

2015-2016 学年夏季学期

《计算机程序设计实训》(0830A033) 安排

1 实习目的

在夏季学期第 3-4 两周，采取习题课辅导、课外小组编程、上机练习及验收、程序设计竞赛等方式，集中研究一定数量的 C 语言程序设计题，以强化同学们的 C 语言基础及综合程序设计能力。

2 课程班及研究小组

本课程属性为实践环节（必修），修课对象为计算机科学与技术专业 2015 级全体同学，按班实施教学。

在班级内每 3~4 位同学组成一个研究小组（自由组合）并推选一位组长（6 月 28 日上机时，报指导教师）。研究小组组长负责召集小组课外讨论（自行确定讨论地点）、实训任务分工、负责收集该组所有成员的程序代码并打包上传、负责汇集编撰课程报告（每小组一份）并提交。

各组成员（包括组长）应全程参与问题的研究工作，须合理分工，并在不同研究问题中轮换角色，例如：框架设计分析及文档编撰（1 名）、功能函数实现（程序员 1~2 名）、测试用例设计（测试员 1 名）。或者组长将任务划分，完成后组员之间相互测试其他组员的代码。通过这种方式，使每位同学都得到 C 语言基础、团队合作、沟通交流方面的训练。

表 1 实训班及相关场地

实训班	习题课教室	实训机房	指导教师（东校区办公室答疑）	
一班(35 人)	周一 AJ103 周三 BJ204	D406	李 青（计 1113）主讲习题课	导生名单 14121165 刘宇澄 14122061 李一帆 14123090 高 越 14121702 刘玉琦 14122931 张自强 14122808 沈亦文 14121840 徐际岚 14122540 张天翔 14121843 金钟镛 14122089 方昊堃
二班(35 人)		D406	李 颖（计 419）	
三班(35 人)		D410	朱 颖（计 505）	
四班(35 人)	周一 AJ104 周三 BJ104	D410	张博锋（计 1109）主讲习题课	
五班(35 人)		D415	赵俊娟（计 805）	
六班(35 人)		D415	沈云付（计 811）	
七班(35 人)	周一 AJ107	D503	宋 波（计 413）主讲习题课	
八班(35 人)	周三 BJ107	D504	张景峤（计 405）	

3 课程表

本课程学分数为 2，总共安排习题课、上机练习、程序上机验收、上机考试等约 40 学时（1 学时按 45 分钟计）。

表 2 实训课程表

第三周	星期一 (6月27日)	星期二 (6月28日)	星期三 (6月29日)	星期四 (6月30日)	星期五 (7月1日)
上午 (8:00-11:00)	实训动员 习题课(1)	上机 报分组情况	习题课(2)	上机	上机 征集各班问题
下午 (14:00-16:00)				答疑 (教师办公室)	
第四周	星期一 (7月4日)	星期二 (7月5日)	星期三 (7月6日)	星期四 (7月7日)	星期五 (7月8日)
上午 (8:00-11:00)	习题课(3) 讨论征集问题	上机		上机	上机验收程序 提交实训报告
下午 (14:00-16:00)		答疑 (教师办公室)	上机考试 12:00-16:00 (机房另行通知)		

注：① 表 2 中，空格处为同学们课外小组研究时间。

② 可以使用任何 C 语言程序集成开发环境，如 GUIDE、MinGW Developer Studio 等。

③ 各班学习委员负责收集本班同学们所提的问题，于 7 月 1 日交给指导教师。

4 考核办法

各小组在完成计算机程序设计实训内容后，由指导教师对小组进行考核。考察小组学生操作的熟练程度、各项任务的完成情况以及实训期间的表现、组织纪律性等情况，综合后按五级十一等评定学生成绩。

平时成绩	“C-字符串处理”验收	在线判题系统练习	上机考试成绩	课程报告
10%	20%	20%	20%	30%

注：① 平时成绩以考勤为主。上机考试以个人为单位。

② 验收情况、课程报告采用“小组自评权重，教师验判评分”的方法确定。例如，假设某小组由 3 位同学。在验收前，由该组同学协商确定各自的权重（权重总和须等于小组人数），如 0.94、1.0、1.06。若教师给出该小组的得分 85 分，则该小组 3 位同学的得分分别为 $0.94 \times 85 = 79.0$ 、 $1.0 \times 85 = 85$ 、 $1.06 \times 85 = 90.1$ 。

③ 验收标准：集成开发环境使用熟练程度(10%)、独立完成性(40%)、回答问题情况(50%)。

5 相关材料及提交要求

学院网站 (<http://cs.shu.edu.cn>) 上“通知公告”/“本科生”栏中发布有本课程的材料。

课程结束时，请各小组组长收集①“C-字符串处理”练习的源代码；②课程报告（电子版）。将上述内容打包压缩生成“2016 程序设计实训_组长学号姓名.rar”发至 reportshu@163.com 邮箱。纸质版课程报告（封面上各位同学签名）提交给指导教师。

实训题集

一、C-字符串处理

关于 C-字符串的练习是理解和掌握指针、函数的极佳素材。要求同学们用一个个独立的函数实现实训题的功能。请组长合理安排组员的任务分工,并在不同练习题中使组员角色轮换。

当每一个功能(即函数)编制完成后,要求负责测试的组员编写测试程序、用各种“刁钻”的参数去测试这些函数,特别是一些特殊情况(如:长度为 0 的空字符串、长度为 1 的字符串、字符串常量、存放于堆空间的字符串等),以证明所编写函数的正确性、健壮性。要求测试程序的界面尽可能“友好的”。当需要输入的时候,应该有合适的、明确的提示(包括输入格式,输入范围)。便于提高教师验收时的工作效率。

1 C-字符串容器

【基础知识】

C-字符串按字符数组的形式存放在计算内存中,并且以特殊字符'\0'作为串结束标志。其中的字符数组被称为字符串的容器,可以是一般数组、动态数组(即堆数组)或常量数组。C-字符串的容器常常需要程序员设置,且其空间需要“足够大”。

由于 C-字符串在内存中连续存放,又有串结束标志字符,并且认为其容器的容量已经设置妥当还足够大,故在 C 语言程序中,常将 **char***或 **const char***类型理解为 C-字符串及 C-字符串常量。

【练习 1】根据程序的运行结果,完成 4 个待完成的 C-字符串处理函数的设计。请参见文件夹 **Codes\C_String01**,其中的文件说明如下。

文件名	说明
C_String01.mdsp	新版 MinGW Developer Studio 工程文件
C_String01.msp	旧版 MinGW Developer Studio 工程文件(计算中心机房适用)
test01.c	含主函数的测试程序
C_String01.h	用户自定义的头文件(主要用于自定义函数声明)
C_String01.c	用户自定义的源程序文件(主要用于自定义函数定义)
C_String01.exe	完成程序后的可执行文件(供参考)

程序的运行结果

<pre>C-String test. Your choice(1--4, 0=quit): <u>1</u> in test1() str1: "Tom", str2: "Jerry" str1: "Jerry", str2: "Tom"</pre>
--

```

return from test1()

C-String test. Your choice(1--4, 0-quit): 2
in test2()
p1: "Tom",      p2: "Jerry"
p1: "Jerry",    p2: "Tom"
return from test2()

C-String test. Your choice(1--4, 0-quit): 3
in test3()
str1: "Tom",    str2: "Jerry"
str1: "Jerry",  str2: "Tom"
return from test3()

C-String test. Your choice(1--4, 0-quit): 4
in test4()
str1: "Tom",    str2: "Jerry"
str1: "Tom & Jerry"
return from test4()

C-String test. Your choice(1--4, 0-quit): 0

```

2 C-字符串处理函数功能测试

【基础知识】

处理 C-字符串本质上是对数组和数组元素进行操作。为方便程序员，C 语言提供了一些函数进行 C-字符串处理。函数的形式参数常常设计为 **const char***或 **char***，即仅传递字符串的首地址给函数，函数体从该地址处逐个字符进行处理，直到遇到字符'\0'为止。

【练习 2.1】写出 scanf, printf, gets, puts 的函数原型，说明其形式参数、返回值的含义，说明 scanf 与 gets 的区别，并测试这些函数的功能。

【练习 2.2】写出 atoi, atol, atof, sscanf, sprintf 的函数原型，说明其形式参数、返回值的含义，并测试这些函数的功能。

【练习 2.3】写出 strlen, strcpy, strncpy, strcat, strncat, strcmp,strupr, strlwr 的函数原型，说明其形式参数、返回类型的含义（特别说明 const 的作用），并测试之。

【要求】①既有常规的字符串测试；②又有一些特殊的字符串进行测试（例如：空字符串、长度为 1 的字符串、两个长度不相等的字符串的测试）；③还有考虑字符串的容器所在地（常量池、栈区、堆区等）。甚至用一些错误的用法（如：字符指针为空、指向不可访问的地址、容器容量不够大）进行测试以说明 C-字符串使用中应该注意的事项。

【提示】使用动态数组时，勿忘使用完毕后释放（free）。

3 自定义 c-字符串处理函数及功能测试

【练习 3】自定义如下函数 `Atoi`、`StrUpr`、`StrLwr`、`StrLen`、`StrCpy`、`StrNCpy`、`StrCat`、`StrCmp` 分别模拟 `atoi`、`strupr`、`strlwr`、`strlen`、`strcpy`、`strncpy`、`strcat`、`strcmp` 函数。利用练习 2.2、练习 2.3 的测试数据测试之（每位组员完成若干函数，然后相互测试）。

4 C-字符串数组排序函数设计

【基础知识】有两种方式存储字符串数组：①二位字符型数；②字符型指针数组。不同的存储，其读/写方式是不同的。

【练习 4】按两种排序规则分别各设计两个函数可对如下表示的字符串数组进行排序。

```
char str[12][8]={"enter", "number", "size", "begin", "of", "cat",  
                "case", "program", "centain", "a", "cake", "side"};  
char *str[12] = {"enter", "number", "size", "begin", "of", "cat", "case",  
                "program", "centain", "a", "cake", "side"};
```

- 1) 按字典序排序，并输出；
- 2) 按字符串的长度（从短到长）排序，若长度相等则按内容（字典序）排序。

【说明】本练习实际上需要编写 4 个排序函数（每个组员主编一个，然后交流、比较其特点），另加两个字符串数组的输出函数、一个测试用的主函数。

二、程序设计在线判题系统练习

网址 <http://202.121.199.212/JudgeOnline/>。

【提示】第四周上机考试将采用在线判题系统。请大家尽快掌握在线判题系统的使用方法，特别是数据输入/输出方法。下面的 5 个示例程序参见文件夹 `Codes`。

1 在线判题系统解题完整示例

【例题 1】（包括题目、解题程序、自测方法的完整示例）阶乘尾部 0 的个数。

阶乘尾部 0 的个数

问题描述 给定一个正整数 n ，计算 $n!$ 的尾部 0 的个数。

输入

输入数据有若干行。每行上有一个正整数 n ，对应一种情形。

输出

对于每一种情形，先输出 “Case #: ”（#为序号，从 1 起），然后输出 n 的值，逗号，结果，换行。

输入样例

```

8
16
30
输出样例
Case 1: 8, 1
Case 2: 16, 3
Case 3: 30, 7

```

图 1 题目示例

在线判题系统的后台服务器上，有实际使用的测试数据文件（假设 Input01.txt 及由标准程序产生的标准结果文件 Output01.txt，它们的内容如下。当然，被测人员是无法获得这两个文件的）。

Input01.txt	Output01.txt
8	Case 1: 8, 1
16	Case 2: 16, 3
30	Case 3: 30, 7
4	Case 4: 4, 0
5	Case 5: 5, 1
24	Case 6: 24, 4
25	Case 7: 25, 6
50	Case 8: 50, 12
150	Case 9: 150, 37
200	Case 10: 200, 49
0	Case 11: 0, 0
10000	Case 12: 10000, 2499

图 2 实际测试数据文件及标准结果文件的内容

由此可见①题目中的样例数据仅是实测数据的一部分；②实测数据中存在一些特殊情况（如：0 的阶乘，10000 的阶乘）。我们知道，用 **unsigned long** 类型计算阶乘，13! 就会溢出而出错。

【参考解答 • C 语言源程序——OJsample01.c】

由输入样例可知，测试数据每行的数据个数是明确已知的（每行一个整数），但行数未知。

C 语言用于输入的函数 `scanf` 返回所读取到的数据个数。例如 `scanf("%d %d", &m, &n)` 返回 2 则表示正确地读取了两个整数；返回 1 则表示仅读取到一个有效数据；返回 0 则表明没有

读取到有效数据。另一方面,当读到“文件结束标志”(Windows 操作系统下,键盘输入为 Ctrl+Z。Linux/UNIX 操作系统下,键盘输入为 Ctrl+D)。利用这些特性可判断是否读完所有测试数据。

```
// OJsample01.c 阶乘尾部 0 的个数
#include <stdio.h>

int zeros(int n)
{
    int m=0, p=5;
    while(p<=n)
    {
        m += n/p;
        p *= 5;    // p 的变化规律为 5 的 1,2,3,...次方
    }
    return m;
}

int main()
{
    int n, k=0;

    while(scanf("%d", &n)==1)
    {
        printf("Case %d: %d, %d\n", ++k, n, zeros(n));
    }
    return 0;
}
```

图 3 示例解题参考程序

【参考解答·C++语言源程序——OJsample01.cpp】

由输入样例可知,测试数据每行的数据个数是明确已知的(每行一个整数),但行数未知。C++中抽取运算符(>>)在默认情况下会滤掉数据前的空白字符(一个或多个空格字符、Tab 字符、换行字符)。当正确地读取到数据时运算结果为非零(即为 **true**),此时,数据后面的空白字符(如最常见的是换行字符)仍留在输入缓冲区中等待下一次输入操作。当不能正确读取到数据时(如抽取到文件结束标志,或键盘输入 Ctrl+Z),抽取操作的结果为 0(即为 **false**)。利用这一特性,可将抽取运算表达式 `cin >> n` 作为继续循环的条件。

```
// OJsample01.cpp 阶乘尾部 0 的个数
#include <iostream>
using namespace std;
int zeros(int n)
{
```

```

    int m=0, p=5;
    while(p<=n)
    {
        m += n/p;
        p *= 5;          // p 的变化规律为 5 的 1,2,3,...次方
    }
    return m;
}
int main()
{
    int n, k=0;

    while(cin >> n)
    {
        cout << "Case " << ++k << ": " << n
              << ", " << zeros(n) << endl;
    }
    return 0;
}

```

图 4 示例解题参考程序

【提交前的自测方法之一】

利用一种 C++编译器（如 MinGW、Dev-C++等集成开发环境）将上述源代码编译、连接成可执行文件 OJsample01.exe。在与文件 OJsample01.exe 所在的文件夹中建立如下两个文件。

(1) 文件 OJsample01in.txt，其内容为题目中测试样例的数据。

```

8
16
30

```

图 5 测试样例数据文件（根据题目中的测试样例自测用）

(2) 文件 run.bat 的内容如下。其功能是启动 OJsample01.exe 文件执行，并将标准输入设备重新定向为 OJsample01in.txt 文件（相当于从键盘输入了文件 OJsample01in.txt 的内容，输入所有内容后再输入了组合键[Ctrl+Z]）。并且将程序的标准输出重新定向为 OJsample01out.txt 文件（相当于将 OJsample01out.txt 文件作为显示器）。

```

OJsample01.exe < OJsample01in.txt > OJsample01out.txt

```

图 6 批处理文件（自测用）

(3) 双击 run.bat 文件后，将产生 OJsample01out.txt 文件，其内容如下。


```
Case 1: 8, 1
Case 2: 16, 3
Case 3: 30, 7
```

图 7 执行批处理文件后产生的结果文件

比较 OJsampl01out.txt 文件的内容与题目中的输出样例（一致），表明所设计的程序对测试样例数据处理的结果正确。进而可考虑提交给在线判题系统评判了。

【提交前的自测方法之二】

另一种自测方法：直接运行程序；从键盘上依次输入测试样例数据，则逐步在显示器上输出结果，观察这些结果是否与“输出样例”一致；最后输入组合键 **[Ctrl+Z]** 表示测试数据输入完毕，结束测试。

请注意：此时输入与输出交织在一起，属于正常现象。

2 在线判题系统解题输入/输出典型方法

【例题 2】判断算是的正确性

判断算式的正确性

问题描述 给定一个算式，该算式中只含一个四则运算符号，操作数及结果均为整数。要求判断该算式的正确性（规定：除法必须除尽才可能正确）。

输入

输入数据有若干行。每行上有一个算式，对应一种情形。

输出

对于每一种情形，先输出“Case #： ”（#为序号，从 1 起，冒号后输出一个空格），然后输出 Yes（表示算式正确）或 No（表示算式错误），最后换行。

输入样例

```
1 + 2 = -3
5 / 2 = 2
4 / 2 = 2
```

输出样例

```
Case 1: No
Case 2: No
Case 3: Yes
```

【分析】输入样例中，每种情形中的数据个数都是 5，分别是：第一个操作数（整型）、运算符号（字符型）、第二个操作数（整型）、等号（字符型）、结果（整型）。解答如下。

```
// OJsampl02.c 判断算式的正确性
#include <stdio.h>
```

```

int main()
{
    int x, y , z, k=0;
    char op, eq;

    while(scanf("%d %c %d %c %d", &x, &op, &y, &eq, &z)==5)
    {
        // 直接接收 5 个输入的数据，类型不尽相同
        printf("Case %d: ", ++k);

        if(op=='+' && x+y==z || op=='-' && x-y==z
            || op=='*' && x*y==z || y!=0 && x%y==0 && x/y==z)
            printf("Yes\n");
        else
            printf("No\n");
    }
    return 0;
}

```

【例题 3】给定 n 及 n 个实数，求该 n 个实数中的最大值。

求给定的 n 个实数中的最大值

问题描述 给定整数 n 及 n 个实数，找出其中的最大值。

输入

输入数据有若干行。每行上第一个数为整数 n ($n>0$)，紧接其后有 n 个实数，对应一种情形，数据之间用空格分隔。

输出

对于每一种情形，先输出 “Case #: ” (#为序号，从 1 起，冒号后输出一个空格)，然后输出计算结果（保留 1 位小数），最后换行。

输入样例

```

3 1.0 3.0 2.0
5 10.0 8.5 15.3 7.8 5.8
1 3.3

```

输出样例

```

Case 1: 3.0
Case 2: 15.3
Case 3: 3.3

```

【分析】输入样例中，每种情形中的数据个数都已告知，故先接收整数 n 的输入，然后用计数循环接收其他数据的输入。须注意的是读取第一个数直接存入 `max` 中。解答如下。

```
// OJsample03.c 求一系列数据的最大值
#include <stdio.h>

int main()
{
    int i, n, k=0;
    double x, max;

    while(scanf("%d", &n)==1)
    {
        printf("Case %d: ", ++k);
        scanf("%lf", &max);          // 读取第一个数据
        for(i=1; i<n; i++)
        {
            scanf("%lf", &x);        // 再循环读取 n-1 个数据
            if(x>max)
                max = x;
        }
        printf("%.1f\n", max);
    }
    return 0;
}
```

【例题 4】计算若干个整数的和。

计算若干个整数的和

问题描述 对于给定的若干个整数，要求计算它们的项数及和。

输入

输入数据有多行。每一行有若干个整数(个数不超过 1024)，希望计算它们的项数及总和。

输出

对于每一行中的数据，要求先输出“Case #：”（#为序号，从 1 起，冒号后输出一个空格），然后输出该行上数据的个数、逗号、空格、总和、换行。

输入样例

```
15 3 42 89
51 201 303 9 755 800
```

输出样例

```
Case 1: 4, 149
Case 2: 6, 2119
```

【分析】本题的难点在于如何读取一行中的多个数据，且数据个数不定。对于这种情形，可以每读取一个数值、接着读取一个字符，判断该字符是否为换行字符('\n')。解答如下。

```
// OJsample04.c  计算若干个整数的项数及总和
#include <stdio.h>

int GetData(int *p)
{
    int n, flag=1;
    char c;
    for(n=0; flag && scanf("%d%c", p+n, &c)==2; n++) // p+n, &p[n]
    {
        // 注意此处"%d%c"中间不要插入空格
        if(c=='\n') flag=0;
    }
    return n;
}

int Sum(int *p, int n)
{
    int sum=0, i;
    for(i=0; i<n; i++)
        sum += p[i];
    return sum;
}

int main()
{
    const int N = 1024;
    int array[N], n, k=0;

    while((n=GetData(array))!=0)
        printf("Case %d: %d, %d\n", ++k, n, Sum(array, n));
    return 0;
}
```

【例题 5】判断回文串。

判断回文

问题描述 判断给定的字符串是否为回文。所谓回文是指一个字符串，将其从左到右读或从右到左读是相同的。

输入

输入数据的第一行为一个整数 n ，其后共有 n 行待判断的字符串（字符串的长度不超过 1000）表示有 n 种情况。

输出

对于每一种情况，要求先输出 “Case #: ”（#为序号，从 1 起，冒号后输出一个空格），然后输出 “YES.”（若对应的字符串为回文）或 “NO.”（若对应的字符串不是回文）、换行。

输入样例

```
5
123 456 789 0 987 654 321
abcdefghijklmn
level
Abba
X
```

输出样例

```
Case 1: YES.
Case 2: No.
Case 3: YES.
Case 4: No.
Case 5: YES.
```

【分析】本题的难点在于用 `scanf` 函数读取整数 n 后，输入缓冲区中留有的字符为整数 n 后面的换行字符。这个字符需要单独处理，即读取后丢弃或忽略。另一个需要注意的是字符串中含有空格字符，因而读取一行字符串应该使用 `gets` 函数。解答如下。

```
// OJsample05.c 判断回文
#include <stdio.h>
#include <string.h>

int Palindrome(const char *str)
{
    int left=0, right=strlen(str)-1;
    while(left<right && str[left]==str[right])
    {
        left++;    right--;
    }
}
```

```

        return !(left<right);
    }

    int main()
    {
        char str[1000];
        int k, n;

        scanf("%d", &n);
        gets(str);          // 读取整数 n 后面的、直至换行字符的所有字符（不用）
        for(k=1; k<=n; k++)
        {
            gets(str);
            printf("Case %d: %s\n", k, (Palindrome(str)? "YES." : "NO."));
        }
        return 0;
    }

```

3 在线判题系统练习题

【练习 1】 $3n+1$ 问题。

3n+1 问题

问题描述 对于给定的一个正整数，若该数为偶数则将其除以 2，若为奇数则将其乘 3 再加 1。反复进行上述过程，直到结果为 1 时停止。这就是著名的“ $3n+1$ ”问题。要求对于给定的整数，求按 $3n+1$ 规则变换到 1 所需要的数字变换次数。

输入 输入数据有多个各分别对应一种情形。

输出 对于每一个整数，要求先输出“Case #：”（#为序号），然后输出读入的整数、逗号、空格、计算结果。若输入的数据为非正整数，结果为-1。

输入样例

1024 1023 0 100 66

输出样例

Case 1: 1024, 10

Case 2: 1023, 62

Case 3: 0, -1

Case 4: 100, 25

Case 5: 66, 27

【练习 2】折纸的厚度。

折纸的厚度

问题描述 假定有一张充分大的纸，并且可以进行许多次对折。若单张纸的厚度为 x ，计算经过多少次对折后，折纸的厚度超过珠穆朗玛峰的高度 8844.43 米。

输入 输入数据有多个，各对应一种情形，表示单纸张的厚度（以毫米为单位）。

输出 对于每一种情形，先输出“Case #： ”（#为序号，从 1 起），然后输出单张纸张的厚度、需要对折的次数、总厚度（以米为单位），换行。浮点型数据保留 1 位小数输出。

输入样例

0.1 0.3 0.5

输出样例

Case 1: 0.1, 27, 13421.8

Case 2: 0.3, 25, 10066.3

Case 3: 0.5, 25, 16777.2

【练习 3】计算 $S(a, n) = a + aa + aaa + \overbrace{aaa \cdots a}^{n \text{ 个 } a}$ 的值。

计算 $S(a, n)$

问题描述 给定整数 a ($1 \leq a \leq 9$) 以及正整数 n ，计算 $S(a, n) = a + aa + aaa + \overbrace{aaa \cdots a}^{n \text{ 个 } a}$ 。

输入 输入数据有多行，每行上有两个整数 a 及 n 对应一种情形。

输出 对于每一种情形，先输出“Case #： ”（#为序号，从 1 起），然后结果。

输入样例

2 5

1 6

5 2

输出样例

Case 1: 24690

Case 2: 123456

Case 3: 60

【练习 4】判断同构数。

判断同构数

问题描述 给定一个十进制正整数，判断其是否为同构数。所谓同构数是这样的一些数，它出现在其平方数的右边，例如 5 是 $5^2 = 25$ 右边的数，25 是 $25^2 = 625$ 右边的数，所以 5 和 25 都是同构数。

输入 输入数据有若干行。每行上有一个正整数，对应一种情形。

输出 对于每一种情形，先输出“Case #:”（#为序号，从 1 起），然后输出 n，逗号，结果（Yes 或 No），换行。

输入样例

5
15
25

输出样例

Case 1: 5, Yes
Case 2: 15, No
Case 3: 25, Yes

【练习 5】算术平均值与几何平均值。

算术平均值与几何平均值

问题描述 给定若干个（不超过 10）实数，计算其算术平均值及几何平均值。

输入 输入数据有多行，每行有若干个浮点型数据代表一种情形。

输出 输出算术平均值、逗号、空格、几何平均值、换行。浮点型数据保留 1 位小数。请注意数据之积为负数时，无法开偶数次方（参见样例 4）。

输入样例

1.5 2.0 6.2 4.3 5
1.5 2.0 6.2 4.3
-1.5 -2.0 6.2 -4.3 5
-1.5 2.0 6.2 4.3

输出样例

Case 1: 3.8, 3.3
Case 2: 3.5, 3.0
Case 3: 0.7, -3.3
Case 4: 2.8, None

【练习 6】乘方计算。

乘方计算

问题描述 给定一个正的 **signed long** 型数据 x ，计算 x^2, x^3, \dots 直到 x^5 或溢出为止。

输入 输入数据有多个，各表示一种情形。

输出 依次输出 x, x^2, x^3, \dots 的值，直到 x^5 或溢出为止。最后换行（注意最后无逗号）。

输入样例

111 500 1000 5000

输出样例

Case 1: 111, 12321, 1367631, 151807041

Case 2: 500, 250000, 125000000

Case 3: 1000, 1000000, 1000000000

Case 4: 5000, 25000000

【练习 7】任意数制转换。

任意数制转换

问题描述 给定一个十进制正整数，将其转换成指定的进制形式。

输入 输入数据有若干行，每一行有两个正整数 n, m （其中 $2 \leq m \leq 36$ ）对应一种情形。

输出 对于每一种情形，先输出“Case #:”（#为序号，从 1 起），输出正整数 n 的 m 进制表示形式。36 进制的基数依次为 $0, 1, \dots, 9, A, B, \dots, Z$ （大写）。

输入样例

100 2

100 20

101 36

输出样例

Case 1: 1100100

Case 2: 50

Case 3: 2T

【练习 8】奇数阶幻方。

奇数阶幻方

问题描述 给定一个正奇数 n ，输出 n 阶幻方。即：将 $1, 2, \dots, n^2$ 填入一个 $n \times n$ 方阵，使方阵的每一行元素的和、每一列元素的和、对角线元素的和均相等。奇数阶幻方可按如下方式排成：将 1 列于第一行正中，依次将自然数排在其前一个数的右上方（按环绕边界处理）；遇到 n 的倍数时，将下一个自然数排在前一个数的下方。

输入 输入数据有若干行。每一行上有一个正奇数 n ($1 \leq n \leq 51$) 对应一种情形。

输出 对于每一种情形，先输出“Case #:”（#为序号，从 1 起），然后输出结果。参见输出样例，每个数据按域宽“%5d”（C++用 setw(5)）输出。

输入样例

3
5

输出样例

Case 1:

8	1	6
3	5	7
4	9	2

Case 2:

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

【练习 9】统计英文单词个数。

统计英文单词个数

问题描述 给定一个字符串，统计其中单词的个数。

输入 输入数据有若干行，每行上有一个字符串对应一种情形。字符串中分隔单词的字符仅考虑空格（可能是连续的空格，无标点符号）。字符串的长度不超过 65536 字符。

输出 对于每一种情形，先输出“Case #:”（#为序号，从 1 起），然后输出结果，换行。

输入样例

This is a test

BASIC FORTRAN Pascal C C++ Java

输出样例

Case 1: 4

Case 2: 6

【练习 10】删除一系列数据中的重复数据。

删除一系列数据中的重复数据

问题描述 给定一系列整型数据，删除其中重复出现的数据使各种数据仅保留一份（可考虑用单向链表或数组实现）。

输入 输入数据有多行，每行有若干个整数，每行总字符数不超过 65536 字节。

输出 对于每一种情形，先输出“Case #:”（#为序号，从 1 起），输出处理结果，换行。

输入样例

```
86 75 90 78 78 93 88 80 86 78
78 85 83 81 85 76 76 92
```

输出样例

```
Case 1: 86 75 90 78 93 88 80
Case 2: 78 85 83 81 76 92
```