# BÀI TẬP 2 – LẬP TRÌNH ĐỆ QUY

# Bài 1:

# Cho dãy A gồm n phần tử số nguyên: A[1], A[2], …, A[n]. Hãy viết các thuật toán sau (hàm/ chương trình con) bằng phương pháp ĐỆ QUY, từ đó xác định độ phức tạp tính toán của các thuật toán mà bạn đã viết:

# 1. Tìm giá trị lớn nhất của dãy A.

2. Đảo ngược dãy A này. Ví dụ: Nếu dãy A=(5, 2, 4), thì sau khi đảo ngược ta có:

A=(4, 2, 5).

3. Kiểm tra dãy A có đối xứng hay không. Ví dụ: Dãy A=(4,2,4) hoặc A=(4,1,1,4) đều là các dãy đối xứng.

4. Đếm số các cặp nghịch đảo trong A. Biết rằng, một cặp nghịch đảo trong A là cặp (A[i], A[j]) sao cho i<j và A[i]>A[j]. Ví dụ: Dãy (5, 2, 4) có 2 cặp nghịch đảo là:

(5, 2) và (5, 4).

# Bài 2:

Xác định độ phức tạp thuật toán của hai thuật toán đệ quy để tính xn như sau.

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuật toán 1** | **Thuật toán 2** |  |  |
| float LT(float x, unsigned int n)  {  if (n==0) return 1;  else  if (n%2==0)  return LT(x, n/2)\*LT(x, n/2);  else  return x\*LT(x, n/2)\*LT(x, n/2);  } | float LT(float x, unsigned int n)  {  if (n==0) return 1;  else  if (n%2==0)  return sqr(LT(x,n/2));  else  return x\*sqr(LT(x,n/2));  } |  |  |

# Bài 3: Tính kết quả nếu với lời gọi hàm sau: f1(4, 2) và f2(3, 2).

**Hàm f1 Hàm f2**

|  |  |
| --- | --- |
| int f1(int x, int y)  {  if (x == 0)  return y;  else  return f1(x - 1, x + y);  } | int f2(int x, int y)  {  if (y == 0)  return 0;  else  if (y > 0 )  return (x + f2(x, y-1));  else  return –f2(x, -y);  } |

**Bài 4:** Xét định nghĩa đệ quy về ước số chung lớn nhất của hai số nguyên m và n như sau:

Dựa vào định nghĩa này hãy viết hàm trả về giá trị ước số chung lớn nhất của hai số nguyên m và n bằng 2 phương pháp: đệ quy và lặp.