#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

// Function to swap two elements

void swap(int\* a, int\* b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

// First implementation of quicksort

int partition1(int arr[], int low, int high) {

int pivot = arr[high];

int i = (low - 1);

for (int j = low; j <= high - 1; j++) {

if (arr[j] < pivot) {

i++;

swap(&arr[i], &arr[j]);

}

}

swap(&arr[i + 1], &arr[high]);

return (i + 1);

}

void quickSort1(int arr[], int low, int high) {

if (low < high) {

int pi = partition1(arr, low, high);

quickSort1(arr, low, pi - 1);

quickSort1(arr, pi + 1, high);

}

}

// Second implementation of quicksort (slightly different partition strategy)

int partition2(int arr[], int low, int high) {

int pivot = arr[low];

int i = low + 1;

int j = high;

while (1) {

while (i <= j && arr[i] <= pivot) i++;

while (i <= j && arr[j] > pivot) j--;

if (i >= j) break;

swap(&arr[i], &arr[j]);

}

swap(&arr[low], &arr[j]);

return j;

}

void quickSort2(int arr[], int low, int high) {

if (low < high) {

int pi = partition2(arr, low, high);

quickSort2(arr, low, pi - 1);

quickSort2(arr, pi + 1, high);

}

}

int main() {

int n, choice, output\_choice;

clock\_t start1, end1, start2, end2;

double time1, time2;

printf("请输入待输入的元素个数:");

scanf("%d", &n);

int\* arr1 = (int\*)malloc(n \* sizeof(int));

int\* arr2 = (int\*)malloc(n \* sizeof(int));

printf("请选择输入方式(0为随机生成,1为手动输入):");

scanf("%d", &choice);

if (choice == 0) {

srand(time(NULL));

for (int i = 0; i < n; i++) {

arr1[i] = rand() % 100000;

arr2[i] = arr1[i];

}

}

else {

printf("请输入%d个元素:", n);

for (int i = 0; i < n; i++) {

scanf("%d", &arr1[i]);

arr2[i] = arr1[i];

}

}

// Measure time for first implementation

start1 = clock();

quickSort1(arr1, 0, n - 1);

end1 = clock();

time1 = ((double)(end1 - start1)) / CLOCKS\_PER\_SEC;

// Measure time for second implementation

start2 = clock();

quickSort2(arr2, 0, n - 1);

end2 = clock();

time2 = ((double)(end2 - start2)) / CLOCKS\_PER\_SEC;

printf("第一段程序所花时间:%.0f\n", time1);

printf("第二段程序所花时间:%.0f\n", time2);

printf("两段代码的运行时间差是:%g\n", end2 - end1);

printf("是否输出快排后的数据(是为1,否为0):");

scanf("%d", &output\_choice);

if (output\_choice == 1) {

printf("快排后的数据序列:");

for (int i = 0; i < n; i++) {

printf("%d ", arr1[i]);

}

printf("\n");

}

free(arr1);

free(arr2);

return 0;

}