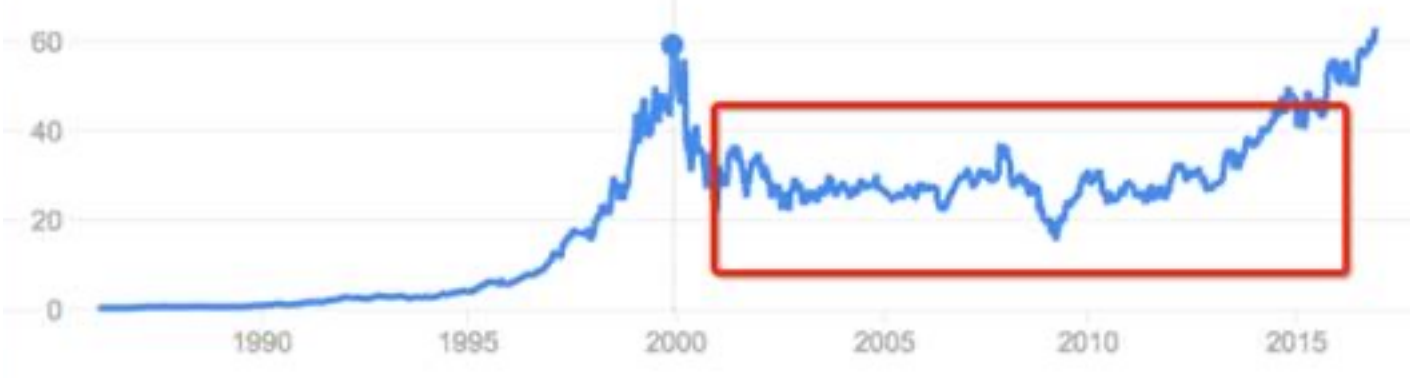
北京郵電大學
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

高级计算机编程实战

熊永平@计算机学院

编程的价值.....



2,500,000,000,000$

三大类编程

- **算法模型（数学复杂）**
  - 数据挖掘、模式识别
  - 图像视频识别
  - NLP
- **系统（工程复杂）**
  - 高并发、hadoop
  - 网络协议栈
  - 文件系统
- **应用（业务复杂）**
  - 工作流
  - APP
  - 业务流程和知识

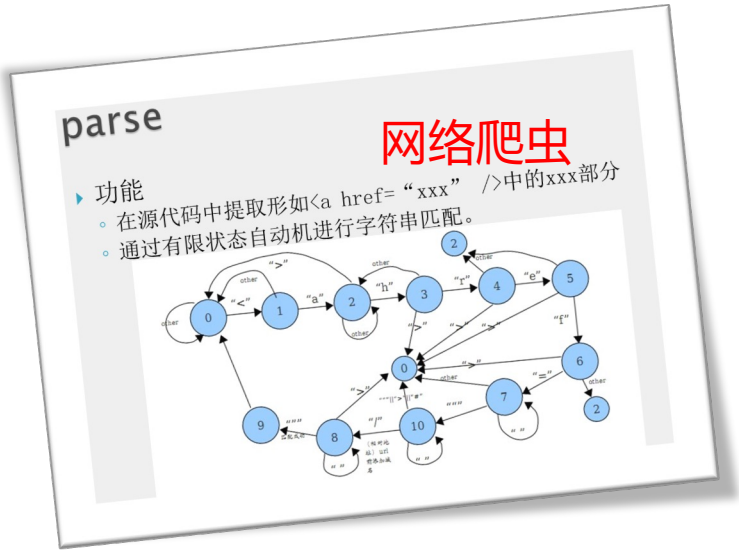
课程理念

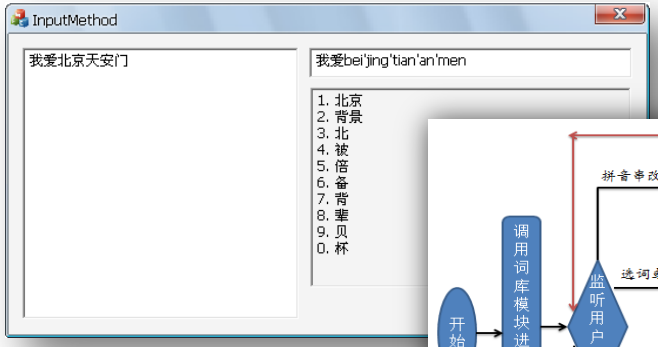
- **基础!**
  - 计算机系统结构
  - 计算机网络原理
  - 大规模数据处理
  - 算法与数据结构
- **编程原则**
  - 忘记框架

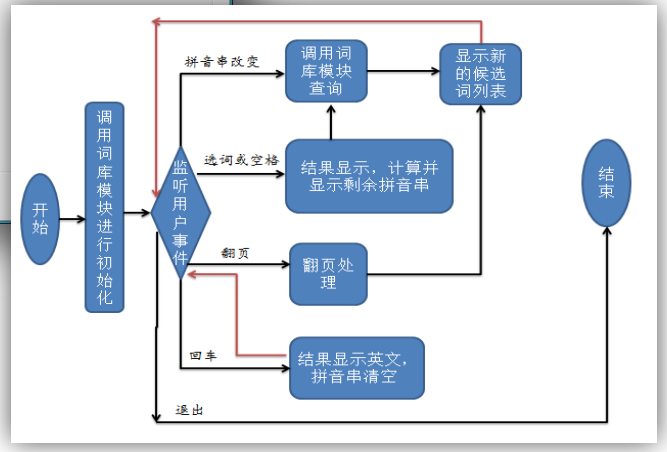
不涉及

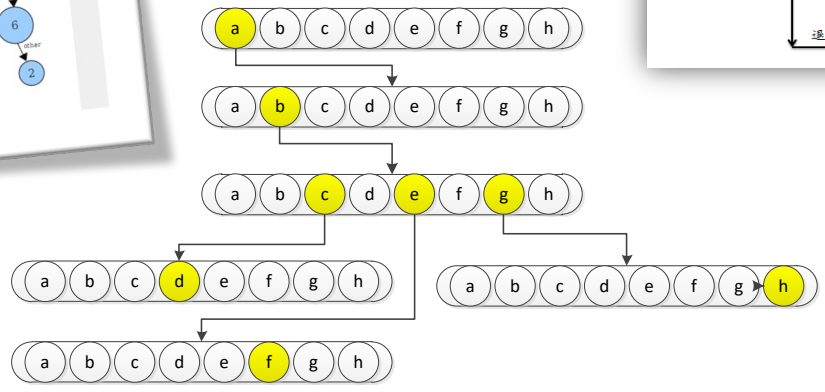
- Web开发(PHP/ASP/AJAX/ROR/JSP….)
- 企业级应用系统开发（Java EE/.Net/Oracle/DB2..)
- ARM/51/MSP430/嵌入式
- Android/IOS/WP的移动开发
- Pytorch/Tensorflow调参工程师

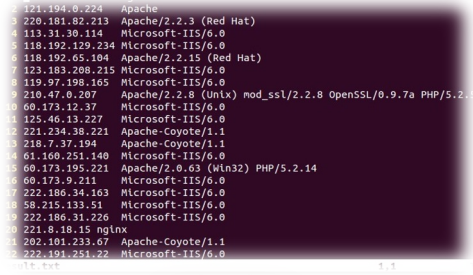
历史实验内容

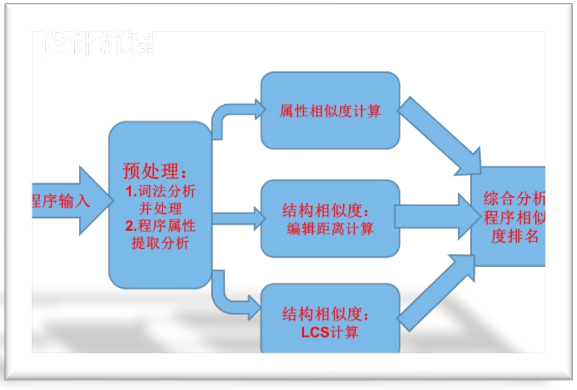

parse
网络爬虫
功能
在源代码中提取形如<a href="xxx" />中的xxx部分
通过有限状态自动机进行字符串匹配。

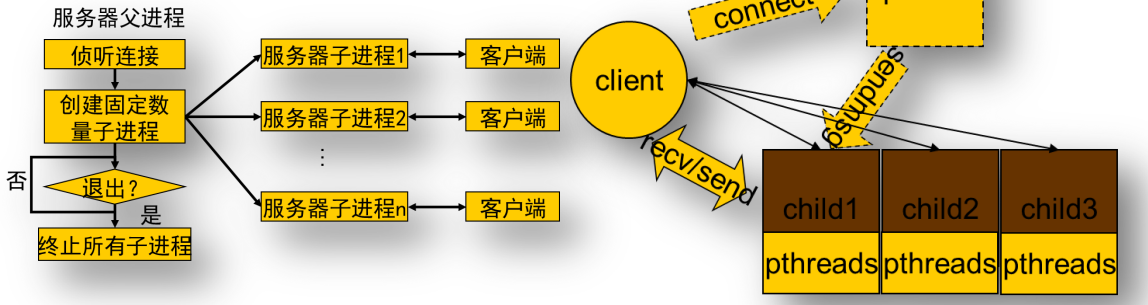

InputMethod
我爱北京天安门
我爱bei'jing'tian'an'men
1. 北京
2. 背景
3. 北
4. 被
5. 倍
6. 备
7. 背
8. 辈
9. 贝
0. 杯


开始
调用词库模块进行初始化
监听用户事件
拼音串改变
调用词库模块查询
显示新的候选词列表
选词或空格
结果显示，计算并显示剩余拼音串
翻页
翻页处理
回车
结果显示英文，拼音串清空
退出
结束

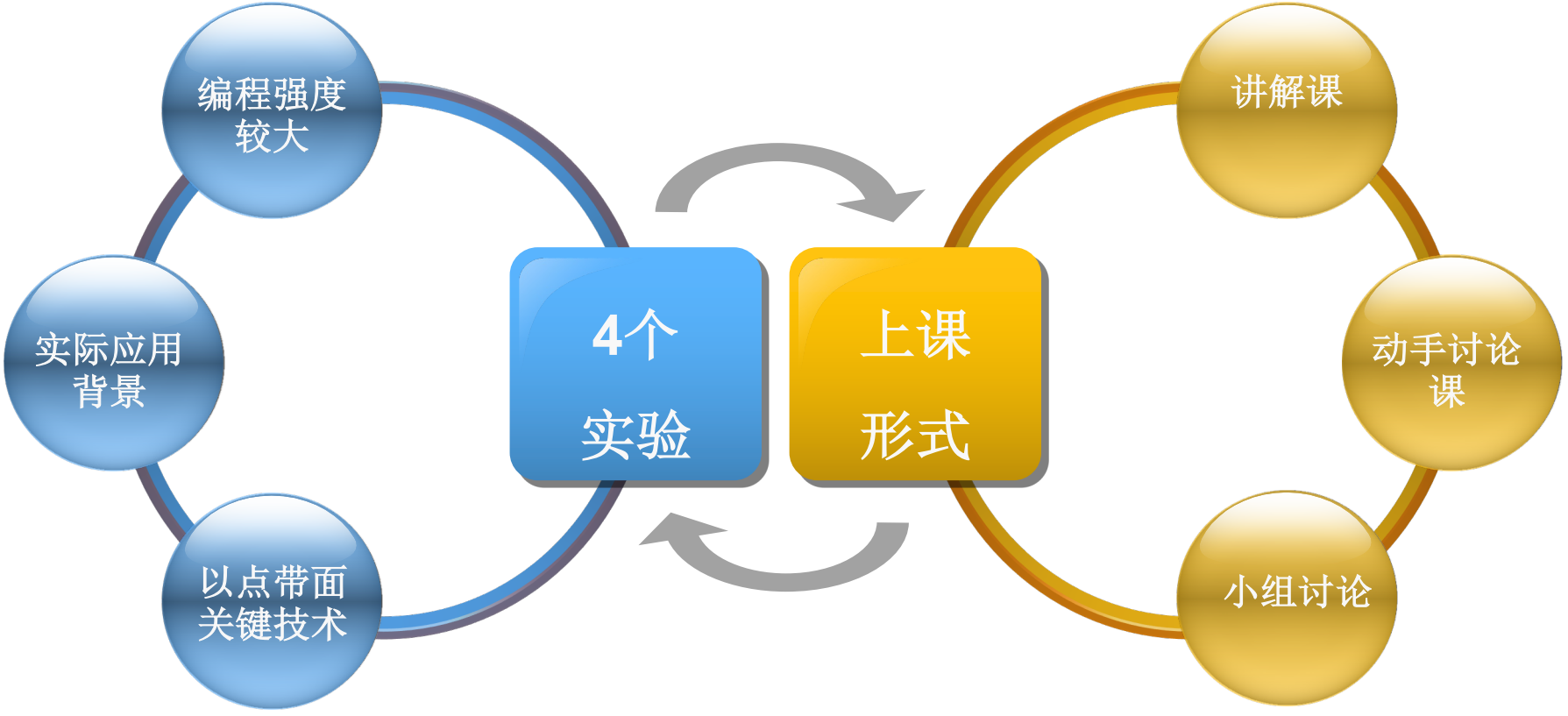

a b c d e f g h




预处理：
1.词法分析并处理
2.程序属性提取分析
属性相似度计算
结构相似度：编辑距离计算
结构相似度：LCS计算
综合分析程序相似度排名


服务器父进程
侦听连接
创建固定数量子进程
退出？
否
是
终止所有子进程
服务器子进程1
服务器子进程2
服务器子进程n
客户端
client
connect
parent
send/recv
child1
child2
child3
pthreads
pthreads
pthreads

课程组成



课程形式（1）：讲解课

- **讲解课**
  - 讲解整个实验的原理和代码
  - 演示实验的关键环节和代码
  - 规定每周的交付
    - 网站分析实验
      - W1：熟悉linux环境和socket编程，构建可以发送简单请求的程序
      - W2：搭建nginx和网站环境，开发可构造http请求协议的客户端，可以分析html并抽取超链
      - W3：熟悉libevent设计模式和epoll的io多路复用原理
      - W4：开发基于libevent的并行网页下载请求，并可输出所有网页到文件
      - W5：设计保存所有超链的图结构和文件结构，并进行持久化
      - W6：利用矩阵迭代算法计算pagerank并计算最重要的网页

课程形式（2）：讨论课

- 同学互相讨论
  - 一般两周一次
  - 现场运行程序展示进展并讲解代码进行讨论
  - 所有学生自带电脑现场编程展示结果
  - 现场展示代码和讲解

实验组成

- **实验1：Hash字符串检索**
  - Hash表、Bloomfilter
  - 3周
- **实验2：m叉查找树字符串检索**
  - 4叉trie树、二叉trie树（Patricia树）、压缩树
  - 3周
- **实验3：多模式字符串匹配**
  - Kmp算法→ac自动机
  - 3周
- **实验4：大规模图结构分析**
  - 根据检索的词共现关系构建图，计算节点的重要度
  - Textrank/pagerank
  - 3周

课程要求

- **编程环境**
  - 只允许用C语言或不利用高级语言库的C++语言
- **成绩计算**
  - 平时成绩：20分
  - 每个实验占20%权重
  - 单个实验：100分
    - 运行结果：
      - 未运行成功：<60分
      - 合理时间成功运行且功能正确：60分
    - 技术指标：20分（内存、比较次数等）
    - 实验报告：10分
    - 课堂表现：10分（实验分享、代码分享等）

热身1

在配置 4G 内存的 64 位计算机上，编写了一个如下程序，能否正确运行？（10）

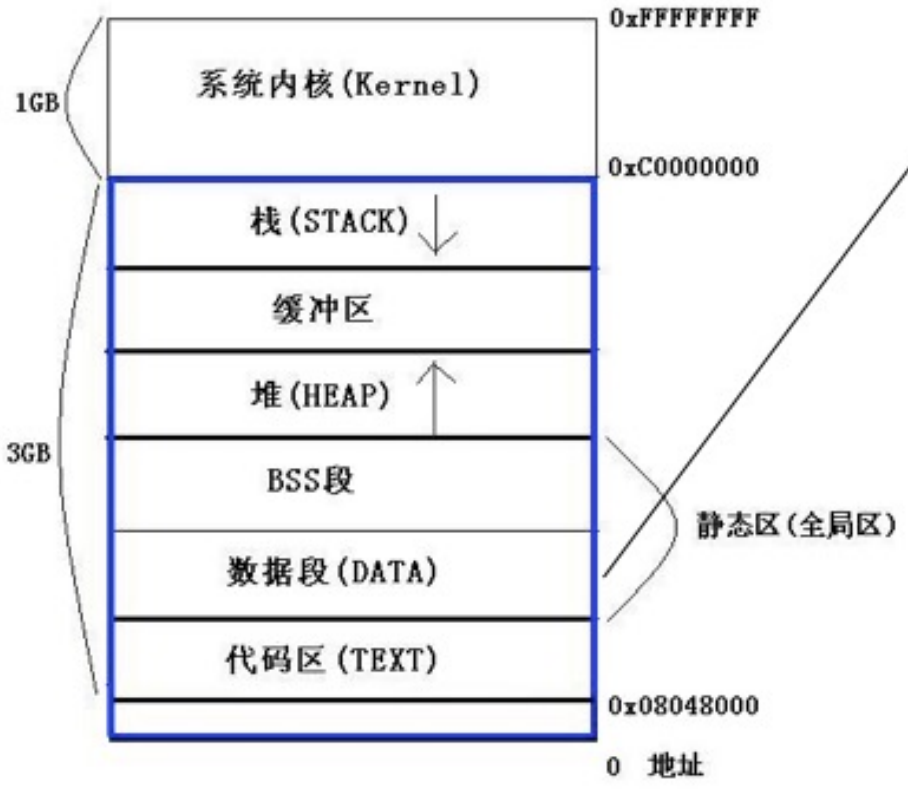
```
main() {
char *p=null; int i=0;
p=malloc(5*1024*1024*1024)//5G
for(i=0;i<5G;i++) //简写
    *(p+i) = 'A';
free(p);}
```

能否正确运行，如果错误，指出运行到哪一步可能出错。

答：

热身2

请将右边代码加粗定义的变量存储位置和左边内存对应区域画线连起来，例如变量 i1 存储在数据段，则画线连接 i1 和 DATA（15 分）




```
int i1 = 10;
static int i2 = 30;
void fun(int i5)
{
    int i3 = 60;
    static int i4 = 70;
    char* str1 = "ABCDE";
    char str2[] = "ABCDE";
    int* pi = malloc(sizeof(int));
}
int main(void){
    fun(50);
    return 0;}
```

热身3

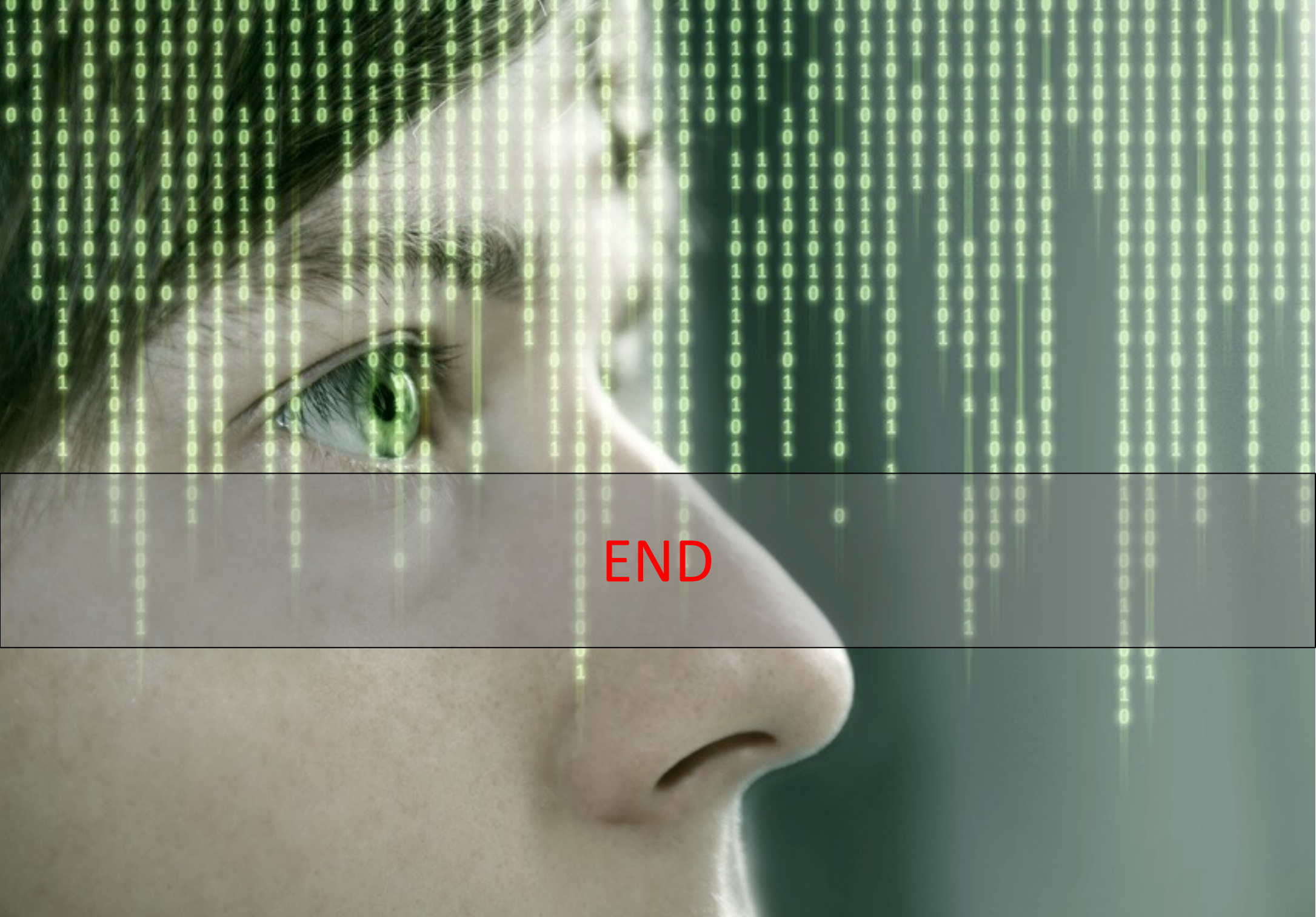
2. 以下四个程序运行 Test 函数结果分别是什么？（20）

```
void GetMemory(char *p)
{
p = (char *)malloc(100);
}
void Test(void)
{
char *str = NULL;
GetMemory(str);
strcpy(str, "hello world");
printf(str);
}
```

```
char *GetMemory(void)
{
char p[] = "hello world";
return p;
}
void Test(void)
{
char *str = NULL;
str = GetMemory();
printf(str);
}
```

```
Void GetMemory2(char **p, int num)
{
*p = (char *)malloc(num);
}
void Test(void)
{
char *str = NULL;
GetMemory(&str, 100);
strcpy(str, "hello");
printf(str);
}
```

```
void Test(void)
{
char *str = (char *) malloc(100);
strcpy(str, “hello”);
free(str);
if(str != NULL)
{
strcpy(str, “world”);
printf(str);
}
}
```


END