

Bài tập thực hành

Cấu trúc dữ liệu và thuật toán

MergeSort (A)

1. Bài 1

Viết chương trình bằng ngôn ngữ lập trình C để hoàn thành bài tập sắp xếp **tăng dần** dùng giải thuật sắp xếp Mergesort.

- Cho mảng K: K[0], K[2], ..., K[n] chứa dãy n+1 số nguyên dương.

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

```
void MergeSort(int b[], int h, int k);
```

- Cho khai báo hàm hòa nhập 2 dãy **tăng dần**. Dãy thứ nhất b có chỉ số phần tử đầu là h, chỉ số phần tử cuối là t. Dãy thứ hai b có chỉ số phần tử đầu là t+1 và chỉ số phần tử cuối là k. Kết quả dãy sau khi hòa nhập là dãy b **tăng dần** có chỉ số từ h tới k.

```
void Merge(int b[], int h, int t, int k);
```

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

```
void Merge(int b[], int h, int t, int k) {  
    int c[t-h+1];  
    int i = 0, u = h, v = t+1;  
  
    for (i = 0; i <= ____ ; i++)  
        c[i] = b[ ____ + i];  
  
    i = 0;  
    while (i <= ____ ) {  
        if (v <= ____ && b[v] < c[i]) {  
            b[u] = ____ ;  
            u++;  
            v = ____;  
        } else {  
            b[u] = ____ ;
```

```

        u++;
        i = ____ ;
    }
}
}

```

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

```

void MergeSort(int b[], int h, int k) {
    if (k - h < 1) return;
    int t = (h+k) / 2;
    MergeSort(b, ____, ____ );
    MergeSort(b, ____, ____ );
    Merge(b, ____, ____, ____ );
}

```

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

```

void print_array(int K[], int i1, int i2) {
    int i;
    for (i = i1; i <= i2; i++) {
        printf("%d ", K[i]);
    }
}

int main(void) {
    int k = 6;
    printf("\nMang truoc khi sap xep: ");
    print_array(K, 0, k);
    MergeSort(K, ____, ____ );
    printf("\nMang sau khi sap xep: ");
    print_array(K, 0, k);
    return 0;
}

```

2. Bài 2

Hoàn thành bài 1 với sắp xếp giảm dần.