Nội dung

* Danh sách liên kết đơn
* Ví dụ quản lý danh sách sinh viên

**Ví dụ 1**. Minh họa các thao tác cơ bản trên danh sách liên kết đơn với mỗi phần tử là kiểu double

|  |
| --- |
| #include<stdio.h>  #include<stdlib.h>    struct Node  {  double data;  struct Node \*next;  };    struct Node \*makeNode(double value)  {  struct Node \*newNode = (struct Node \*)malloc(sizeof(struct Node));  newNode->data = value;  newNode->next = NULL;  return newNode;  }  // ham them nut vao dau danh sach  void insertFirst(struct Node \*\*head, double value)  {  struct Node \*newNode = makeNode(value);  newNode->next = \*head;  \*head = newNode;  }  // ham them nut vao dau danh sach  struct Node \* insertFirst2(struct Node \*head, double value)  {  struct Node \*newNode = makeNode(value);  newNode->next = head;  head = newNode;  return head;  }  // ham in danh sach  void printList(struct Node \*head)  {  printf("Danh sach cac phan tu:\n");  while(head!=NULL)  {  printf("%.2lf,",head->data);  head = head->next;  }  printf("\n");  }    // in danh sach theo thu tu nguoc  void printList\_rev(struct Node \*head)  {  if(head==NULL) return;  printList\_rev(head->next);  printf("%.2lf,",head->data);  }  // ham dem so phan tu trong danh sach  int countNodes(struct Node \*head)  {  int count=0;  while(head!=NULL)  {  count++;  head = head->next;  }  return count;  }  // tim phan tu co khoa K trong danh sach  struct Node \* searchKey(struct Node \*head, double k)  {  while(head!=NULL)  {  if(head->data==k) return head;  head = head->next;  }  return NULL;  }    struct Node \* searchKey\_Rec(struct Node \*head, double k)  {  if(head == NULL) return NULL;  if(head->data==k) return head;  return searchKey\_Rec(head->next, k);  }    // ham chen phan tu moi vao sau vi tri duoc tro boi con tro p  // p phai khac NULL  void insertAfter(struct Node \*p, double value)  {  struct Node \*newNode = makeNode(value);  newNode->next = p->next;  p->next = newNode;  }    // chen phan tu vao cuoi danh sach  // head phai khac NULL  void insertLast(struct Node \*head, double value)  {  while(head->next!=NULL)head=head->next;  insertAfter(head,value);  }    void insertLast\_Rec(struct Node \*head, double value)  {  if(head->next!=NULL)insertLast\_Rec(head->next,value);  else insertAfter(head,value);  }    // ham xoa phan tu gia tri k trong danh sach  // xoa dau hoac xoa giua/cuoi  void removeKey(struct Node \*\*head, double k)  {  if((\*head)==NULL) return;  struct Node \*p; // con tro toi phan tu can xoa  // neu khoa can xoa nam dung phan tu dau  if((\*head)->data==k)  {  p = \*head;  \*head = (\*head)->next;  free(p);  }  else // xoa giua/cuoi  {  // tim phan tu can xoa  p = \*head;  while(p!=NULL)  {  if(p->data==k) break;  p=p->next;  }  // khoa can xoa KHONG ton tai trong danh sach  if(p==NULL) return;  // tim vi tri truoc vi tri can xoa  struct Node \*pre = \*head;  while(pre->next!=p) pre = pre->next;  // ngat p khoi danh sach  pre->next = p->next;  free(p);  }  }  void destroyList(struct Node \*\*head)  {  struct Node \*p;  while((\*head)!=NULL)  {  p=\*head;  \*head = (\*head)->next;  free(p);  }  \*head = NULL;  }  int main()  {  struct Node \*head = NULL;  insertFirst(&head, 12.5);  head = insertFirst2(head,17.6);  insertFirst(&head, 2.5);  struct Node \*p = searchKey(head, 17.6);  if(p!=NULL)insertAfter(p, 22.2);  insertLast(head,11.11);  insertLast\_Rec(head,12.12);  printList(head);  printf("Danh sach hien co %d phan tu\n",countNodes(head));  printf("In danh sach theo thu tu nguoc\n");  printList\_rev(head);  printf("\n");  printf("\nXoa 2.5\n");  removeKey(&head, 2.5);  printList(head);  printf("Danh sach hien co %d phan tu\n",countNodes(head));  printf("\nXoa 12.12\n");  removeKey(&head, 12.12);  printList(head);  printf("Danh sach hien co %d phan tu\n",countNodes(head));  printf("\nXoa toan bo danh sach\n");  destroyList(&head);  printList(head);  printf("Danh sach hien co %d phan tu\n",countNodes(head));  return 0;  } |

**Bài tập 1. Chương trình quản lý bán hàng**

Quản lý sản phẩm và số lượng bằng danh sách liên kết đơn

Thông tin sản phẩm gồm

* Mã sản phẩm: char[10]
* Tên sản phẩm: char[250]
* Hãng sản xuất: char [100]
* Số lượng: int

Danh sách các sản phẩm sẽ được đọc từ file text với format dạng

|  |
| --- |
| DG01 Dau goi dau Clear 500ml  Unilever  10 DG02  Dau goi dau Clear 1000ml  Unilever  2  DG03  Dau goi dau Sunsilk 300ml  Unilever  15  DR01  Kem danh rang colgate 200gr  Unilever  12  DR02  Ban chai rang colgate  Unilever  6 |

Các thao tác cần xử lý là:

* Bổ sung một sản phẩm vào danh sách (nhập từ bàn phím hoặc từ file)
  + Sản phẩm nếu đã có mã thì tăng số lượng lên
  + Nếu mã chưa tồn tại (sp mới nhập về) bổ sung thêm vào danh sách - thêm vào cuối, đầu, giữa đều OK
* Xuất kho (bán hàng)
  + Kiểm tra mã sản phẩm đã có mã tồn tại trong danh sách và số lượng đủ để xuất
  + Nếu số lượng kho > số lượng xuất --> trừ số lượng tương ứng
  + Nếu số lượng kho == số lượng xuất --> loại bỏ sản phẩm hết khỏi danh sách
* Kiểm tra sản phẩm
  + Nhập vào 1 mã, kiểm tra xem mã sản phẩm đó còn trong kho hay không
  + Nếu còn thì in ra số lượng, ngược lại báo hết
* In ra toàn bộ sản phẩm hiện có trong kho dưới dạng bảng  
  STT | tên sản phẩm | hãng sản xuất | số lượng trong kho
* Ghi danh sách sản phẩm đang có ra file theo đúng format ban đầu

**Bài tập 2**. Theo dõi thời gian online của mạng internet

Mỗi tài khoản muốn sử dụng internet sẽ cần đăng nhập và mỗi lần đăng nhập thì sẽ dùng được trong khoảng 30 phút, sau đó muốn sử dụng tiếp sẽ phải đăng nhập lại.

Bạn sẽ có 1 file log chứa các thông tin đăng nhập tài khoản trong ngày hiện tại gồm

|  |
| --- |
| 07:30:21 IM121233434  07:32:21 IM521232431  07:35:01 IM425232430  08:15:21 IM121233434  09:05:01 IM425232430  09:15:21 IM121233434  09:25:21 IM321233334  09:25:21 IM121233430  10:30:21 IM121233434 |

Thời gian và mã định danh phần cứng của mỗi máy. Mỗi đăng nhập này sẽ có thời gian dùng tối đa 30’ (trong thời gian này máy sẽ KHÔNG phải đăng nhập)

Hãy lưu trữ thông tin file log này vào trong máy và tính

* Mã máy này dùng mạng nhiều nhất
* Ước lượng tổng thời gian dùng của máy để có có chế xử lý (VD. Cấm ID của máy)
* Khoảng thời gian nào mà KHÔNG có người dùng (cửa hàng vắng khách)
* Tìm khung giờ (theo giờ chẵn) khoảng thời gian nào cửa hàng đông khách nhất. Ví dụ với file trong minh họa thì khung giờ đông khách nhất là 9:00 – 10:00 (vì có 4 đăng nhập)
* Chú ý: nếu thời gian đăng nhập là 9h40 thì khách có thể nằm trong 2 khung giờ là 9:00-10:00 và 10:00 –11:00