

结对编程

总体要求：

一、 组内两位同学都要积极思考、相互讨论、共同完成各个题目。**不允许一人一题分开独立做，更不允许一个人做题，一个人在忙与作业无关的其他事情。**

二、 同组同学应就题目要求，讨论清楚代码设计与实现之后，再由一人输入代码，另一人则应看盯着屏幕看，并及时提醒出现的错误（包括与讨论的方案不一致的地方），不得做其他事情（如思考另一题）。中途可以更换角色，换另一个人输入代码。

三、 对题目内容有任何疑问或困惑，可以与其他组同学讨论交流思路与要点，但不得直接观看其他组已完成的实际代码。应善于在讨论过程中用纸与笔来辅助表达设计思路。

四、 讨论时的音量大小，至少要让同组同学能轻松听清楚，不需要刻意压制自己的正常音量。

1. 先请仔细观看课上播放的关于机器人进行快速排序的动画。注意：动画中的快速排序与教材中的快速排序是有区别的。

任务：

严格按照动画中快速排序的思路和操作过程，设计并实现一种递归快速排序函数，完成对一维整数数组的从小到大排序。不得转换成教材上的算法实现！

下面是程序大致的框架，其中主函数中提供了部分测试数据，供大家参考：

```
#include <iostream>          /// cout
#include <cstdlib>             /// rand, srand
#include <ctime>               /// time
using namespace std;

void qsort(int* array, int left, int right)
{
    /// 请按照动画中的算法，实现本函数。必要时可以调用自定义的辅助函数。
}

void print(int* array, int num)
{
    for (int i=0; i<num; i++)
        cout << array[i] << ' ';
    cout << endl;
}

int main()
{
    int a[1] = {2};
    qsort(a, 0, 0);
    print(a, 1);

    int b[2] = {4, 1};
    qsort(b, 0, 1);
    print(b, 2);
}
```

```

int c[5] = {1, 2, 3, 4, 5};
qsort(c, 0, 4);
print(c, 5);

int d[5] = {5, 4, 3, 2, 1};
qsort(d, 0, 4);
print(d, 5);

int e[6] = {8, 1, 3, 2, 0, 4};    /// 4 added
qsort(e, 0, 5);
print(e, 6);

int f[6] = {2, 3, 3, 1, 4, 7};    /// 7 added
qsort(f, 0, 5);
print(f, 6);

return 0;
}

```

下面是一个产生 10 个随机小数(在区间[0,1]中均匀分布)的示例程序

```

#include <iostream>    /// cout
#include <cstdlib>      /// rand, srand
#include <ctime>        /// time
using namespace std;
int main()
{
    srand((unsigned int) time(NULL));    /// 设置随机序列的种子

    for(int k=0; k<10; k++)              /// 输出 10 个随机小数
        cout << (float)rand() / RAND_MAX << endl;

    return 0;
}

```