

Môn thi: Cấu trúc dữ liệu và giải thuật
Mã lớp: Các lớp IT003 - Hệ đại trà, chất lượng cao
Thời gian làm bài: 75 phút – Đề 2
(Sinh viên không được sử dụng tài liệu)

Thang điểm gợi ý :

Câu	Nội dung	Điểm
1.a	Trình bày đúng các bước thuật toán tìm kiếm nhị phân	2 điểm
1.b	Mô tả từng bước kết quả chạy thuật toán tìm kiếm nhị phân	2 điểm
2	Ý 1: Định nghĩa đúng cấu trúc dữ liệu cho ngăn xếp	0.5 điểm
	Ý 2: Viết các hàm phụ trợ: hàm thêm vào ngăn xếp, lấy phần tử trong ngăn xếp,...	1 điểm
	Ý 3 : Viết hàm / đoạn lệnh đổi số N từ hệ 10 sang hệ 16 và gọi hàm phụ trợ	0.5 điểm
3.a	Định nghĩa đúng cấu trúc dữ liệu cho danh sách liên kết	1 điểm
3.b	Viết đúng hàm nhập vào danh sách M người mua vé, M nhập từ bàn phím	1 điểm
3.c	Viết đúng hàm hiển thị danh sách người mua vé loại giảm giá	1 điểm
3.d	Viết đúng hàm đếm số lượng người mua vé chọn loại ghế ngồi là ghế VIP	1 điểm

Tổng điểm : 10

Câu 1: (4 điểm)

- a. Hãy trình bày các bước thuật toán tìm kiếm nhị phân.

Đáp án tham khảo:

Trình bày được đầy đủ các bước trong giải thuật tìm kiếm nhị phân : 2 điểm.

- Giả sử mảng hiện hành A gồm N phần tử. các bước của giải thuật tìm kiếm phần tử X trong mảng A, như sau:
 - Bước 1: Gán Left=0; Right=N-1; (dãy từ A₀ đến A_{N-1})
 - Bước 2:
 - Mid=(Left + Right) / 2;
 - So sánh A[Mid] với x. Có 3 khả năng
 - A[mid]= X: tìm thấy X. Dừng giải thuật
 - A[mid]>X : Cập nhật lại Right= Mid-1;
 - A[mid]<X : Cập nhật lại Left= Mid+1;
 - Bước 3: Nếu Left <=Right
 - Lặp lại bước 2
 - Ngược lại : Dừng

- b. Mô tả từng bước kết quả chạy thuật toán để tìm giá trị x=10 trong mảng A={1,

2, 3, 4, 6, 7, 8, 9} bằng thuật toán tìm kiếm nhị phân.

Đáp án tham khảo:

Số phần tử $N = 8$, phần tử cần tìm $X = 10$

1	2	3	4	6	7	8	9
---	---	---	---	---	---	---	---

Lần 1

0.25 điểm

$Left = 0, Right = N - 1 = 7$

$Mid = (Left + Right) / 2 = (0 + 7) / 2 = 3$

0.25 điểm

$A[Mid] = A[3] = 4 < X = 10$: cập nhật $Left = Mid + 1 = 3 + 1 = 4$

Do $Left \leq Right$: thuật toán tiếp tục

Lần 2

0.25 điểm

$Left = 4, Right = N - 1 = 7$

$Mid = (Left + Right) / 2 = (4 + 7) / 2 = 5$

0.25 điểm

$A[Mid] = A[5] = 7 < X = 10$: cập nhật $Left = Mid + 1 = 5 + 1 = 6$

Do $Left \leq Right$: thuật toán tiếp tục

Lần 3

0.25 điểm

$Left = 6, Right = N - 1 = 7$

$Mid = (Left + Right) / 2 = (6 + 7) / 2 = 6$

0.25 điểm

$A[Mid] = A[6] = 8 < X = 10$: cập nhật $Left = Mid + 1 = 6 + 1 = 7$

Do $Left \leq Right$: thuật toán tiếp tục

Lần 4

0.25 điểm

Left = 7, Right = N - 1 = 7

Mid = (Left + Right) / 2 = (7 + 7) / 2 = 7

0.25 điểm

A[Mid] = A[7] = 9 < X = 10 : cập nhật Left = Mid + 1 = 7 + 1 = 8

Do Left > Right : thuật toán kết thúc.

Kết luận : không có X = 10 trong dãy.

Câu 2: (2 điểm)

Hãy viết chương trình chuyển đổi một số nguyên N trong hệ thập phân (hệ 10) sang biểu diễn ở hệ thập lục phân (hệ 16), sử dụng cấu trúc ngăn xếp.

Ví dụ: nhập N = 10 hệ 10 sẽ chuyển thành A hệ 16; nhập N=190 sẽ chuyển thành BE hệ 16.

Đáp án tham khảo:

- SV có thể sử dụng mảng hoặc danh sách liên kết cho ngăn xếp.
- Đáp án tham khảo sử dụng danh sách liên kết.

Định nghĩa CTDL

```
struct NODE {  
    string data;  
    NODE* next;  
};
```

```
struct LIST {  
    NODE* head;  
    NODE* tail;  
};
```

Các hàm phụ trợ

```
void TaoList(LIST& l) {  
    l.head = l.tail = NULL;  
}
```

```
NODE* TaoNode(string x) {  
    NODE* p = new NODE;  
    p->next = NULL;  
    p->data = x;  
    return p;  
}
```

```

void push(LIST& l, NODE* p) {
    if (l.head == NULL) {
        l.head = l.tail = p;
    }
    else {
        p->next = l.head;
        l.head = p;
    }
}

void pop(LIST& l) {
    NODE* temp = new NODE;
    if (l.head == NULL)
        return;
    else {
        NODE* temp = l.head;
        l.head = temp->next;
        cout << temp->data;
    }
    delete temp;
}

void Xuat(LIST l) {
    for (NODE* k = l.head; k != NULL; k = k->next)
        pop(l);

    cout << endl;
}

```

Hàm chuyển đổi số N từ hệ 10 sang hệ 16

```

void ChuyenSangThapLucPhan(LIST l, int n) {
    NODE* p;
    int x;
    cout << "Thap luc phan: ";
    if (n == 0) {
        cout << 0 << endl;
        return;
    }
    while (n != 0) {
        x = n % 16;
        n /= 16;
        if (x == 10) p = TaoNode("A");
        else if (x == 11) p = TaoNode("B");
        else if (x == 12) p = TaoNode("C");
        else if (x == 13) p = TaoNode("D");
        else if (x == 14) p = TaoNode("E");
        else if (x == 15) p = TaoNode("F");
    }
}

```

```

        else p = TaoNode(to_string(x));
        push(l, p);
    }
    Xuat(l);
}

int main() {
    LIST l3;
    TaoList(l3);
    cout << "Nhap so thap phan: ";
    int n;
    cin >> n;
    if (n >= 0) {
        ChuyenSangThapLucPhan(l3, n);
    }
    else cout << "Moi ban nhap lai so >= 0" << endl;
}

```

Câu 3: (4 điểm)

Giả sử người ta sử dụng một danh sách liên kết đơn gồm N phần tử ($1 < N < 100$) để cài đặt ứng dụng quản lý mua vé xem ca nhạc, mỗi phần tử trong danh sách là thông tin của một người mua vé xem ca nhạc, bao gồm các thông tin: Họ tên; Số lượng vé muốn mua; Loại ghế (1: ghế VIP, 2: ghế thường); Ngày, tháng năm xem ca nhạc; Loại vé (1: vé giảm giá, 0: vé không giảm giá).

Hãy thực hiện các yêu cầu sau :

a. Định nghĩa cấu trúc dữ liệu cho danh sách liên kết : 1 điểm

Đáp án tham khảo:

- CTDL thông tin người xem ca nhạc

```

struct NXN
{ char hoten[50];
  int soluongve;
  char ngaythang[50];
  int loaighe;
  int loaive;
};

```

- CTDL lưu thông tin cho 1 node trên DSLK

```

typedef struct item
{ NXN data;
  struct item *next;
}node;

```

- CTDL cho DSLK đơn

```

typedef struct DS
{ node * head;
  node *tail;
}list;

```

b. Viét hàm nháp vào danh sách M người mua vé, M nháp từ bàn phím ($0 < M < N$) :
1 điểm

Đáp án tham khảo:

Viét hàm nháp danh sách đúng : 1 điểm

```
void Nhapdanh sach(list &l)
{
    node *p;
    char hoten[50], ngaythang[50];
    int soluong, loaive, loaighe, m;
    do
    { cout << "Nhap danh sach m nguoi mua ve ";
      cin >> m;
    } while(m < 0);

    for(i=0; i<m; i++)
    { cin.ignore();
      cout << "nhap ho ten nguoi thu" << i + 1;
      cin >> hoten;
      cout << "nhap so luong ve ";
      cin >> soluong;
      cin.ignore();
      cout << "nhap ngay thang ";
      cin >> ngaythang;
      cout << "nhap thong tin loai ghe";
      cin >> loaighe;
      cout << "nhap thong tin giam gia";
      cin >> loaive;
      p = createnode(hoten, soluong, ngaythang, loaighe, loaive);
      AddHead(l, p);
    }
}
```

Viét hàm thêm vào danh sách đúng

```
void AddHead(list &l, node *p)
{ if (l.head == NULL)
    { l.head = p;
      l.tail = p;
    }
    else
    { p->next = l.head;
      l.head = p;
    }
}
```

```

hoặc
void AddTail(list &l, node *p)
{ if (l.head == NULL)
  { l.head = p;
    l.tail = p;
  }
else
  {l.tail->next = p;
   l.tail = p;
  }
}

```

c. Viết hàm hiển thị danh sách người mua vé loại giảm giá. : 1 điểm

Đáp án tham khảo:

```

void hienthi(list l)
{ node *p;
  p= l.head;
  while(p!=NULL)
  { if (p->data.loaive==1)
    cout << p->data.hoten << "\n";
    p=p->next;
  }
}

```

d. Viết hàm đếm số lượng người mua vé chọn loại ghế ngồi là ghế VIP : 1 điểm

Đáp án tham khảo:

```

void demsoluong(list l)
{ node *p;
  int count=0;
  p= l.head;
  while(p!=NULL)
  { if (p->data.loaighe==1)
    count++;
    p=p->next;
  }
  cout << "so nguoi mua ve loai ghe VIP :." << count;
}

```

HẾT

Duyệt đề

Giảng viên ra đề