

Sai

Đúng

Ds rỗng

q = q->next

Sai

Sai

1. **Hàm xóa 1 phần tử**

Thuật toán minh họa cho xóa 1 phần tử theo mã vật liệu

Bước 1: Bắt đầu

Bước 2: Nếu Q.Head == NULL đưa ra thông báo rỗng rồi kết thúc

Bước 3: Khởi tạo Node \*p = Q.Head và biến mavl lưu mã vật liệu muốn xóa

Bước 4: Nhập vào mavl muốn xóa

Bước 5: Nếu mavl == p->data.maVatLieu thì

- Nếu p->prev == NULL (p là phần tử đầu tiên của ds) :

+ Nếu p->prev == NULL (ds chỉ có 1 p.tử) thì gán

Q.Head = Q.Tail = NULL, delete p; và chuyển đến bước 8 - Nếu p->next == NULL (p là tử cuối) thì Q.Tail = p->prev;

Q.Tail->next = NULL, delete p; và chuyển đến bước 8

- Trường hợp còn lại, ta thiết lập liên kết giữa Node trước và sau p, sau đấy xóa Node p đi và chuyển đến bước 8

+ p->prev->next = p->next

+ p->next->prev = p->prev

+ delete p

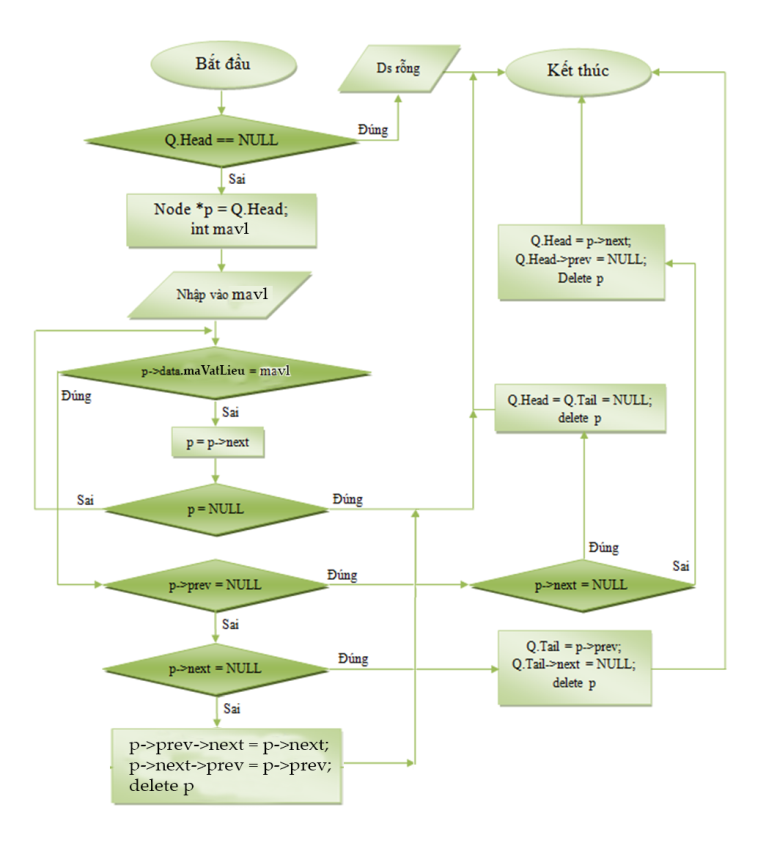
+ Chuyển đến bước 8.

Bước 6: p = p->next

Bước 7: Nếu p == NULL thì chuyển đến bước 8, ngược lại quay về bước 5

Bước 8: Kết thúc

Sơ đồ thuật toán



# **IV. CHỨC NĂNG BÀI TOÁN QUẢN LÝ NHẬP VẬT TƯ TẠI CÔNG TY XÂY DỰNG**

Chương trình khi chạy ban đầu xuất xuất hiện danh sách lựa chọn chức năng:

1. Quản lý danh sách khách hàng(dslk đơn)

2. Quả lý danh sách đơn vị thi công(dslk đôi)

3.Quản lý danh sách nhân viên(stack)

4. Quản lý danh sách hợp đông(queue)

Mời bạn chọn:

Nếu chọn số 1, gồm các chức năng sau:

cout<< "1.nhap thong tin"<<endl;

cout<< "2.xuat thong tin"<<endl;

cout<< "3.luu file"<<endl;

cout<< "4.doc file"<<endl;

cout<< "5.tim thong tin theo cccd"<<endl;

cout<< "6.sua thong tin khach hang bang ma cccd"<<endl;

cout<< "8.xoa thong tin khach hang"<<endl;

Mời bạn chọn:

Nếu bạn chọn số 2, gồm các chức năng:

cout<< "1.nhap thong tin"<<endl;

cout<< "2.xuat thong tin"<<endl;

cout<< "3.luu file"<<endl;

cout<< "4.doc file"<<endl;

cout<< "5.tim thong tin don vi theo id"<<endl;

cout<< "6.sua thong tin don vi theo id"<<endl;

cout<< "7.xoa thong tin don vi"<<endl;

cout<< "8.sap xep theo chi phi"<<endl;

Mời bạn chọn:

Nếu bạn chọn số 3, gồm các chức năng:

cout<< "1.nhap thong tin"<<endl;

cout<< "2.xuat thong tin"<<endl;

cout<< "3.luu file"<<endl;

cout<< "4.doc file"<<endl;

cout<< "5.top"<<endl;

cout<< "6.ktra rong"<<endl

cout<< "7.lay phan tu dau top: "<<endl;

Mời bạn chọn:

Nếu bạn chọn số 4, gồm các chức năng:

cout<< "1.nhap thong tin"<<endl;

cout<< "2.xuat thong tin"<<endl;

cout<< "3.luu file"<<endl;

cout<< "4.doc file"<<endl;

cout<< "5.top\_eq"<<endl;

cout<< "6.push\_eq"<<endl;

cout<< "7.ktra danh sach rong"<<endl;

cout<< "8.pop\_eq"<<endl;

Mời bạn chọn:

# **V. LẬP TRÌNH - CÀI ĐẶT - KIỂM THỬ - TEST**

## DSLK DON QUAN LI KHACH HANG

1. Code xử lý chính:

struct KhachHang{

char fullname[10];

char address[20];

char id\_contract[20];

char phone [10];

char birthday [10];

int cccd;

float expense;

};

//hàm nhập một khach hang

void input (KhachHang &khachhang){

cout<< "full name: ";

fflush(stdin);

gets(khachhang.fullname);

cout<< "birthday: ";gets(khachhang.birthday);

cout<< "address: ";gets(khachhang.address);

cout<< "cccd: ";cin>>khachhang.cccd;

cout<< "phone: ";

cin.ignore();gets(khachhang.phone);

cout<< "expense: ";cin>>khachhang.expense;

fflush(stdin);

cout<< "id\_contract: ";gets(khachhang.id\_contract);

cout<< "======================"<<endl;

}

// hàm xuất khách hàng

void output(KhachHang khachhang){

cout<< setw(10)<< khachhang.fullname;

cout<< setw(10)<< khachhang.birthday;

cout<< setw(10)<< khachhang.address;

cout<< setw(10)<< khachhang.cccd;

cout<< setw(10)<< khachhang.phone;

cout<< setw(10)<< khachhang.expense;

cout<< setw(10)<< khachhang.id\_contract;

cout<<endl;

}

// tạo node

struct node {

KhachHang data;

struct node \*pNext;

};

// tạo danh sach quan li boi phead và ptail

struct LIST {

node \*pHead;

node \*ptail;

};

// khoi tao dnah sach

void create\_list(LIST &List)

{

List.pHead = NULL;

List.ptail = NULL;

}

// ham khoi tao trả về node

node \*create\_node(KhachHang khachhang)

{

node \*new\_KH = new node;

new\_KH->data = khachhang;

new\_KH->pNext =NULL;

return new\_KH;

}

// hàm thêm đầu danh sách

void add\_first\_list(LIST &List,node\*p)

{

if(List.pHead == NULL){

List.pHead = List.ptail = p;

}

else{

p->pNext = List.pHead;

List.pHead = p;

}

}

// hàm thêm cuối danh sách

void add\_last\_list(LIST &List,node \*p)

{

if(List.pHead == NULL){

List.pHead = List.ptail = p;

}

else{

List.ptail->pNext = p;

List.ptail = p;

}

}

//hàm nhập danh sách

void input\_list(LIST &List)

{

int n;

cout<< "so luong khach hang can quan li: ";cin>>n;

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

{

KhachHang khachhang;

cout<< "khach hang thu "<<i+1<<endl;

input(khachhang);

node \*p = create\_node(khachhang);

int pick;

cout<< "1.add first"<<endl;

cout<< "2.add last"<<endl;cin>>pick;

switch (pick){

case 1:

add\_first\_list(List,p);

break;

case 2:

add\_last\_list(List,p);

break;

}

}

}

// hàm tìm thông tin khách hàng bằng cccc

void find\_infor\_by\_cccd (LIST List)

{

int cccd\_temp;

cout<< "nhap cccd cua khach hang muon tim: ";cin>>cccd\_temp;

for(node \*k = List.pHead;k !=NULL; k=k->pNext){

if(k->data.cccd == cccd\_temp){

output(k->data);

}

}

}

// ham xoa dau danh sach

void remove\_firstList (LIST &List)

{

if(List.pHead == NULL)

cout<< "ko co du lieu"<<endl;

else

{

node \*p = List.pHead;

List.pHead = List.pHead->pNext;

delete p;

}

}

// ham xoa cuoi danh sach

void remove\_lastList(LIST &List)

{

for(node \*i=List.pHead;i!= NULL;i=i->pNext)

{

if(i->pNext == List.ptail)

{

delete List.ptail;

i->pNext = NULL;

List.ptail = i;

}

}

}

//xóa khách hàng bằng cccd

void remove\_cccd(LIST &List)

{

node \*p;

int cccd\_remove;

cout<< "nhap cccd cua khach hang can xoa: ";

cin>>cccd\_remove;

if(cccd\_remove == List.pHead->data.cccd)

{

remove\_firstList(List);

return;

}

if(cccd\_remove == List.ptail->data.cccd)

{

remove\_lastList(List);

return;

}

for(node \*i=List.pHead;i!= NULL;i=i->pNext)

{

if(i->data.cccd == cccd\_remove)

{

p->pNext = i->pNext;

delete i;

return;

}

p=i;

}

}

// huy toan bo danh sach

void huy\_toan\_bo\_ds(LIST &List)

{

for(node \*i=List.pHead;i!= NULL;i=i->pNext)

{

node\* p = List.pHead;

List.pHead=List.pHead->pNext;

delete p;

}

List.ptail=NULL;

}

// sắp sếp theo số tiền đầu tư

void sort\_by\_expense(LIST List)

{

for(node \*i = List.pHead;i!=NULL;i=i->pNext)

{

for(node \*j=i->pNext;j!=NULL;j=j->pNext)

{

if(i->data.expense > j->data.expense){

KhachHang temp = i->data;

i->data = j->data;

j->data = temp;

}

}

}

}

// khác hàng có chi phí cao nhất

void KhacHang\_co\_chi\_phi\_thap(LIST List)

{

sort\_by\_expense(List);

output(List.pHead->data);

}

//tổng tiền của các khách hàng

void tong\_tien\_cua\_cac\_khach\_hang (LIST List)

{

float sum = 0;

for(node \*i = List.pHead;i!=NULL;i=i->pNext)

{

sum += i->data.expense;

}

cout<< "tong tien: "<<sum<<endl;

}

// ham sua thong tin khach hang

void fix\_info(LIST List)

{

int cccd\_temp;

cout<< "nhap cccd cua khach hang muon sua: ";cin>>cccd\_temp;

for(node \*k = List.pHead;k !=NULL; k=k->pNext){

if(k->data.cccd == cccd\_temp){

input(k->data);

}

}

}

// đọc file

void write\_file (LIST List)

{

FILE \*f;

f = fopen("file.dat", "wb");

int n = 0;

node\* p;

for (p = List.pHead; p != NULL; p = p->pNext)

n = n + 1;

fwrite(&n, sizeof(int), 1, f);

for (p = List.pHead; p != NULL; p = p->pNext)

fwrite(&p->data, sizeof(KhachHang), 1, f);

fclose(f);

cout << "\nSuccessfully Saved File!";

}

// ghi file

void read\_file(LIST &List)

{

FILE\* f;

KhachHang x;

int i, n;

node\* p;

node\* q;

f = fopen("file.dat", "rb");

fread(&n, sizeof(int), 1, f);

bool check = true;

int count = 0;

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

fread(&x, sizeof(KhachHang), 1, f);

p = create\_node(x);

for (q = List.pHead; q != NULL; q = q->pNext) {

if (q->data.cccd == p->data.cccd) {

check = false;

}

}

if (check) {

add\_last\_list(List, p);

count++;

}

}

if (count > 0)

cout << "\nSuccessfully Read File!";

else

cout << "\nUnsuccessfully Read File Due To Duplicated ProfileCode!";

fclose(f);

}

// hàm hiện danh sách

void show\_list(LIST List){

for(node \*k = List.pHead;k !=NULL; k=k->pNext){

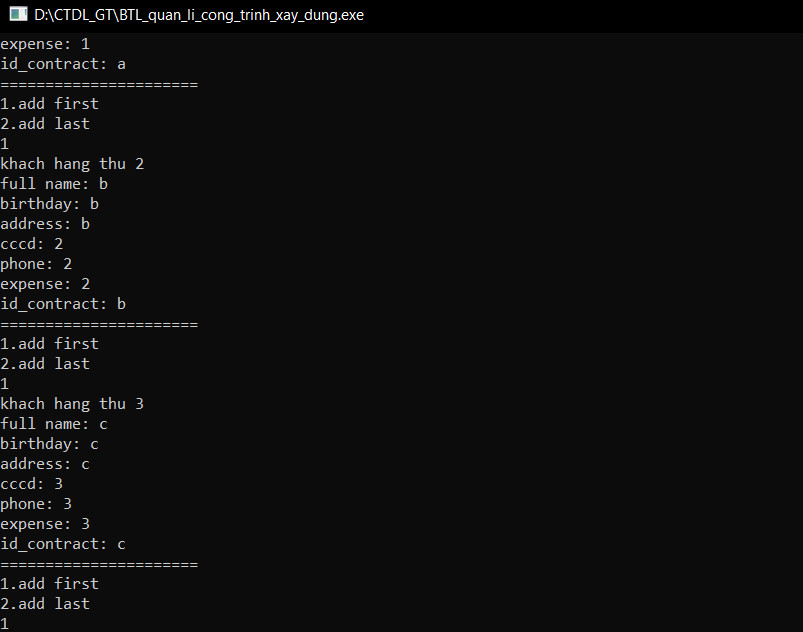
output(k->data);

}

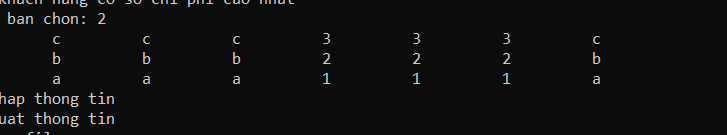
}

2. Kiểm thử

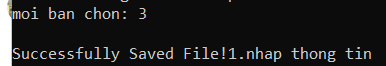
1.nhập thông tin



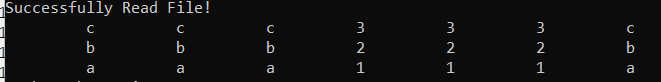
2.in thông tin



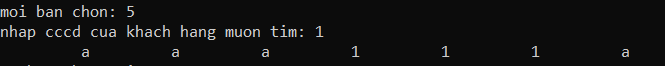
3.lưu file



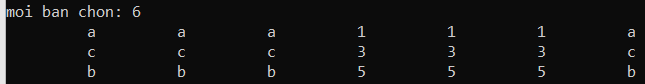
4.đọc file



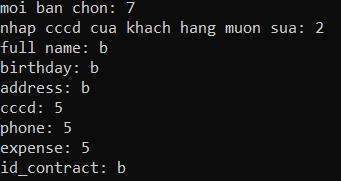
5. tim thong tin theo cccd



6. sap sep khach hang theo chi phi



7.sưa thong tin khach hang



8.xoa thông tin khác hàng



9.tính tổng



10 thông tin khách hàng có chi phí thấp nhất



## DSLK ĐÔI QUẢN LÍ ĐƠN VỊ THI CÔNG

1. Code xử lý chính:

//============DANH SACH LIÊN KẾT ĐÔI=====================

// quan li don vi thi cong

// TẠO KIEU DU LIEU

struct company {

char id\_company[15];

char id\_contact[15];

char time\_start[15];

char progress[15];

int number\_workers;

float material\_money,cost;

};

// TẠO NODE

struct node\_company{

company data;

struct node\_company \*pNext;

struct node\_company \*pPrev;

};

// TSO DANH SACH DE QUAN LI

struct LIST\_company {

node\_company\*pHead;

node\_company\*ptail;

};

// HAM NHAP

void input\_company (company &company\_temp)

{

cout<< "id\_company: ";fflush(stdin);gets(company\_temp.id\_company);

cout<< "id\_contact: ";gets(company\_temp.id\_contact);

cout<< "time\_start(dd/mm/yy): ";gets(company\_temp.time\_start);

cout<< "number\_workers: ";cin>>company\_temp.number\_workers;

cout<< "cost: ";cin>>company\_temp.cost;

cout<< "material\_money: ";cin>>company\_temp.material\_money;

cout<< "progress: ";fflush(stdin);gets(company\_temp.progress);

cout<< "------------------------------"<<endl;

}

// HÀM XUẤT

void output\_datacompany(company company\_temp)

{

cout<< setw(10)<<company\_temp.id\_company;

cout<< setw(10)<<company\_temp.id\_contact;

cout<< setw(10)<<company\_temp.time\_start;

cout<< setw(10)<<company\_temp.material\_money;

cout<< setw(10)<<company\_temp.cost;

cout<< setw(10)<<company\_temp.number\_workers;

cout<< setw(10)<<company\_temp.progress;

cout<<endl;

}

// TAO DANH SACH

void createList\_company(LIST\_company &List)

{

List.pHead = NULL;

List.ptail = NULL;

}

// HÀM TRẢ VỀ NODE

node\_company \*creteNode\_company(company company\_temp)

{

node\_company \*p = new node\_company;

p->data = company\_temp;

p->pNext = NULL;

p->pPrev = NULL;

return p;

}/\*

void input\_list\_company(LIST\_company &List\_c)

{

int n;

cout<< "nhap vao so luong don vi thi cong can quan li: "<<endl;

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

{

company temp;

input\_company(temp);

node\_company \*p = creteNode\_company(temp);

addFirstLisTCompany(List\_c,p);

}

}\*/

// HAM THEM ĐẦU

void addFirstLisTCompany(LIST\_company &List,node\_company \*temp)

{

if(List.pHead == NULL && List.ptail == NULL){

List.pHead = List.ptail = temp;

}else

{

temp->pNext = List.pHead;

List.pHead ->pPrev = temp;

List.pHead = temp;

}

}

// HÀM THÊM CUỐI

void addLastListCompany(LIST\_company &List,node\_company \*temp)

{

if(List.pHead == NULL && List.ptail == NULL){

List.pHead = List.ptail = temp;

}

else

{

List.ptail->pNext = temp;

temp->pPrev = List.ptail;

List.ptail = temp;

}

}

// HÀM NHẬP DANH SÁCH VÀO LIST

void input\_list\_company(LIST\_company &List\_c)

{

int n;

cout<< "nhap vao so luong don vi thi cong can quan li: "<<endl;cin>>n;

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

{

company temp;

input\_company(temp);

node\_company \*p = creteNode\_company(temp);

addFirstLisTCompany(List\_c,p);

}

}

// duyet danh sach tu dau den cuoi

void display1(LIST\_company List){

node\_company \*temp = List.pHead;

while(temp != NULL)

{

output\_datacompany(temp->data);

temp = temp->pNext;

}

cout<<endl;

}

// duyet tu cuoi len dau

void display2(LIST\_company List){

node\_company \*temp = List.ptail;

if(List.ptail == NULL){

return;

}

while(temp != NULL)

{

output\_datacompany(temp->data);

temp = temp->pPrev;

}

cout<<endl;

}

// tim don vi thi cong them ma don vi

void find\_id\_company (LIST\_company List){

char idFind[10];

cout<< "input id\_company find: ";fflush(stdin);gets(idFind);

for(node\_company \*i = List.pHead; i!= NULL;i = i->pNext)

{

if (strcmp(idFind,i->data.id\_company)==0)

{

output\_datacompany (i->data);

}

}

}

// sua lai don vi thi cong theo id

void fix\_id\_company (LIST\_company List){

char idFix[10];

cout<< "input id\_company fix: ";fflush(stdin);gets(idFix);

for(node\_company \*i = List.pHead; i!= NULL;i = i->pNext)

{

if (strcmp(idFix,i->data.id\_company)==0)

{

input\_company(i->data);

cout<< "fix info is successful"<<endl;

break;

}

}

}

// HAM XOA ĐẦU

void removeFirstCompany(LIST\_company &List)

{

if(List.pHead == NULL){

return;

}

List.pHead = List.pHead->pNext;

List.pHead ->pPrev = NULL;

}

// HÀM XÓA CUỐI

void removeLastCompany (LIST\_company &List)

{

if(List.pHead == NULL){

return;

}

List.ptail = List.ptail->pPrev;

List.ptail ->pNext = NULL;

}

void remove\_id(LIST\_company &List)

{

char id\_remove[10];

cout<< "nhap id don vi thi cong can xoa: ";fflush(stdin);gets(id\_remove);

if(strcmp(List.pHead->data.id\_company,id\_remove)==0)

{

removeFirstCompany(List);

return;

}

else if(strcmp(List.ptail->data.id\_company,id\_remove)==0)

{

removeLastCompany(List);

return;

}

else

{

for(node\_company \* i=List.pHead;i!=NULL;i=i->pNext)

{

if(strcmp(i->data.id\_company,id\_remove)==0)

{

i->pPrev->pNext = i->pNext;

i->pNext->pPrev = i->pPrev;

i->pPrev = NULL;

i->pNext = NULL;

delete i;

break;

}

}

}

}

void sort\_by\_cost(LIST\_company List)

{

for(node\_company \*i = List.pHead;i!=NULL;i=i->pNext)

{

for(node\_company \*j=i->pNext;j!=NULL;j=j->pNext)

{

if(i->data.cost > j->data.cost){

company temp = i->data;

i->data = j->data;

j->data = temp;

}

}

}

display1(List);

}

// ĐỌC FILE

void write\_file\_company(LIST\_company List)

{

FILE \*f;

f = fopen("file\_company.dat", "wb");

int n = 0;

node\_company\* p;

for (p = List.pHead; p != NULL; p = p->pNext)

n = n + 1;

fwrite(&n, sizeof(int), 1, f);

for (p = List.pHead; p != NULL; p = p->pNext)

fwrite(&p->data, sizeof(company), 1, f);

fclose(f);

cout << "\nSuccessfully Saved File!";

}

// GHI FILE

void read\_file\_company(LIST\_company &List)

{

FILE\* f;

company x;

int i, n;

node\_company\* p;

node\_company\* q;

f = fopen("file\_company.dat", "rb");

fread(&n, sizeof(int), 1, f);

bool check = true;

int count = 0;

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

fread(&x, sizeof(company), 1, f);

p = creteNode\_company(x);

for (q = List.pHead; q != NULL; q = q->pNext) {

if (q->data.id\_company == p->data.id\_company) {

check = false;

}

}

if (check) {

addFirstLisTCompany(List, p);

count++;

}

}

if (count > 0)

cout << "\nSuccessfully Read File!";

else

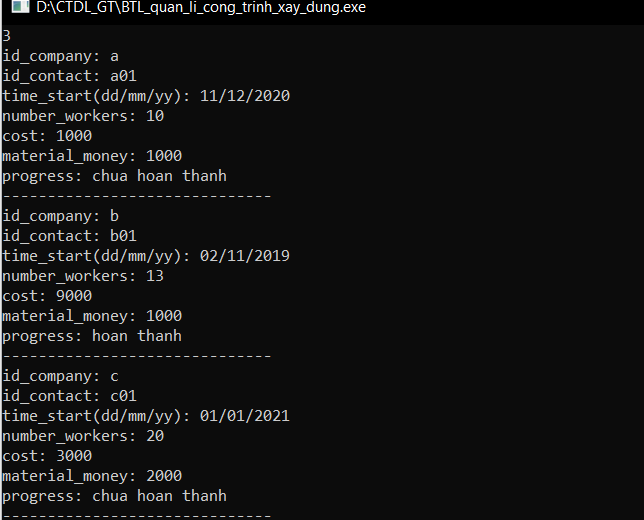
cout << "\nUnsuccessfully Read File Due To Duplicated ProfileCode!";

fclose(f);

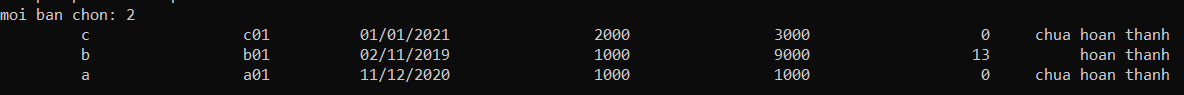
}

2. Kiểm thử

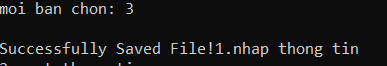
1.nhập thông tin



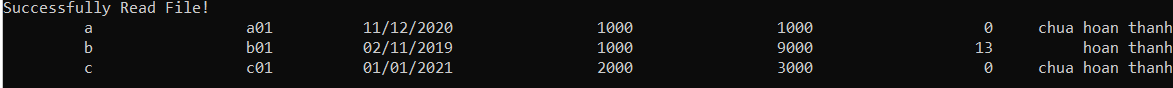
2.xuât danh sách



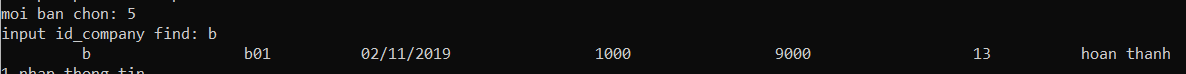
3. ghi file



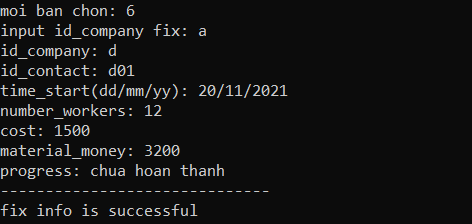
4. đọc file



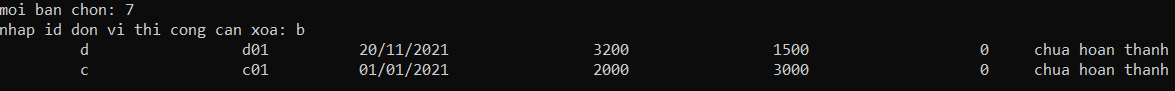
5.tim thông tin đơn vi theo id



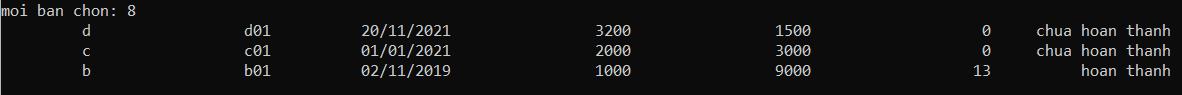
6.sủa thông tin đơn vị theo id



7.xoa thông tin đơn vị theo id



8.sắp xếp theo cost



## STACK QUẢN LÝ NHÂN VIÊN

1. Code xử lý chính:

//==================== STACK NHAN VIEN===================

struct NhanVien {

char id[10];

char fullname[10];

char id\_contact[10];

float luong\_CB,hoa\_hong;

};

void input\_nhanvien(NhanVien &nhanvien)

{

cout<< "id: ";fflush(stdin);

gets(nhanvien.id);

cout<< "fullname: ";gets(nhanvien.fullname);

cout<< "id\_contact: ";gets(nhanvien.id\_contact);

cout<< "luong co ban: ";cin>>nhanvien.luong\_CB;

cout<< "hoa hong: ";cin>>nhanvien.hoa\_hong;

cout<< "============="<<endl;

}

void output\_nhanvien(NhanVien nhanvien)

{

cout<< setw(10)<< nhanvien.id;

cout<< setw(10)<< nhanvien.fullname;

cout<< setw(10)<< nhanvien.id\_contact;

cout<< setw(10)<< nhanvien.luong\_CB;

cout<< setw(10)<< nhanvien.hoa\_hong;

cout<<endl;

}

struct node\_nhanvien{

NhanVien data;

struct node\_nhanvien\*pNext;

};

struct stack\_Listnhanvien{

node\_nhanvien\*pTop;

};

struct ListNhanvien

{

node\_nhanvien \*pHead;

node\_nhanvien \*ptail;

};

void createListNhanvien (ListNhanvien &l)

{

l.pHead = NULL;

l.ptail = NULL;

}

node\_nhanvien \*create\_nodestack(NhanVien nhanvien)

{

node\_nhanvien \*temp = new node\_nhanvien;

temp->data = nhanvien;

temp->pNext = NULL;

return temp;

}

void create\_stack\_nhanvien (stack\_Listnhanvien &s)

{

s.pTop = NULL;

}

//ktra rong

bool isEmpty(stack\_Listnhanvien s)

{

if(s.pTop == NULL)return 1;

return 0;

}

//them vao dau

void pust(stack\_Listnhanvien &s,node\_nhanvien \*temp)

{

// node\_nhanvien \*temp = create\_nodestack(nhanvien);

if(s.pTop== NULL)

s.pTop = temp;

else{

temp->pNext = s.pTop;

s.pTop = temp;

}

}

// lay thong tin va xoa no di

bool pop(stack\_Listnhanvien &s,NhanVien &x)

{

if(s.pTop == NULL)return false;

else{

node\_nhanvien \*temp= s.pTop;

x = temp ->data;

s.pTop = s.pTop ->pNext;

delete temp;

}

return true;

}

void top(stack\_Listnhanvien s)

{

if(s.pTop!=NULL)

output\_nhanvien(s.pTop->data);

}

void input\_stack(stack\_Listnhanvien &s)

{

int n;

cout<< "so luong nhan vien: ";cin>>n;

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

{

NhanVien nhanvien;

input\_nhanvien(nhanvien);

node\_nhanvien \*temp= create\_nodestack(nhanvien);

pust(s,temp);

//themvaodau(l,temp);

}

}

void themvaodau(ListNhanvien &l, node\_nhanvien\* nhanvien )

{

if(l.pHead == NULL)

l.pHead = l.ptail = nhanvien;

else

{

nhanvien->pNext = l.pHead;

l.pHead = nhanvien;

}

}

void themvaocuoi(ListNhanvien &l,node\_nhanvien \*nhanvien)

{

if(l.pHead == NULL)

l.pHead = l.ptail = nhanvien;

else{

l.ptail->pNext = nhanvien;

l.ptail= nhanvien;

}

}

void gan\_stack (stack\_Listnhanvien &s,ListNhanvien &l)

{

while(isEmpty(s)== false)

{

NhanVien temp;

pop(s,temp);

node\_nhanvien \*p = create\_nodestack(temp);

themvaodau(l,p);

}

}

void add\_stack(stack\_Listnhanvien &s,ListNhanvien &l)

{

for(node\_nhanvien \*i = l.pHead;i!=NULL;i=i->pNext)

{

pust(s,i);

}

}

void output\_stack(stack\_Listnhanvien s)

{

if(isEmpty(s) == true)

cout<< "ko co du lieu de xuat ..."<<endl;

else

{

while(isEmpty(s) == false)

{

NhanVien temp;

pop(s,temp);

output\_nhanvien(temp);

}

}

}

void write\_file\_stack(ListNhanvien l)

{

FILE \*f;

f = fopen("file\_stack.dat", "wb");

int n = 0;

node\_nhanvien\* p;

for (p = l.pHead; p != NULL; p = p->pNext)

n = n + 1;

fwrite(&n, sizeof(int), 1, f);

for (p = l.pHead; p != NULL; p = p->pNext)

fwrite(&p->data, sizeof(NhanVien), 1, f);

fclose(f);

cout << "\nSuccessfully Saved File!";

}

void read\_file\_nhanvien(ListNhanvien &l,stack\_Listnhanvien &s)

{

FILE\* f;

NhanVien x;

int i, n;

node\_nhanvien\* p;

node\_nhanvien\* q;

f = fopen("file\_company.dat", "rb");

fread(&n, sizeof(int), 1, f);

bool check = true;

int count = 0;

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

fread(&x, sizeof(NhanVien), 1, f);

p = create\_nodestack(x);

for (q = l.pHead; q != NULL; q = q->pNext) {

if (q->data.id == p->data.id) {

check = false;

}

}

if (check) {

themvaocuoi(l, p);

add\_stack(s,l);

count++;

}

}

if (count > 0)

cout << "\nSuccessfully Read File!";

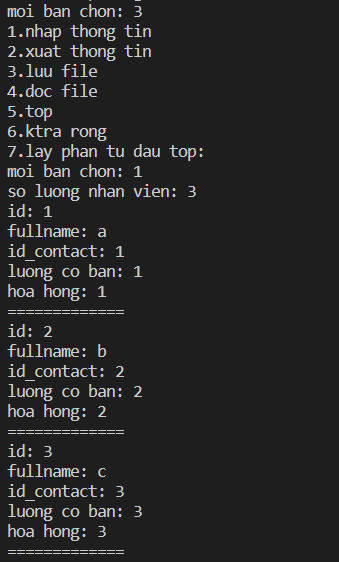
else

cout << "\nUnsuccessfully Read File Due To Duplicated ProfileCode!";

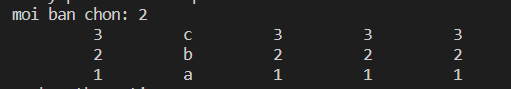
fclose(f);

}

1. Kiểm thử:
2. Nhập thông tin:



1. In thông tin:



1. Kiểm tra rỗng:



1. Top:



## QUEUE QUẢN LÝ HỢP ĐỒNG

1. Code xử lý chính:

//=================================QUEUE======================

struct hopdong{

char mahopdong [10];

char ngaykiket [10];

char maNV [10];

float so\_tien\_du\_tru,coc;

};

void nhaphopdong(hopdong &h)

{

cout<< "ma hop dong: ";fflush(stdin);

gets(h.mahopdong);

cout<< "ngay ki ket: ";gets(h.ngaykiket);

cout<< "ma nhan vien: ";gets(h.maNV);

cout<< "so tien du tru: ";cin>>h.so\_tien\_du\_tru;

cout<< "coc: ";cin>>h.coc;

cout<< "------------------"<<endl;

}

void xuathopdong(hopdong h)

{

cout<< setw(10)<<h.mahopdong;

cout<< setw(10)<<h.ngaykiket;

cout<< setw(10)<<h.maNV;

cout<< setw(10)<<h.so\_tien\_du\_tru;

cout<< setw(10)<<h.coc;

cout<<endl;

}

struct node\_hopdong{

hopdong data;

struct node\_hopdong\*pNext;

};

struct queue\_hd{

node\_hopdong\*pHead;

node\_hopdong\*ptail;

};

void khoitaoList\_queue(queue\_hd &hd){

hd.pHead = NULL;

hd.ptail = NULL;

}

node\_hopdong \*tao\_hopdong(queue\_hd &hd,hopdong hp)

{

node\_hopdong \*temp = new node\_hopdong;

temp->data = hp;

temp->pNext = NULL;

return temp;

}

bool isEmpty\_eq(queue\_hd hd)

{

if(hd.pHead == NULL) return true;

return false;

}

// co che them vao cuoi

void pust\_eq(queue\_hd &hd,node\_hopdong \*temp)

{

if(isEmpty\_eq(hd)==true)

hd.pHead = hd.ptail = temp;

else{

hd.ptail->pNext = temp;

hd.ptail = temp;

}

}

void push\_add(queue\_hd &hd)

{

hopdong x;

nhaphopdong(x);

node\_hopdong \*temp = tao\_hopdong(hd,x);

pust\_eq(hd,temp);

cout<< "done" <<endl;

}

//lay dau va xoa ra

bool pop\_eq(queue\_hd &hd,hopdong &temp)

{

if(isEmpty\_eq(hd) == true) return false;

else{

node\_hopdong \*h = hd.pHead;

temp = h->data;

hd.pHead = hd.pHead->pNext;

delete h;

}

return true;

}

void top\_eq(queue\_hd hd)

{

if(isEmpty\_eq(hd) == true)

cout<< "ko co du lieu"<<endl;

else

xuathopdong(hd.pHead->data);

}

void nhapds(queue\_hd &hd)

{

int n;

cout<< "nhap danh sach :";cin>>n;

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

{

hopdong h;

nhaphopdong(h);

node\_hopdong \*temp = tao\_hopdong(hd,h);

pust\_eq(hd,temp);

}

}

void xuatds(queue\_hd &hd)

{

while(isEmpty\_eq(hd)== false)

{

hopdong a;

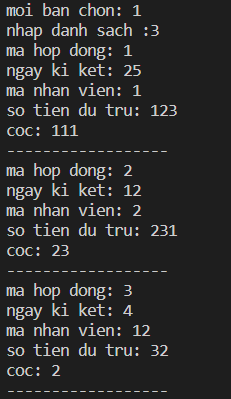
pop\_eq(hd,a);

xuathopdong(a);

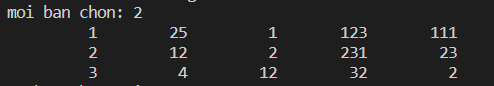
}

}

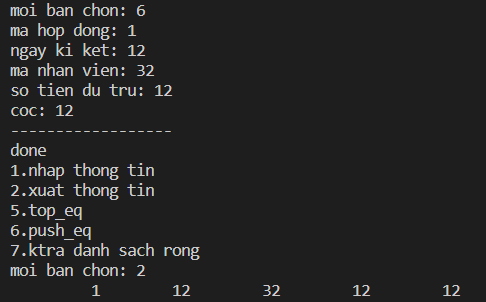
1. Kiểm thử:
2. Nhập



1. In:



1. Push:



THE END