

Lab 04: Đệ quy

Thời gian dự kiến: 01 tuần

1 Đệ quy cơ bản

Sử dụng prototype cho trước, viết các hàm thực hiện các yêu cầu sau (sử dụng kỹ thuật đệ quy):

1. Tính tổng $S = 1 + 2 + 3 + \dots + n$

```
long long sumSequence(long long n);
```

2. Tính giá trị $n! = 1 \times 2 \times 3 \times \dots \times n$

```
long long factorial(long long n);
```

3. Tính x^n

```
long long power(long long x, long long n);
```

4. Đếm số lượng chữ số của một số nguyên n

```
int countDigits(int n);
```

5. Đếm số lượng chữ số **lẻ** của một số nguyên n

```
int countOddDigits(int n);
```

6. Kiểm tra tất cả chữ số của số nguyên n có phải là số chẵn hay không?

```
bool isEvenDigits(int n);
```

7. Tìm ước chung lớn nhất (GCD), bội chung nhỏ nhất (LCM) của hai số nguyên a và b

```
int GCD(int a, int b);  
int LCM(int a, int b);
```

8. Tìm số đảo ngược của một số nguyên n (VD: $1234 \rightarrow 4321$)

```
int reverseNumber(int n);
```

9. Chuyển một số thập phân về dạng nhị phân, in kết quả ra màn hình.

```
void binaryValue(int n);
```

10. Người ta định nghĩa công thức của số Fibonacci thứ n bằng hệ thức đệ quy

$$\begin{cases} F_0 = 0, F_1 = 1 \\ F_n = F_{n-1} + F_{n-2} \text{ nếu } n \geq 2 \end{cases}$$

Tìm số Fibonacci thứ n ?

```
long long fibonacci(long long n);
```

11. Kiểm tra một chuỗi s có phải là một chuỗi đối xứng (palindrome) hay không?

```
bool isPalindrome(char* s, int left, int right);
```

12. (*) Cho một chuỗi s gồm n kí tự **phân biệt**, in ra màn hình tất cả các hoán vị có thể có của s

- VD: $s = ABCD, n = 4$ sẽ in ra $ABDC, ACBD, ACDB, ..$

```
void stringPermutation(char* s, int i, int n);
```

13. Tìm và trả về vị trí đầu tiên mà **key** xuất hiện trong mảng a **đã được sắp xếp tăng dần**, nếu **key** không xuất hiện trong a thì trả về -1 .

```
int binarySearch(int* a, int left, int right, int key);
```

2 Danh sách liên kết

Cho cấu trúc của một Node trong danh sách liên kết:

```
1 struct Node {  
2     int key;  
3     Node* pNext;  
4 };
```

Sử dụng prototype cho trước, viết các hàm thực hiện các yêu cầu sau (sử dụng kỹ thuật đệ quy):

1. Chèn một Node vào cuối danh sách liên kết

```
void insertBack(Node*& listRoot, Node* node);
```

2. Xóa node đầu tiên có giá trị **key** trong DSLK

```
bool deleteValue(Node*& listRoot, int key);
```

3. Tìm và trả về index của node đầu tiên có giá trị **key**, nếu không có node nào mang giá trị **key** thì trả về -1 .

```
int findKey(Node* listRoot, int key);
```

4. Giải phóng một DSLK

```
void releaseList(Node*& listRoot);
```