# 程序设计实训练习1

# 整数倒序1

题目描述

输入一个整数，按倒序输出这个整数各个未上的数字。

输入

输入数据仅一行，包含一个整数n。0<=n<=2.1*10^9

输出

输出仅一行，为倒序的整数n各个位上的数字，每两个数字之间用一个空格间隔开

```
#include”stdio.h”

int main()
{
        int n;
        scanf(“%d”,&n);
        while(n>=10)
        {
                printf(“%d ”,n%10);
                n/=10;
        }
        printf(“%d\n”,n);
}
```

# 整数倒序2

题目描述

输入一个整数，将这个数字的各个位数倒序，组成一个新的数字，输出这个新数。

输入

输入数据仅一行，包含一个整数n。0<=n<=2.1*10^9

输出

输出仅一行，为将整数n倒序后组成的新的数字。

```
#include”stdio.h”

int main()
{
        int n,m=0;
        scanf(“%d”,&n);
        while(n)
        {
                m=m*10+n%10;
                n/=10;
        }
        printf(“%d\n”,m);
}
```

能通过80%的数据

尝试输入数据2047483647，整型能表示的最大数字为2147483647

```
#include”stdio.h”
int main() {
        int n,m=0;
        scanf(“%d”,&n);
        if(!(n%10)) {
                while(n) {
                        m=m*10+n%10;
                        n/=10;
                }
                printf(“%d”,m);
        }
        else
                while(n) {
                        printf(“%d”,n%10);
                        n/=10;
                }
        printf(“\n”);
}
```

如果结尾有0，则各位数字倒序相加之后得到的新数字一定在整型表示范围内。

如果结尾没有0，直接倒序输出，不需要做加法。

```
#include”stdio.h”
int main()
{
        int n,m=0;
        scanf(“%d”,&n);
        while(n%10==0) n/=10;
        while(n)
        {
                printf(“%d”,n%10);
                n/=10;
        }
        printf(“\n”);
}
```

自右向左，从第一个非0位置开始输出。

```
#include”stdio.h”
int main()
{
        int n,i;
        char c[11];
        gets(c);
        n=strlen(c)-1;
        while(c[n]==’0’)  n--;
        for(i=n;i>=0;i--)  printf(c[i]);
        printf(“\n”);
}
```

# 有趣的图形

题目描述

输入一个整数n，输出n行的由*组成的直角三角形

输入

一个整数n，n<20

输出

n行的由*组成的直角三角形

```
#include <stdio.h>

int main(void)
{
    int n,i,j;
    scanf("%d",&n);
    for(i=0; i<n ; ++i)
    {
        for(j=0; j<=i; ++j)       putchar('*');
        putchar('\n');
    }

    return 0;
}
```

# 有趣的图形2

题目描述

输入一个整数n，输出2n-1行的由*组成的菱形

输入

一个整数n，n<20

输出

输出2n-1行的由*组成的菱形

```
#include <stdio.h>

int main(void) {
    int n, i, j, t;
    scanf("%d", &n);
    for(i=0; i<n; ++i)  {
        for(j=n-i-1; j>0; --j)      putchar(' ');
        for(j=i*2+1; j>0; --j)     putchar('*');
        putchar('\n');
    }
    for(i=n-2; i>=0; --i)  {
        for(j=n-i-1; j>0; --j)       putchar(' ');
        for(j=i*2+1; j>0; --j)      putchar('*');
        putchar('\n');
    }
    return 0;
}
```

将输出的图形上下分隔为一个正的三角形和一个倒的三角形

同时左右分隔成空格部分和*部分

分别使用循环语句控制输出

# 有趣的图形3

题目描述

输入两个整数m和n，输出以m和n分别为长和宽的由*围成的矩形

输入

输入仅一行，包含两个整数，m和n，0<m,n<=2.1*10^9

输出

输出m行n列的，由*围成的矩形

```
#include <stdio.h>

int main(void) {
    int m, n, i, j;
    scanf("%d%d", &n, &m);
    for(i=1; i<=n; ++i) {
        for(j=0; j<m; ++j)  {
            if(i==1 || i==n)    putchar('*');
            else if(j==0 || j== m-1)     putchar('*');
            else    putchar(' ');
        }
        putchar('\n');
    }
    return 0;
}
```

# 字符统计

题目描述

输入一串字符，以#作为结束标志，统计这串字符中英文字母、数字和其它字符的个数

输入

输入数据为若干个字符，以#作为输入结束的标志

输出

分别输出英文字母个数，数字个数，和其它字符个数，每两个数字之间用一个空格隔开

#不统计

```
#include <stdio.h>
#include <ctype.h>

int main(void)  {
    int alpha=0, num=0, other=0;
    char c;
    while((c=getchar())!='#') {
        if(isalpha(c))
            ++alpha;
        else if(isdigit(c))
            ++num;
        else
            ++other;
    }
    printf("%d %d %d", alpha, num, other);
}
```

# 完数

题目描述

一个数如果恰好等于它的所有真因子的和，这个数就称为完数。例如6的真因子有1、2、3，而6=1+2+3，所以6是一个完数。编程找出一定范围内的所有完数

输入

输入数据仅一行，包含两个数字m和n，保证0<=m<=n<=2.1*10^9

输出

输出[m,n]之间的所有完数的个数

# ·程序运行时间初探——循环的代价

输入$n$，计算$S=1!+2!+3!+\ldots+n!$的末6位（不含前导0）。$n\leq 10^6$，$n!$表示前$n$个正整数之积。

样例输入：

10

样例输出：

37913

# ·程序运行时间初探——循环的代价

```
#include<cstdio>

int main()
{
        int n, s = 0;
        scanf("%d", &n);
        for(int i = 1; i <= n; i++)
        {
                int factorial = 1;
                for(int j = 1; j <= i; j++)
                        factorial *= j;
                s += factorial;
        }
        printf("%d\n", s % 1000000);
        return 0;
}
```

请分别输入n=10,n=30000进行测试

如果输入n=10^6会怎样？

# ·程序运行时间初探——循环的代价

```
#include<stdio.h>
#include<time.h>
int main()
{
        const int MOD = 1000000;
        int n, S = 0;
        scanf("%d", &n);
        for(int i = 1; i <= n; i++)
        {
                int factorial = 1;
                for(int j = 1; j <= i; j++)
                factorial = (factorial * j % MOD);
                S = (S + factorial) % MOD;
        }
        printf("%d\n", S);
        printf("Time used = %.2f\n", (double)clock() / CLOCKS_PER_SEC);
        return 0;
}
```

键盘输入的时间也被计算在内——因为键盘输入是在程序启动之后才进行的。为了避免输入数据的时间影响测试结果，可使用如下技巧：

1、使用time.h头文件中的clock（）函数获得程序运行时间。
注意：常数CLOCKS_PER_SEC 和操作系统相关，请不要直接使用clock（）的返回值，而应总是除以CLOCKS_PER_SEC，获得程序运行的秒数

2、在Windows命令行下执行echo 10|abc，系统会自动把10输入，其中abc是可执行文件名。

# ·程序运行时间初探——循环的代价

表2-1　程序2-8的输出结果与运行时间表

| $n$ | 20 | 40 | 80 | 160 | 1600 | 6400 | 12800 | 25600 | 51200 |
|---|---|---|---|---|---|---|---|---|---|
| 答案 | 820313 | 940313 | 940313 | 940313 | 940313 | 940313 | 940313 | 940313 | 940313 |
| 时间 | <0.01 | <0.01 | <0.01 | <0.01 | 0.05 | 0.70 | 2.70 | 11.08 | 43.72 |

由表2-1可知：第一，程序的运行时间大致和$n$的平方成正比（因为$n$每扩大1倍，运行时间近似扩大4倍）。甚至可以估计$n = 10^6$时，程序大致需要近5个小时才能执行完。

我们可以通过以上规律估算出当$n$＝10^6时，程序大致需要近5个小时才能执行完。通常，程序设计竞赛在不作特殊说明的情况下都要求在1S内运行完成并输出结果，大量的循环会消耗大量的时间

算法的**时间复杂度**是一个函数，它定量描述了该算法的运行时间。
通常我们只考虑算法最大时间复杂度的数量级，此时的时间复杂度常用大O符号表述。
如上题的算法时间复杂度可写作O(n^2)

# 完数

题目描述

一个数如果恰好等于它的所有真因子的和，这个数就称为完数。例如6的真因子有1、2、3，而6=1+2+3，所以6是一个完数。编程找出一定范围内的所有完数

输入

输入数据仅一行，包含两个数字m和n，保证0<=m<=n<=2.1*10^9

输出

输出[m,n]之间的所有完数的个数

```c
#include <stdio.h>

int main(void)
{
	int total=0,m,n;

	scanf("%d%d",&m,&n);
	if (m<=6 && n>=6) total++;
	if (m<=28 && n>=28) total++;
	if (m<=496 && n>=496) total++;
	if (m<=8128 && n>=8128) total++;
	if (m<=33550336 && n>=33550336) total++;

	printf("%d\n",total);
}
```

# 水仙花数1

题目描述

如果一个三位数ABC，满足A^3+B^3+C^3=ABC，则称这个数为水仙花数。例如153=1^3+5^3+3^3，所以153是水仙花数

编程找出给定范围内的所有的三位水仙花数

输入

输入仅一行，包含两个数字m,n，两个数字用空格隔开，切保证m和n都是三位数，m<=n。

输出

按从小到大顺序输出所有的三位水仙花数，每个数字一行。如果给定范围内没有水仙花数，输出NO ANSWER!

```
#include <stdio.h>
int main(void){
    int m, n, i, j, a, b, c, t, found = 0;
    scanf("%d%d", &m, &n);
    for(i=m; i<=n; ++i) {
        a = i/100;
        b = (i/10)%10;
        c = i%10;
        t = a*a*a + b*b*b + c*c*c;
        if(t == i)  {
            found = 1;
            printf("%d\n", i);
        }
    }
    if(!found)
        printf("NO ANSWER!\n");
}
```

本地可以测试发现水仙花数特别少

```
#include <stdio.h>

int main(void)
{
        int total=0,m,n;

        scanf("%d%d",&m,&n);
        if (m<=153 && n>=153) {total++;printf("153\n");}
        if (m<=370 && n>=370) {total++;printf("370\n");}
        if (m<=371 && n>=371) {total++;printf("371\n");}
        if (m<=407 && n>=407) {total++;printf("407\n");}

        if (!total)
        printf("NOANSWER!\n");
}
```

# 第三小的数

题目描述

输入一组数字，输出这些数字中第三小的数。

输入

第一行一个整数n，代表数字的总数。n<2.1*10^9
接下来n行，每行一个数字

输出

输出n个数字中第三小的数

```
#include <iostream>
using namespace std;

int main()
{
	int n;
	cin >> n;
	auto arr = new int[n];
	for (int i = 0; i < n; i++) {
		cin >> arr[i];
	}
	for (int i = 0; i < 3; i++) {
		for (int j = n - 1; j > i; j--) {
			if (arr[j - 1] > arr[j]) {
				int temp = arr[j - 1];
				arr[j - 1] = arr[j];
				arr[j] = temp;
			}
		}
	}
	cout << arr[2];
	return 0;
}
```

BY 162230319

借鉴选择排序的方法，只需选择三次

```
#include<iostream>
using namespace std;
const int N=2e9+10;
int a[3]={N,N,N};
int main()    {
            int n;      cin>>n;
            while(n--)            {
                        int t;
                        cin>>t;
                        if(t<a[0])              {
                                    a[2]=a[1];a[1]=a[0];a[0]=t;
                        }
                        else if(t<a[1])        {
                                    a[2]=a[1];a[1]=t;
                        }
                        else if(t<a[2]) a[2]=t;
            }
            cout<<a[2];
            return 0;
}
```

BY 162240221

定义三个变量，存储前三小的值

题目F

```
#include"stdio.h"
int main()
{
	int n,i,a,b,c,m;
	a=b=c=2147483647;
	scanf("%d",&n);
	for(i=0;i<n;i++)
	{
		scanf("%d",&m);
		if (m<a)
		{
			c=b;b=a;a=m;
		}
		else if(m<b)
		{
			c=b;b=m;
		}
		else if(m<c) c=m;
	}
	printf("%d\n",c);
	return 0;
}
```

# 统计单词个数

题目描述

有一行字符，仅由小写字母和空格字符组成。
未被空格分隔开的，连续的若干个字母视为一个单词，每个单词至少包含1个字母。用来分隔单词的空格可能有任意多个。
请计算给出的一行字符中，总共有多少单词。

输入

一行仅由小写字母和空格组成的字符，字符个数不超过10000个。

输出

仅一行，为单词个数。

```
#include<stdio.h>
int main()
{
        char str[10000];
        int word=0,letter=0,i=0;
        gets(str);
        while(str[i])
        {
                if(letter==0&&(str[i]>='a'&&str[i]<='z'))
                        letter=1;word++;
                else if(str[i]==' ') letter=0;
                i++;
        }
        printf("%d",word);
        return 0;
 }
```

```
#include<iostream>
#include<string>
using namespace std;

int main()
{
        string a;
        int i=0;
        int  count=0;
        getline(cin,a);
        for(i=0;i<a.length();i++)
        {
                if(a[i]!=' '&&(a[i+1]==' '||a[i+1]=='\0'))
                {
                        count++;
                }
        }
        cout<<count<<endl;
        return 0;
}
```

BY 162220109