

Lab4
黄子骞 21307130013

实验思路：

首先，定义一个保存学生信息的结构 `stu`，并将所有的学生数据导入到一个以 `stu` 为元素的 `vector` 中（复制黏贴近终端来进行输入，以 `ctrl+z` 结尾）。

之后对数据进行处理，计算出每个学生姓名对应的 `key` 值（此处采用将姓名拼音的每一个字母的 `Ascii` 码相加，最后 `mod 97`）。

之后对每一个学生信息使用 `hash_Insert` 来将信息加入到哈希表中，此处采用 `linear probing`。（遇到已占用位则将索引+1 继续探索）

实验过程：

实现 `read_data` 函数：读取终端中学生信息。

实现 `generate_key` 函数：计算姓名对应 `key` 值。

实现 `add_key` 函数：将 `key` 放入 `vector` 中

实现 `hash_Insert`： `linear probing`，找到空位后将该位置的 `hashSlot` 结构中 `inf` 变量赋值为相应的学生信息（同时保存 `times2find` 变量）

实现 `init_hashmap`：利用 `hash_Insert`，将学生信息插入哈希表进行初始化

实现 `hash_delete`：找到目标元素并将其 `flag` 设置为 `false`，`times2find` 设置为 0

实现 `hash_search`：根据插入时的规则寻找并时刻比较被比较元素与当前 `hashSlot` 的 `key` 值是否相同，相同则找到（由于本实验数据量少，在查找时增加了比较 `name` 是否相同的操作，防止相同 `key` 值情况出现）

实现 `evaluate`：求各 `hashSlot` 的 `times2find` 变量之和，求平均

实验结果：

`Evaluate` 函数结果为 1.34783 次。