

# Bài tập thực hành

## Cấu trúc dữ liệu và thuật toán

### Quicksort (A)

#### 1. Bài 1

Viết chương trình bằng ngôn ngữ lập trình C để hoàn thành bài tập sắp xếp. Cho mảng K: K[1], K[2], ..., K[n] chứa dãy n số nguyên dương. K[0]= 0.

- Cho khai báo của hàm sắp xếp dãy số nguyên theo thứ tự tăng dần dùng giải thuật sắp xếp -Quicksort. LB là chỉ số của phần tử đầu, UB là chỉ số phần tử cuối của dãy cần sắp xếp.

```
void QuickSort(int K[], int LB, int UB);
```

- Cho khai báo hàm phân đoạn dãy K có chỉ số phần tử đầu là LB, chỉ số phần tử cuối là UB, chọn phần tử chốt là phần tử đầu. Hàm trả về vị trí của phần tử chốt sau khi thực hiện phân đoạn.

```
int Part(int K[], int LB, int UB);
```

a) Hoàn thiện mã chương trình của hàm Part()

```
int Part(int K[], int LB, int UB) {  
    int i, m;  
    i = ____;  
    int j = ____;  
  
    while (i < j) {  
        while (K[i] ____ K[LB]) i = ____;  
        while (K[j] ____ K[LB]) j = ____;  
        if (i < j) {  
            m = K[i];  
            K[i] = K[j];  
            K[j] = m;  
            i = ____;  
            j = ____;  
        }  
    }  
}
```

```

    if (K[LB] > K[j]) {
        m = ____;
        K[LB] = ____;
        K[j] = ____;
    }
    return j;
}

```

b) Hoàn thiện mã chương trình của hàm QuickSort()

```

void QuickSort(int K[], int LB, int UB) {
    int j;
    if (LB < UB) {
        j = Part(____);
        QuickSort(____);
        QuickSort(____);
    }
}

```

c) Hoàn thiện mã chương trình của hàm main() sắp xếp dãy từ 1 đến 10.

```

void print_array(int k[], int n) {
    int i;
    for (i = 0; i < n; i++) {
        printf("%d ", k[i]);
    }
}

```

```

int main(void) {
    int K[12]={0,42,23,74,11,65,58,94,36,99,87,30000};
    printf("\nMang truoc khi sap xep: ");
    print_array(K, 11);
    QuickSort(K, ____, ____);
}

```

```
printf("\nMang sau khi sap xep: ");  
print_array(K, 11);  
return 0;  
}
```

## 2. Bài 2

Viết hàm Part(), hàm QuickSort() và hàm main() trong Bài 1 trường hợp sắp xếp giảm dần.