专题二：基础数据结构-链表

实验报告

报告小组：47 | 小组成员：郭书廷(3170104871),叶帆(3170102410) | 指导老师：张引，田沈晶，熊海辉

项目：共享子串、多项式计算、通讯录管理

Project1-共享子串

问题描述	存储字符串，将相同的后缀用同一个条链表存储。

算法分析与代码设计	**程序目的：**
	1.如原题所述。
	2.题目原意为输入两个字符串，找出他们的公共后缀，并用链表存储公共后缀。但仅存储两个字符串缺乏实际意义，我们注意到这个题目与一种数据结构——trie 树非常相似，所以本程序的目的是用 trie 树实现一个字典。
	结构设计：
	1.使用链表，将题述结构分为三个部分：一个公共子串，两个非公共子串，将三者链接即可。
	2.我们设计了一个前缀 trie 树的模板，树的节点设计如下：
	struct node{ struct node* s[95]; int leaf; char* mean; };
	其中 s 为子节点数组(assci 字符中共有 95 个可打印字符)，leaf 表示该节点是不是一个单词的结束，mean 为以该节点结束的单词的释义。
	模板设计：
	struct node{ struct node* s[95]; int leaf; char* mean; };
	typedef struct node Prefix_Trie_base;
	typedef struct node* Prefix_Trie_iterator;
	struct Prefix_Trie_ { struct node* root; };
	
	用此类型定义一颗树：
	typedef struct Prefix_Trie_* Prefix_Trie;
	
	新建一棵树：
	Prefix_Trie New_Prefix_Trie();
	
	向树 t 添加含义为 mn 的单词 str:
	void Insert(Prefix_Trie t, const char* str, const char* mn);
	
	在树 t 中查找单词 str，返回释义(char*):
	char* Find(Prefix_Trie t, const char* str);
	
	将树 t 的节点释放至内存池：
	void Free_Prefix_Trie(Prefix_Trie t);

	释放内存池： `void Free_all();`
使用方法	1.执行 strstr.exe，输入两个不含空格的字符串即可。 2.将可执行文件与字典文件 Dict.txt 放在同一目录，输入单词即可。
Project2-多项式计算	
问题描述	计算多项式的和、差、积。
算法分析与代码设计	**功能：** 程序采用命令式交互，执行代码，可以使用一下命令进行多项式计算(系数和求值支持高精度)。 定义变量 var,变量名符合 C 语言标准 set <var> or s <var> 将变量 var1 的值拷贝到 var2 copy <var1> <var2> or cp <var1> <var2> 手动修改变量 var，输入一个多项式 temp，使 var=var+temp add_to <var> or m <var> 输出多项式 var print <var> or p <var> 输出多项式 var 的项数 size <var> 清除所有变量，重置计算器 clear or ca 将两个变量相加/减/乘，并将答案放入预置变量 ans add/minus/mul <var1> <var2> 计算 var 的值（输入 x 的值，答案可以保存到文件） calc <var> or cc <var> 求导 diff <var> 清屏 cls 帮助 help or h 退出 exit or q **算法：** 1.使用第一题的 trie 树模板存储、查找变量名。 2.多项式存储使用链表。 3.使用高精度算法，高精度数的存储使用不定长数组 vector(非 C++模板)。 4.高精度乘法使用快速傅里叶变换。（多项式的乘法仍使用 n^2 方法） 5.求值乘方运算使用倍增法。 6.多项式按指数大小排序使用随机化快排。 **方法简述：** 对于两个多项式如果进行加法，则执行三个步骤： 1.将两个多项式拼接 2.对拼接后的多项式进行排序 3.合并同类项 如果进行减法，则将多项式各项的符号全部翻转，然后进行加法。 如果进行乘法，则将两个多项式的项两两相乘，将乘积放入新的链表，在对新链表排序，合并同类项。 **注：** 由于代码总长度较大(约 1300+行)，关于 vector、倍增法求乘方（快速幂）、链表与简易内存池的实现细节详见 ppt、代码及注释。

<table>
<tr><td>使用说明</td><td>运行可执行文件，按程序提示及上述指令，输入计算即可。</td></tr>
<tr><td colspan="2">Project3-通讯录管理</td></tr>
<tr><td>问题描述</td><td>实现通讯录管理器，满足添加、删除、查找、导入、导出等功能。</td></tr>
<tr><td>算法分析与
代码设计</td><td>
代码设计：
使用第一题的 trie 树模板，将单词释义换成字符串的 vector 即可。
Trie 树节点和 vector 结构体定义如下：

<pre>
struct node{ struct node* s[256]; int leaf; vector vec; };

typedef char* __tp_vector;
const __tp_vector __INIT_DATA_VECTOR=0;
typedef __tp_vector vec_base;
typedef __tp_vector* vector_iterator;
struct vector_
{
  /*vec->end is a virtual element, you may not access it.*/
  vector_iterator data;/*实际存储空间*/
  vector_iterator begin;/*首指针*/
  vector_iterator end;/*尾指针*/
  int size,mx_size;/*当前元素个数和最大元素个数*/
};
</pre>
UI 设计：
i <name> <info> 插入/新建信息
d <name> <num> 删除 name 的第 num 条信息，如果 num 为 0 则删除整个联系人
f <name> 查找联系人
e <file_name> 导出至 file_name
p 输出所有联系人
q 退出
</td></tr>
</table>

使用说明	运行可执行文件，输入要导入的联系人文件名，按上述命令操作即可，输入 h 获取帮助。 **注：**导入后的所有操作都在内存中进行，除非导出，否则程序不会自动保存。