

BÀI TẬP ÔN THI CUỐI KỲ

1. Danh sách liên kết

Dạng 1	<p>Cài đặt các thao tác trên danh sách liên kết đơn:</p> <ol style="list-style-type: none">1. Tạo dslk đơn L gồm N Node với info thuộc kiểu số nguyên2. Duyệt ds L3. Đếm số lượng các Node có info là số nguyên tố4. Sắp xếp ds L theo thứ tự tăng dần giá trị info5. Thêm 1 Node có info = x vào ds L sao cho ds L vẫn có thứ tự6. Kiểm tra ds L có phải là ds chứa toàn số chẵn hay không? ds L là ds tăng hay giảm hay không tăng không giảm7. Tìm Node có giá trị info là số tăng / số đối xứng8. Tìm Node có giá trị info lớn nhất / nhỏ nhất9. Tìm Node có giá trị info xuất hiện đúng K lần / nhiều nhất10. In giá trị info các Node ở vị trí lẻ trong ds L (Vị trí đầu tiên trong ds là số lẻ 1)								
Dạng 2	<p>Viết chương trình quản lý SÁCH trong thư viện bằng danh sách liên kết đơn, với thông tin mỗi node gồm:</p> <table><thead><tr><th>THÔNG TIN SÁCH</th><th>KIỂU DỮ LIỆU</th></tr></thead><tbody><tr><td>- Mã sách</td><td>- int</td></tr><tr><td>- Loại sách (A, B, C)</td><td>- char</td></tr><tr><td>- Tình trạng (1: còn, 0 : hết)</td><td>- int</td></tr></tbody></table> <ol style="list-style-type: none">1. Nhập danh sách gồm n quyển sách trong thư viện với đầy đủ các thông tin trong bảng mô tả trên.2. Kiểm tra loại sách nào có nhiều nhất trong thư viện.3. Thống kê số lượng sách theo từng loại sách.4. Sắp xếp danh sách theo thứ tự giảm dần mã sách.5. Xóa tất cả các sách có tình trạng = 0.	THÔNG TIN SÁCH	KIỂU DỮ LIỆU	- Mã sách	- int	- Loại sách (A, B, C)	- char	- Tình trạng (1: còn, 0 : hết)	- int
THÔNG TIN SÁCH	KIỂU DỮ LIỆU								
- Mã sách	- int								
- Loại sách (A, B, C)	- char								
- Tình trạng (1: còn, 0 : hết)	- int								
Dạng 3	<pre>#include<iostream> using namespace std; // Câu 1 - 2 điểm</pre>								

```

// Câu 2 - 4 điểm

// Câu 3 - 4 điểm

//-----
int main()
{
    // Câu 1 - 2 điểm
    int A[10]={6,2,7,1,5,9,8,4,0,1};
    int N=10;
    BubbleSort(A,N);

    // Câu 2 - 4 điểm
    List l;
    CreateList(l);
    InputList(l,N);
    DanhsachTang(l);

    // Câu 3 - 4 điểm
    int Arr[11]={44,18,88,13,37,59,99,15,23,55,71};
    int M=11;
    int Tong=0;
    TREE T;
    InputTree(T, A, M);
    TongNodeChan(T,Tong);

    cout<<endl;
    //system("pause");
    return 0;
}

```

Khai báo các cấu trúc dữ liệu và hàm cần thiết để hàm main thực thi được

2. Stack - Queue

1. Cài đặt Stack trên dslk đơn
2. Ứng dụng Stack để giải bài toán đảo mảng
3. Ứng dụng Stack để giải bài toán chuyển đổi hệ cơ số
4. Ứng dụng Stack để giải bài toán Bracket Matching

1. Cài đặt Queue trên dsll đơn
2. Ứng dụng Queue để giải bài toán Palindrome
3. Ứng dụng Queue để giải bài toán Demerging

3. Các thuật toán tìm kiếm và sắp xếp

Cài đặt các thuật toán trên mảng 1 chiều số nguyên A gồm n phần tử

1. Thuật toán tìm kiếm tuyến tính (Cải tiến)
2. Thuật toán tìm kiếm nhị phân
3. Thuật toán tìm kiếm nội suy
4. Thuật toán sắp xếp InterchangeSort
5. Thuật toán sắp xếp BubbleSort
6. Thuật toán sắp xếp SelectionSort
7. Thuật toán sắp xếp InsertionSort
8. Thuật toán sắp xếp HeapSort
9. Thuật toán sắp xếp QuickSort
10. Thuật toán sắp xếp MergeSort

Cài đặt các thuật toán trên danh sách liên kết đơn gồm n node:

1. Thuật toán sắp xếp InterchangeSort
2. Thuật toán sắp xếp SelectionSort
3. Thuật toán sắp xếp QuickSort

4. Cây nhị phân tìm kiếm

4.1. Kiểm tra thao tác trên cây

Cho dãy số sau: 56, 37, 20, 83, 41, 25, 39, 70, 71, 49, 60, 79, 58

1. Hãy vẽ cây nhị phân tìm kiếm
2. Duyệt trước NLR, giữa LNR, sau LRN
3. Liệt kê node lá, node gốc, node 1 con, node 1 con bên trái, 1 con bên phải, node 2 con
4. Cho biết chiều cao cây, các node ở từng mức, đường đi từ node gốc đến node X
5. Xóa lần lượt các node sau đây: 20, 37, 70, 83, 56

4.2. Kiểm tra code

Viết chương trình thực hiện các thao tác trên cây nhị phân tìm kiếm NPTK với Key thuộc kiểu số nguyên

1. Khai báo cấu trúc cây NPTK
2. Tạo 1 cây rỗng

3. Tạo 1 nút có trường Key bằng x
4. Thêm 1 nút vào cây nhị phân tìm kiếm
5. Thêm n nút vào cây nhị phân tìm kiếm
6. Duyệt cây LNR, NLR, LRN
7. Tìm 1 nút có khoá bằng x trên cây
8. Xóa 1 nút có Key bằng x trên cây
9. Tính chiều cao cây
10. Viết hàm xuất / đếm / tính tổng giá trị các node có giá trị là số chẵn / số nguyên tố / có giá trị lớn hơn x và nhỏ hơn y ...
11. Viết hàm xuất / đếm / tính tổng giá trị các node lá / node 1 con (trái hoặc phải) / node 1 con bên trái / node 1 con bên phải / node 2 con
12. Viết hàm xuất / đếm / tính tổng giá trị các node ở tầng thứ K trên cây
13. Viết hàm xuất / đếm / tính tổng giá trị các node ở tất cả trên cây theo thứ tự từ tầng 0 đến tầng h-1 của cây (với h là chiều cao của cây)
14. Viết hàm xuất / đếm / tính tổng giá trị các node ở tầng cao hơn tầng thứ K trên cây
15. Viết hàm xuất / đếm / tính tổng giá trị các node ở tầng thấp hơn tầng thứ K trên cây

Đối với bài tập cây NPTK yêu cầu **liệt kê (xuất) / đếm số lượng / tính tổng giá trị** các node theo yêu cầu ta có code tổng quát như sau:

```
void DieuKienNode(Tree T, int &Dem, int &S)
```

```
{
    if (T)
    {
        if (T->Key % 2 != 0) // Điều kiện node --> Ví dụ như node 1 con / node 2 con /
                           // node lá / node có giá trị chẵn - lẻ - số nguyên tố...
        {
            cout << T->Key << " "; // xuất hoặc liệt kê các node theo điều kiện
            Dem++;                  // "dem" thể hiện số lượng các node theo điều kiện
            S += T->Key;             // "S" thể hiện tổng giá trị các node theo điều kiện
        }
        DieuKienNode(T->pLeft, Dem, S);
        DieuKienNode(T->pRight, Dem, S);
    }
}
```

Đối với bài tập cây NPTK yêu cầu **liệt kê (xuất) / đếm số lượng / tính tổng giá trị** các node trên tầng K ta có code tổng quát như sau:

```
void TangK(TREE T, int TangK, int &Dem, int &S)
```

```
{
```

```

if (T)
{
    TangK--;
    if (TangK==0)
    {
        cout << T->key << " "; // xuất hoặc liệt kê các node trên tầng K
        Dem++;                // "dem" thể hiện số lượng các node trên tầng K
        S += T->key            ;// "S" thể hiện tổng giá trị các node trên tầng K
    }
    TangK(T->pLeft, TangK, Dem, S);
    TangK(T->pRight, TangK, Dem, S);
}
}

```

5. B-Tree

Vẽ cây B-Tree bậc 5 từ dãy các khóa sau: 20, 40, 10, 30, 15, 35, 7, 26, 18, 22, 5, 42, 13, 46, 27, 8, 32, 38, 24, 45, 25

Xóa node 8, 22, 32, 38, 35 (35 hoặc 40)

6. Bảng băm

Thêm các nút 32, 53, 22, 92, 17, 34, 24, 37, 56 vào bảng băm. Hàm băm $H = \text{Key} \% 11$

Giải quyết xung đột bằng các phương pháp băm sau:

Phương pháp nối kết

- Nối kết trực tiếp - Direct Chaining
- Nối kết hợp nhất - Coalesced Chaining

Phương pháp băm lại

- Dò tuyến tính - Linear probing
- Dò bậc 2 - Quadratic probing
- Băm kép - Double hashing

	Phương pháp	Hàm băm $f(\text{key})$	Giải quyết xung đột
Nối kết	Nối kết trực tiếp - Direct Chaining	$H(\text{key}) = \text{key} \% M$	Gom thành 1 dslk
	Nối kết hợp nhất - Coalesced Chaining	$H(\text{key}) = \text{key} \% M$	Nút trống phía cuối mảng

Băm lại	Dò tuyến tính - Linear probing	$H(\text{key}) = \text{key} \% M$	$H'(\text{key}) = (H(\text{key}) + i) \% M$
	Dò bậc 2 - Quadratic probing	$H(\text{key}) = \text{key} \% M$	$H'(\text{key}) = (H(\text{key}) + i^2) \% M$
	Băm kép - Double hashing	$H_1(\text{key}) = \text{key} \% M$ $H_2(\text{key}) = \text{key} \% M$	$H'(\text{key}) = (H_1(\text{key}) + i * H_2(\text{key})) \% M$
$i = 0, 1, 2, \dots, M-1$			

7. Đồ thị

