Đại Học Bách Khoa Hà Nội

Trường Điện - Điện Tử

======o0o======

BÁO CÁO MÔN HỌC

Cấu trúc dữ liệu và giải thuật

Đề tài:

Hệ thống quản lí kho hàng

Thông tin thành viên nhóm thực hiện :

HỌ VÀ TÊN	MSSV
Vũ Việt Anh	20233996
Đỗ Thanh Bình	20239668
Vũ Trí Dũng	20234003
Nguyễn Quốc Huy	20234013

Hà Nội, 06/2025

Thành viên	Nhiệm vụ chính	Chi tiết công việc
Vũ Việt Anh	Tổng quan hệ thống	- Tổng quan các chức năng - Định nghĩa cấu trúc cho hàng hóa và phiếu - Hiển thị hàng hóa - Menu
Đỗ Thanh Bình	Thống kê và lưu trữ dữ liệu	- Thống kê hàng hóa (8 loại) - Readflie và savefile
Vũ Trí Dũng	Quản lí phiếu nhập và xuất kho	- Thêm phiếu nhập/xuất - Cập nhật số liệu hang tồn - Lưu trữ danh sách phiếu - Hiển thị Phiếu
Nguyễn Quốc Huy	Quản lý hàng hóa	- Sửa thông tin hàng hóa - Xóa thông tin hàng hóa - Tìm kiếm hàng hóa (4 loại)

1. Menu:

```
void menu(){
    productList = NULL;
    readFile("baocao.txt");
    readReceiptFromFile("phieu.txt");
    int number;
    do {
        printf( "\n-----------------------------------------\n" ) ;
        printf("=============QUAN LI KHO HANG==============\n");
        printf("-----------------------------------------\n");
        printf ("[1]. Them hang hoa.\n");
        printf ("[2]. Them phieu nhap/xuat.\n");
        printf ("[3]. Sua thong tin hang hoa.\n") ;
        printf ("[4]. Tim kiem hang hoa.\n");
        printf ("[5]. Xoa hang hoa.\n");
        printf ("[6]. Hien thi hang hoa/phieu.\n" );
        printf ("[7]. Thong ke hang hoa.\n" );
        printf ("[0]. Thoat chuong trinh.\n");
        printf ("Chon tinh nang: ");
        scanf("%d", &number);
        getchar();
```

```
        switch (number) {
            case 1:
                system("cls");
                addProduct();
                break;
            case 2:
                system("cls");
                int loaiPhieu;
                printf("Chon loai phieu:\n");
                printf("[1] Phieu nhap kho\n");
                printf("[2] Phieu xuat kho\n ");
                printf("Lua chon:");
                scanf("%d", &loaiPhieu);
                getchar();

                Receipt p;
                if (loaiPhieu == 1) {
                p = IMPORTReceipt();
                    runReceipt(&p);
                } else if (loaiPhieu == 2) {
                  p = EXPORTReceipt();
                    runReceipt(&p);
                } else {
                printf("Lua chon khong hop le.\n");
                break;
               }

                // Thêm phiếu vào danh sách
                addReceipt(p);
                printf("Da them va xu ly phieu thanh cong.\n");
                break;
            case 3:
                system("cls");
                editProduct();
                break;
            case 4:
                system("cls");
                searchProduct();
                break;
            case 5:
                system("cls");
                deleteProduct();
                break;
            case 6:
                system("cls");
                int choice;
                printf("Chon hien thi:\n");
```

```
                printf("[1].Hien thi hang hoa.\n");
                printf("[2].Hien thi chi tiet cac phieu.\n");
                printf("Lua chon: ");
                scanf("%d", &choice);
                getchar();
                if (choice == 1) {
                    displayProducts();}
                else if (choice == 2) {
                    displayReceipt();
                }
                else {
                     printf("Lua chon khong hop le.\n");
                }
                break;
            case 7:
                system("cls");
                int choices;
                printf("Chon thong ke:\n");
                printf("[1].Thong ke hang hoa.\n");
                printf("[2].Thong ke so phieu.\n");
                printf("Lua chon: ");
                scanf("%d", &choices);
                getchar();
                if (choices == 1) {
                    statisticsProduct();}
                else if (choices == 2) {
                    statisticsReceipt();
                }
                else {
                     printf("Lua chon khong hop le.\n");
                }
                break;

            case 0:
                saveFile("baocao.txt");
                saveReceiptToFile("phieu.txt");
                printf("Thoat chuong trinh.\n") ;
                printf("Cam on ban da su dung chuong trinh!\n");
                break;
            default:
                system("cls");
                printf("Lua chon khong hop le. Vui long chon lai!\n");
                break;
        }
    } while (number != 0);
}
```

2. Tạo hàng hóa: Cho phép người dùng nhập thông tin sản phẩm và thêm vào danh sách với ID tự tạo.

```
// Hàm tạo ID sản phẩm tự động dạng SP001, SP002,...
char* createProductID(ProductNode* head) {
    int maxID = 0;
    while (head != NULL) {
        int num = 0;
        sscanf(head->data.ID, "SP%d", &num);
        if (num > maxID) maxID = num;
        head = head->next;
    }

    char* newID = (char*)malloc(10);
    sprintf(newID, "SP%03d", maxID + 1);
    return newID;
}

// Hàm tạo sản phẩm mới
Product createProduct() {
    Product p;
    char* idMoi = createProductID(productList);
    strcpy(p.ID, idMoi);
    free(idMoi);

    printf("ID tu dong tao: %s\n", p.ID);
    printf("Nhap ten: "); inputString(p.Name, sizeof(p.Name));
    printf("Nhap don vi tinh: "); inputString(p.unitOfMeasurement,
sizeof(p.unitOfMeasurement));
    printf("Nhap nha cung cap: "); inputString(p.Supplier,
sizeof(p.Supplier));
    printf("Nhap so luong dang co: "); scanf("%d", &p.Quantity);
    getchar();
    printf("Nhap don gia: "); scanf("%f", &p.Price);
    getchar();
    return p;
}

// Thêm sản phẩm vào danh sách (sắp xếp theo ID tăng dần)
void addProduct() {
    Product p = createProduct();
    ProductNode* newNode = (ProductNode*)malloc(sizeof(ProductNode));
    newNode->data = p;
    newNode->next = NULL;
```

```
    if (productList == NULL) {
        productList = newNode;  // Nếu danh sách rỗng, gán luôn
    } else {
        ProductNode* current = productList;
        while (current->next != NULL) {
            current = current->next;  // Duyệt đến cuối danh sách
        }
        current->next = newNode;  // Thêm vào cuối
    }

    printf("Da them san pham thanh cong!\n");
}
```

3. Tạo phiếu nhập xuất: Cho phép tạo phiếu nhập hoặc phiếu xuất và xử lý cập nhật vào kho.

```
// phiếu nhập
Receipt IMPORTReceipt() {
    Receipt p;
    strcpy(p.type, "IMPORT");
     p.isProcessed = 0; //  đánh dấu chưa xử lý
    printf("=== Tao Phieu Nhap Kho ===\n");
    printf("Nhap ma hang (ID): "); inputString(p.ID, sizeof(p.ID));
    printf("Nhap so luong: "); scanf("%d", &p.quantity);
    getchar();
    ProductNode* prod = productList;
    while (prod) {
        if (strcmp(prod->data.ID, p.ID) == 0) {
            strcpy(p.name, prod->data.Name);
            p.price = prod->data.Price;
            break;
        }
        prod = prod->next;
    }
    if (prod == NULL) {
        strcpy(p.name, "");
        p.price = 0;
    }
    printf("Nhap ngay (dd/mm/yyyy): ");
    inputString(p.date, sizeof(p.date));
    return p;
}
// phiếu xuất
Receipt EXPORTReceipt() {
    Receipt p;
```

```
    strcpy(p.type, "EXPORT");
     p.isProcessed = 0; //  đánh dấu chưa xử lý
    printf("=== Tao Phieu Xuat Kho ===\n");
    printf("Nhap ma hang (ID): "); inputString(p.ID, sizeof(p.ID));
    printf("Nhap so luong: "); scanf("%d", &p.quantity);
    getchar();
    ProductNode* prod = productList;
    while (prod) {
        if (strcmp(prod->data.ID, p.ID) == 0) {
            strcpy(p.name, prod->data.Name);
            p.price = prod->data.Price;
            break;
        }
        prod = prod->next;
    }
    if (prod == NULL) {
        strcpy(p.name, "");
        p.price = 0;
    }
    printf("Nhap ngay (dd/mm/yyyy): ");
    inputString(p.date, sizeof(p.date));
    return p;
}
void runReceipt(Receipt* p) {
    if (p->isProcessed == 1) // tránh xử lý lại phiếu đã xử lý
    return;

    ProductNode* prod = productList;
    while (prod) {
        if (strcmp(prod->data.ID, p->ID) == 0) {
            if (strcmp(p->type, "IMPORT") == 0) {
                prod->data.Quantity += p->quantity;
                p->isSuccess = 1;
            } else if (strcmp(p->type, "EXPORT") == 0) {
                if (prod->data.Quantity >= p->quantity) {
                    prod->data.Quantity -= p->quantity;
                    p->isSuccess = 1;
                } else {
                    printf("Canh bao: Khong du hang de xuat kho (ID: %s, Yeu
cau: %d, Ton kho: %d)\n",
                           p->ID, p->quantity, prod->data.Quantity);
                    p->isSuccess = 0;
                    // Không trừ số lượng khi thất bại
                }
            }
            break;
        }
```

```
        prod = prod->next;
    }
    p->isProcessed = 1; // đánh dấu đã xử lý
}
```

4. Sửa thông tin hang hóa: Lựa chọn 1 thông tin của sản phẩm và cập nhật lại thông tin của sản phẩm.

```
//Hàm sửa thông tin sản phẩm
void editProduct(){
    char id[10];
    printf("Nhap ID san pham can sua: ");
    inputString(id, sizeof(id));

    ProductNode* current = productList;
    while (current) {
        if (strcmp(current->data.ID, id) == 0) {
            int choice;
            printf("Da tim thay san pham: %s\n", current->data.Name);
            printf("Chon thong tin muon sua:\n");
            printf("[1] Ten\n");
            printf("[2] Don vi tinh\n");
            printf("[3] Nha cung cap\n");
            printf("[4] So luong\n");
            printf("[5] Don gia\n");
            printf("Lua chon: ");
            scanf("%d", &choice);
            getchar();
            switch (choice) {
            case 1:
                printf("Nhap ten moi: ");
                inputString(current->data.Name, sizeof(current->data.Name));
                break;
            case 2:
                printf("Nhap don vi tinh moi: ");
                inputString(current->data.unitOfMeasurement,
sizeof(current->data.unitOfMeasurement));
                break;
            case 3:
                printf("Nhap nha cung cap moi: ");
                inputString(current->data.Supplier,
sizeof(current->data.Supplier));
                break;
            case 4:
                printf("Nhap so luong moi: ");
                scanf("%d", &current->data.Quantity);
```

```
                getchar();
                break;
            case 5:
                printf("Nhap don gia moi: ");
                scanf("%f", &current->data.Price);
                getchar();
                break;
            default:
                printf("Lua chon khong hop le!\n");
                return;
            }
            printf("Da cap nhat thong tin san pham!\n");
            return;
        }
        current = current->next;
    }
    printf("Khong tim thay san pham co ID: %s\n", id);
}
```

5. Tìm kiếm hàng hóa: Tìm sản phẩm theo ID, tên, nhà cung cấp, theo khoảng đơn giá hoặc theo khoảng số lượng

```
//Hàm tìm sản phẩm
void searchProduct() {
    int choice;
    printf("[1] Theo ID hoac ten\n");
    printf("[2] Theo nha cung cap\n");
    printf("[3] Theo khoang don gia\n");
    printf("[4] Theo khoang so luong\n");
    printf("Lua chon: ");
    scanf("%d", &choice);
    getchar();

    ProductNode* current = productList;
    int found = 0;

    switch (choice) {
    case 1: {
        char keyword[50];
        printf("Nhap ID hoac ten san pham: ");
        inputString(keyword, sizeof(keyword));
        while (current) {
            Product p = current->data;
            if (strcmp(p.ID, keyword) == 0 || strstr(p.Name, keyword) != NULL)
{
                if (!found) {
```

```
                    printf("\n%-10s %-25s %-15s %-25s %-10s %-12s\n",
                           "ID", "Ten", "Don vi tinh", "Nha cung cap", "So
luong", "Don gia");
                    found = 1;
                }
                printf("%-10s %-25s %-15s %-25s %-10d %-12.2f\n",
                       p.ID, p.Name, p.unitOfMeasurement, p.Supplier,
p.Quantity, p.Price);
            }
            current = current->next;
        }
        break;
    }

    case 2: {
        char supplier[50];
        printf("Nhap ten nha cung cap: ");
        inputString(supplier, sizeof(supplier));
        while (current) {
            Product p = current->data;
            if (strstr(p.Supplier, supplier)) {
                if (!found) {
                    printf("\n%-10s %-25s %-15s %-25s %-10s %-12s\n",
                           "ID", "Ten", "Don vi tinh", "Nha cung cap", "So
luong", "Don gia");
                    found = 1;
                }
                printf("%-10s %-25s %-15s %-25s %-10d %-12.2f\n",
                       p.ID, p.Name, p.unitOfMeasurement, p.Supplier,
p.Quantity, p.Price);
            }
            current = current->next;
        }
        break;
    }

    case 3: {
        float minPrice, maxPrice;
        printf("Nhap don gia thap nhat: ");
        scanf("%f", &minPrice);
        printf("Nhap don gia cao nhat: ");
        scanf("%f", &maxPrice);
        getchar();

        while (current) {
            Product p = current->data;
            if (p.Price >= minPrice && p.Price <= maxPrice) {
```

```
                if (!found) {
                    printf("\n%-10s %-25s %-15s %-25s %-10s %-12s\n",
                           "ID", "Ten", "Don vi tinh", "Nha cung cap", "So
luong", "Don gia");
                    found = 1;
                }
                printf("%-10s %-25s %-15s %-25s %-10d %-12.2f\n",
                       p.ID, p.Name, p.unitOfMeasurement, p.Supplier,
p.Quantity, p.Price);
            }
            current = current->next;
        }
        break;
    }

    case 4: {
        int minQty, maxQty;
        printf("Nhap so luong thap nhat: ");
        scanf("%d", &minQty);
        printf("Nhap so luong cao nhat: ");
        scanf("%d", &maxQty);
        getchar();

        while (current) {
            Product p = current->data;
            if (p.Quantity >= minQty && p.Quantity <= maxQty) {
                if (!found) {
                    printf("\n%-10s %-25s %-15s %-25s %-10s %-12s\n",
                           "ID", "Ten", "Don vi tinh", "Nha cung cap", "So
luong", "Don gia");
                    found = 1;
                }
                printf("%-10s %-25s %-15s %-25s %-10d %-12.2f\n",
                       p.ID, p.Name, p.unitOfMeasurement, p.Supplier,
p.Quantity, p.Price);
            }
            current = current->next;
        }
        break;
    }

    default:
        printf("Lua chon khong hop le.\n");
        return;
    }

    if (!found) {
```

```
        printf("Khong tim thay san pham.\n");
    }
}
```

6. Xóa hàng hóa: Xóa sản phẩm khỏi danh sách.

```
// Hàm xóa sản phẩm
void deleteProduct(){
    char id[10];
    printf("Nhap ID san pham muon xoa: ");
    inputString(id, sizeof(id));

    ProductNode* current = productList;
    ProductNode* prev = NULL;

    while (current) {
        if (strcmp(current->data.ID, id) == 0) {
            if (prev == NULL) {
                productList = current->next;
            }
            else {
                prev->next = current->next;
            }
            free(current);
            printf("Da xoa san pham co ID: %s\n", id);
            return;
        }
        prev = current;
        current = current->next;
    }
    printf("Khong tim thay san pham co ID: %s\n", id);
}
```

7. Hiển thị hang hóa/phiếu:

- Hiển thị toàn bộ sản phẩm

```
void displayProducts() {
    ProductNode* current = productList;
    printf("\n%-10s | %-20s | %-10s | %-20s | %-10s | %-12s | %-12s\n",
           "ID", "Ten sp", "Don vi", "Nha cung cap", "So luong", "Gia",
"Tong");
    while (current) {
        Product p = current->data;
        float total = p.Quantity * p.Price;
        printf("%-10s | %-20s | %-10s | %-20s | %-10d | %-12.2f | %-12.2f\n",
```

```
                p.ID, p.Name, p.unitOfMeasurement, p.Supplier, p.Quantity,
p.Price, total);
        current = current->next;
    }
}
```

- Hiển thị toàn bộ phiếu:

```
void displayReceipt() {
    if (receiptList == NULL) {
        printf("Khong co phieu nao trong danh sach.\n");
        return;
    }

    printf("\n================ CHI TIET CAC PHIEU ================\n");
    printf("%-10s | %-25s | %-12s | %-10s | %-15s | %-12s | %-12s | %-15s\n",
           "Ma SP", "Ten SP", "Loai phieu", "So luong", "Ngay", "Don gia",
"Thanh tien", "Trang thai");

    ReceiptNode* current = receiptList;
    while (current != NULL) {
        Receipt r = current->data;
        float thanhTien = r.quantity * r.price;
        const char* loaiPhieu = strcmp(r.type, "IMPORT") == 0 ? "NHAP" :
"XUAT";
        const char* trangThai = (r.isSuccess == 1) ? "THANH CONG" : "KHONG
THANH CONG";

        printf("%-10s | %-25s | %-12s | %-10d | %-15s | %-12.2f | %-12.2f |
%-15s\n",
                r.ID, r.name, loaiPhieu, r.quantity, r.date, r.price,
thanhTien, trangThai);
        current = current->next;
    }
}
```

8. Thống kê:

- Thống kê hàng hóa: Tổng số loại, số lượng, giá trị, nhà cung cấp, sản phẩm sắp hết, sắp xếp theo giá trị tồn kho hoặc số lượng, giá trị trung bình của các mặt hàng

```
void statisticsProduct() {
    int choice;
    if (productList == NULL) {
        printf("Danh sach san pham rong.\n");
        return;
```

```
    }

    int tongLoai = 0;
    int tongSoLuong = 0;
    float tongGiaTri = 0;

    ProductNode* current = productList;
    while (current != NULL) {
        tongLoai++;
        tongSoLuong += current->data.Quantity;
        tongGiaTri += current->data.Quantity * current->data.Price;
        current = current->next;
    }

    do{
        printf("\n===== THONG KE HANG HOA =====\n");
        printf("[1]. Tong so loai mat hang\n");
        printf("[2]. Tong so luong hang\n");
        printf("[3]. Tong gia tri hang ton kho\n");
        printf("[4]. Thong ke theo nha cung cap\n");
        printf("[5]. Thong ke theo gia tri ton kho giam dan\n");
        printf("[6]. San pham sap het hang (SL < 10)\n");
        printf("[7]. Gia tri trung binh cua 1 mat hang\n");
        printf("[8]. Thong ke theo so luong giam dan\n");
        printf("[0]. Thoat\n");
        printf("Lua chon: ");
        scanf("%d", &choice);
        getchar();

        switch (choice) {
            case 1:
                printf("Tong so loai mat hang: %d\n", tongLoai);
                break;
            case 2:
                printf("Tong so luong hang: %d\n", tongSoLuong);
                break;
            case 3:
                printf("Tong gia tri hang ton kho: %.2f\n", tongGiaTri);
                break;
            case 4: {
                printf("\n--- Thong ke theo nha cung cap ---\n");
                ProductNode* outer = productList;
                while (outer != NULL) {
                    int daThongKe = 0;
                    ProductNode* check = productList;
                    while (check != outer) {
```

```
                        if (strcmp(check->data.Supplier, outer->data.Supplier)
== 0) {
                            daThongKe = 1;
                            break;
                        }
                        check = check->next;
                    }
                    if (!daThongKe) {
                        int sl = 0;
                        float gt = 0;
                        ProductNode* inner = productList;
                        while (inner != NULL) {
                            if (strcmp(inner->data.Supplier,
outer->data.Supplier) == 0) {
                                sl += inner->data.Quantity;
                                gt += inner->data.Quantity *
inner->data.Price;
                            }
                            inner = inner->next;
                        }
                        printf("NCC: %s | SL: %d | Gia tri: %.2f\n",
outer->data.Supplier, sl, gt);
                    }
                    outer = outer->next;
                }
                break;
            }
            case 5: {
    printf("\n--- Sap xep san pham theo gia tri ton kho Giam Dan ---\n");

    //Dem so phan tu
    int count = 0;
    ProductNode* temp = productList;
    while (temp != NULL) {
        count++;
        temp = temp->next;
    }

    if (count == 0) {
        printf("Khong co san pham de thong ke.\n");
        break;
    }

    // Copy danh sach vao mang de sap xep
    Product* arr = (Product*)malloc(count * sizeof(Product));
    temp = productList;
    for (int i = 0; i < count; i++) {
```

```
        arr[i] = temp->data;
        temp = temp->next;
    }

    //Sap xep giam dan theo gia tri ton kho
    for (int i = 0; i < count - 1; i++) {
        for (int j = i + 1; j < count; j++) {
            float val1 = arr[i].Quantity * arr[i].Price;
            float val2 = arr[j].Quantity * arr[j].Price;
            if (val1 < val2) {
                Product tmp = arr[i];
                arr[i] = arr[j];
                arr[j] = tmp;
            }
        }
    }

    //ghi ket qua ra file thongke.txt
    FILE* f = fopen("thongke.txt", "w");
    if (f == NULL) {
        printf("Khong the mo file thongke.txt de ghi.\n");
        free(arr);
        break;
    }

    fprintf(f, "DANH SACH SAN PHAM SAP XEP THEO GIA TRI TON KHO GIAM DAN:\n");
    fprintf(f, "%-10s %-20s %-15s %-20s %-10s %-10s %-15s\n",
            "ID", "Ten", "Don vi", "Nha cung cap", "So luong", "Don gia", "Gia
tri ton kho");

    for (int i = 0; i < count; i++) {
        float giaTri = arr[i].Quantity * arr[i].Price;
        fprintf(f, "%-10s %-20s %-15s %-20s %-10d %-10.2f %-15.2f\n",
                arr[i].ID, arr[i].Name, arr[i].unitOfMeasurement,
arr[i].Supplier,
                arr[i].Quantity, arr[i].Price, giaTri);
    }

    fclose(f);
    printf("Da ghi ket qua thong ke vao file 'thongke.txt'.\n");

    //in ra man hinh
    for (int i = 0; i < count; i++) {
        printf("%-10s | %-20s | SL: %-3d | Don gia: %.2f | Gia tri: %.2f\n",
               arr[i].ID, arr[i].Name, arr[i].Quantity, arr[i].Price,
               arr[i].Quantity * arr[i].Price);
    }
```

```
    free(arr);
    break;
}
            case 6: {
                printf("\n--- San pham sap het hang (SL < 10) ---\n");
                ProductNode* ptr = productList;
                int found = 0;
                while (ptr != NULL) {
                    if (ptr->data.Quantity < 10) {
                        printf("ID: %s | Ten: %s | SL: %d\n",
                               ptr->data.ID, ptr->data.Name,
ptr->data.Quantity);
                        found = 1;
                    }
                    ptr = ptr->next;
                }
                if (!found)
                    printf("Khong co san pham nao sap het hang.\n");
                break;
            }
            case 7: {
                if (tongLoai == 0) {
                    printf("Khong co mat hang nao de tinh trung binh.\n");
                } else {
                    float avg = tongGiaTri / tongLoai;
                    printf("Gia tri trung binh cua 1 mat hang: %.2f\n", avg);
                }
                break;
            }
            case 8: {
    printf("\n--- Sap xep san pham theo so luong Giam Dan ---\n");

    //dem so san pham
    int count = 0;
    ProductNode* temp = productList;
    while (temp != NULL) {
        count++;
        temp = temp->next;
    }

    if (count == 0) {
        printf("Khong co san pham de thong ke.\n");
        break;
    }

    //sao chep du lieu sang mang de sap xep
```

```
    Product* arr = (Product*)malloc(count * sizeof(Product));
    temp = productList;
    for (int i = 0; i < count; i++) {
        arr[i] = temp->data;
        temp = temp->next;
    }

    //sap xep giam dan theo so luong
    for (int i = 0; i < count - 1; i++) {
        for (int j = i + 1; j < count; j++) {
            if (arr[i].Quantity < arr[j].Quantity) {
                Product tmp = arr[i];
                arr[i] = arr[j];
                arr[j] = tmp;
            }
        }
    }

    //ghi vao file soluong.txt
    FILE* f = fopen("soluong.txt", "w");
    if (f == NULL) {
        printf("Khong the mo file soluong.txt de ghi.\n");
        free(arr);
        break;
    }

    fprintf(f, "DANH SACH SAN PHAM SAP XEP THEO SO LUONG GIAM DAN:\n");
    fprintf(f, "%-10s %-20s %-15s %-20s %-10s %-10s\n",
            "ID", "Ten", "Don vi", "Nha cung cap", "So luong", "Don gia");

    for (int i = 0; i < count; i++) {
        fprintf(f, "%-10s %-20s %-15s %-20s %-10d %-10.2f\n",
                arr[i].ID, arr[i].Name, arr[i].unitOfMeasurement,
arr[i].Supplier,
                arr[i].Quantity, arr[i].Price);
    }

    fclose(f);
    printf("Da ghi ket qua vao file 'soluong.txt'.\n");

    //in ra man hinh
    for (int i = 0; i < count; i++) {
        printf("%-10s | %-20s | So luong: %-4d | Don gia: %.2f\n",
               arr[i].ID, arr[i].Name, arr[i].Quantity, arr[i].Price);
    }

    free(arr);
```

```
    break;
}
            case 0:
                printf("Thoat thong ke.\n");
                break;
            default:
                printf("Lua chon khong hop le. Vui long chon lai!\n");
                break;
        }
    } while (choice != 0);
}
```

- Thống kê số phiếu: Đếm số lượng phiếu theo từng loại.

```
//Thong ke so phieu
void statisticsReceipt() {
    int soPhieuNhap = 0;
    int soPhieuXuat = 0;
    ReceiptNode* current = receiptList;
    while (current) {
        Receipt* p = &current->data;
        // Đếm số phiếu
        if (strcmp(p->type, "IMPORT") == 0) {
            soPhieuNhap++;
        } else if (strcmp(p->type, "EXPORT") == 0) {
            soPhieuXuat++;
        }
        current = current->next;
    }
    printf("\n===== Thong ke phieu =====\n");
    printf("So phieu nhap: %d\n", soPhieuNhap);
    printf("So phieu xuat: %d\n", soPhieuXuat);
}
```

9. Lưu trữ sau mỗi lần sử dụng: lưu cả hang hóa và phiếu

- Hàm saveFile: có chức năng lưu lại các dữ liệu của chương trình cho lần chạy tiếp theo.
- Hàm readFile: có chức năng đọc dữ liệu được lưu từ lần chạy trước

```
//ghi du lieu ra file
void saveFile(const char* tenFile) {
    FILE* f = fopen(tenFile, "w");
    if (f == NULL) {
        printf("Khong the mo file %s de ghi du lieu.\n", tenFile);
        return;
    }
```

```
    ProductNode* current = productList;
    while (current != NULL) {
        fprintf(f, "%s,%s,%s,%s,%d,%.2f\n",
                current->data.ID,
                current->data.Name,
                current->data.unitOfMeasurement,
                current->data.Supplier,
                current->data.Quantity,
                current->data.Price);
        current = current->next;
    }

    fclose(f);
    printf("Da luu du lieu hang hoa vao file '%s'.\n", tenFile);
}

//doc phieu tu file
void readReceiptFromFile(const char* tenFile) {
    FILE* f = fopen(tenFile, "r"); // ĐÚNG: m? d? d?c
    if (f == NULL) {
        printf("Khong the mo file %s de doc du lieu.\n", tenFile);
        return;
    }

    while (!feof(f)) {
        Receipt r;
        if (fscanf(f, "%[^,],%[^,],%d,%[^,],%f,%[^,],%d\n",
                   r.ID, r.type, &r.quantity, r.date,
                   &r.price, r.name, &r.isProcessed) == 7) {
            ReceiptNode* newNode = (ReceiptNode*)malloc(sizeof(ReceiptNode));
            newNode->data = r;
            newNode->next = NULL;

            if (receiptList == NULL) {
                receiptList = newNode;
            } else {
                ReceiptNode* temp = receiptList;
                while (temp->next != NULL)
                    temp = temp->next;
                temp->next = newNode;
            }
        }
    }

    fclose(f);
    printf("Da doc danh sach phieu tu file '%s'.\n", tenFile);
}
```

```
//ghi phieu ra File
void saveReceiptToFile(const char* tenFile) {
    FILE* f = fopen(tenFile, "w");
    if (f == NULL) {
        printf("Khong the mo file %s de ghi.\n", tenFile);
        return;
    }

    ReceiptNode* current = receiptList;
    while (current != NULL) {
        fprintf(f, "%s,%s,%d,%s,%.2f,%s,%d\n",
                current->data.ID,
                current->data.type,
                current->data.quantity,
                current->data.date,
                current->data.price,
                current->data.name,
                current->data.isProcessed);
        current = current->next;
    }

    fclose(f);
    printf("Da luu danh sach phieu ra file '%s'.\n", tenFile);
}
```