ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

LẬP TRÌNH C CƠ BẢN

Danh sách tuyến tính

1

Nội dung

- Cài đặt các thao tác cơ bản trên danh sách liên kết đơn
- Bài toán quản lý hồ sơ
- Cài đặt bằng danh sách liên kết đơn
- Cài đặt bằng danh sách liên kết đôi

2

Các thao tác cơ bản trên danh sách liên kết đơn

- Mỗi nút của danh sách liên kết đơn có cấu trúc sau

```
typedef struct Node{
    int value;
    struct Node* next;
}Node;
```

- Cài đặt các hàm
  - Node* insertLast(Node* h, int v);// insert a node at that last position
  - Node* removeFirst(Node* h, int v);// remove a first node having value v
  - Node* removeAll(Node* h, int v);// remove all nodes having value v
  - int count(Node* h);// count number of nodes
  - Node* reverse(Node* h);// reverse the linked list

Các thao tác cơ bản trên danh sách liên kết đơn

- Định nghĩa cấu trúc dữ liệu

```
#include <stdio.h>
typedef struct Node{
    int value;
    struct Node* next;// point to the next
      //element of the current element
}Node;

Node*makeNode(int v){// allocate memory for a new node
    Node* p = (Node*)malloc(sizeof(Node));
    p->value = v; p->next = NULL;
    return p;
}
```

Các thao tác cơ bản trên danh sách liên kết đơn

- Chèn thêm 1 nút vào cuối danh sách (sử dụng và không sử dụng đệ quy)

```
Node* insertLast(Node* h, int v){
      Node* p = h;
     if(h == NULL){
            return makeNode(v);
      }
      // general case
      while(p->next != NULL)
            p = p->next;

      Node* q = makeNode(v);
      p->next = q;
      return h;
}
```

```
Node* insertLastRecursive(Node* h, int v){
    if(h == NULL){
        return makeNode(v);
    }
    h->next = insertLastRecursive(h->next, v);
    return h;
}
```


5

Các thao tác cơ bản trên danh sách liên kết đơn

- Loại bỏ 1 nút có giá trị bằng v (không dùng đệ quy)

```
Node* removeNode(Node* h, int v){
    Node* p = h;
    if(h == NULL) return NULL;
    if(h->value == v
        Node* tmp = h; h = h->next;
        free(tmp);  return h;
    }
    while(p->next != NULL){
        if(p->next->value == v) break;
        p = p->next;
    }
    if(p->next != NULL){
        Node* q = p->next; p->next = q->next; free(q);
    }
    return h;
}
```


6

Các thao tác cơ bản trên danh sách liên kết đơn

- Loại bỏ 1 nút có giá trị bằng v (sử dụng đệ quy)

```
Node* removeNodeRecursive(Node* h, int v){
    if(h == NULL) return NULL;
    if(h->value == v){
        Node* tmp = h; h = h->next; free(tmp); return h;
    }
    h->next = removeNodeRecursive(h->next, v);
    return h;
}
```


Các thao tác cơ bản trên danh sách liên kết đơn

- Loại bỏ tất cả các nút có giá trị bằng v (sử dụng đệ quy)

```
Node* removeAll(Node* h, int v){
    // remove all nodes having value v from the linked list headed by h
    if(h == NULL) return NULL;
    if(h->value == v){
        Node* tmp = h; h = h->next; free(tmp);
        h = removeAll(h,v); // continue to remove other elements having value v
        return h;
    }
    h->next = removeAll(h->next,v);
    return h;
}
```

Các thao tác cơ bản trên danh sách liên kết đơn

- Đếm số nút trên danh sách (sử dụng và không sử dụng đệ quy)

```
int countRecursive(Node* h){
    if(h == NULL) return 0;
    return 1+countRecursive(h->next);
}
```

```
int count(Node* h){
    int cnt = 0;
    Node* p = h;
    while(p != NULL){
        cnt += 1;
        p = p->next;
    }
    return cnt;
}
```


Các thao tác cơ bản trên danh sách liên kết đơn

- Đảo ngược thứ tự các nút trong danh sách

```
Node* reverse(Node *h){
    Node* p = h;
    Node* pp = NULL;
    Node* np = NULL;
    while(p != NULL){
        np = p->next;
        p->next = pp;
        pp = p;
        p = np;
    }
    return pp;
}
```

Bài tập quản lý hồ sơ sinh viên

- Hồ sơ sinh viên bao gồm:
  - name: tên của sinh viên
  - email: địa chỉ email của sinh viên
- Viết chương trình trong chế độ tương tác dòng lệnh thực hiện các nghiệp vụ cơ bản trong quản lý hồ sơ sinh viên
  - Đọc dữ liệu từ file văn bản vào danh sách
  - In danh sách sinh viên
  - Thêm 1 hồ sơ vào cuối danh sách
  - Xóa 1 hồ sơ
  - Tìm kiếm hồ sơ
  - Lưu hồ sơ vào file văn bản

Quản lý hồ sơ sinh viên: danh sách liên kết đơn

- Khai báo dữ liệu

```
#include <stdio.h>
#define MAX_L 256

typedef struct Profile{
    char name[MAX_L]; // ten sinh vien
    char email[MAX_L];// email cua sinh vien
    struct Profile* next;
}Profile;

Profile* first, *last;
```

Quản lý hồ sơ sinh viên: danh sách liên kết đôi

- Định nghĩa cấu trúc dữ liệu

```
#include <stdio.h>
#define MAX_L 256

typedef struct Profile{
    char name[MAX_L];
    char email[MAX_L];
    struct Profile* next;// pointer to the next element
    struct Profile* prev;// pointer to the predecessor
}Profile;

Profile* first, *last;
```


13

14