**Bài tập đệ quy:**

Bài 1: Tính tổng các số bằng đệ quy

-Thuât toán thứ nhất:

* Để tính S(n) ta tính S(n-1) cộng với n;

Dùng điều kiện neo:

If(n==1) return 1;

Else

Return sum(n-1)+n;

* Điều kiện nhập là n >=1

Ta chon n=3

S(1)=0+1=1

S(2)=1+2=3

S(3)=3+3=6

**Bài 2: Tìm số lớn nhất trong mảng bằng đệ quy**

-Cho mảng cho gồm 3 phần tử:

+Điều kiện nếu ( n==1)

Return a[0];

+Ngược lại:

-Nếu(**max**(a,n-1) > a[n-1]) return **max(**a,n-1);

(Tìm max trong số (n-1) phần tử đầu, so sánh với phần tử cuối cùng để tìm ra max trong mảng)

Else return a[n-1];

**Baì 4: Tìm kiếm tuyến tính bằng đệ quy:**

-Nhập 1 mảng gồm n phần tử

-Nếu n < 1 tìm kiếm trả về 0 và dừng

If(n<1)

Return 0;

-Ngươc lại:

+Gọi đệ quy RlinearSearch(a, n-1, x);

Tìm kiếm từ cuối mảng trở về đầu mảng

Nếu phần tử cuối bằng với x 🡪 in ra tìm thấy và dừng

If (a[n-1]==x)

{ printf(“Tìm thấy”); return 1;}

Xuất kết quả:

If (n==0)

Printf(“Không tìm thấy”);

**Bài 5: Mô tả thuật toán đệ quy in ra chuỗi đảo ngược:**

Nhập 1 mảng gồm n phần tử

Nếu mảng có một phần tử thì in ra

If(n==1) printf(“%d”, a[0]);

Ngươc lại:

In ra phần tử cuối cúng trước

Printf(“%d”, a[n-1]);

Sau đó gọi đệ quy xuất ra các phần tử còn lại

xuatNguoc(a,n-1);

**Bài 7:Kiểm tra dãy đối xứng:**

Nhập 1 mảng gồm 5 phần tử

Cho vòng lặp

For (i=0;i<n/2;i++)

Nếu phần tử không bằng với phần tử cuối thì trả về 0 và dừng

If(a[i]!=a[n-1-i]);

Return 0;

Break;

Ngược lại: return 1;

Xuất kết quả:

If(doiXung(a,5)==1)

Printf(“Mảng đối xứng”);

Else

Printf(“Mảng không đối xứng”);

Âsds