

# Ex41 (2023/05/23)

次のプログラムは、クイックソートのコード例である。以下の関数を完成させて、実行できることを確認しなさい。また、クイックソートの安定性を調べなさい。なお、このプログラムでは以下の2つの関数を利用している。

```
void quick_sort(double a[], int left, int right);
```

並べ替えをする再帰関数

```
int partition(double a[], int left, int right);
```

pivotの選択と、pivotの両側で入れ替えが必要であれば入れ替え

# Ex41 (2023/05/23)

```
int main(void) {  
  
    float data[10] ;  
    int n=10;  
  
    set_data(data, n, 0);  
    print_data(data, n);  
  
    quick_sort(data, 0, n-1);  
  
    print_data(data,n);  
  
    return 0;  
}
```

```
void quick_sort(float a[], int left, int right){  
    int mid;  
    if (left < right){  
          
        printf("left: %d-%d¥n", left, mid);  
          
        printf("right: %d-%d¥n", mid+1, right);  
          
    }  
}  
  
int partition(float a[], int left, int right){  
    float pivot, swap;  
    int ij;  
  
    pivot = a[ left ];  
    i = left - 1;  
    j = right + 1;  
    while(1){  
          
    }  
}
```

# Ex42 (2023/05/23)

マージソートプログラムを完成させよ。

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

float my_random(float lower, float upper);
void set_data(float a[], int n, int seed);
void print_data(float a[], int n);
void merge_sort(float a[], int left, int right);

int main(void) {
    float data[10];
    int n=10;

    set_data(data, n, 0);
    print_data(data, n);
    merge_sort(data, 0, n-1);
    print_data(data,n);

    return 0;
}
```

```
void merge_sort(float a[], int left, int right){
    int mid, i, j, k;
    float temp[10];

    if(left>=right)
        return;

    mid = (left+right)/2;
    
    for (i=left; i<=mid; i++)
        temp[i]=a[i];
    for (i=mid+1, j=right; i<= right; i++,j--)
        temp[i]=a[j];
    
}
```