

编程题汇总第一卷

肘后备急方（你的算法题急救手册）

- 《肘后备急方》，古代汉医方剂著作。是中国第一部临床急救手册。中医治疗学专著。8卷，70篇。东晋时期葛洪著。
- 我的面试算法备急方第一卷：基础编程题 目的是那些很久没有写过c/c++ 代码的同学的一剂急药 让你敲完这68题 回忆起编程题的基础部分 输入输出 循环 等等.....（已完结）
- 面试算法备急方第二卷：中级编程题 这些题都是各个公司笔试题的前几题 是面试的入门题 主要包括字符串 数组 一般的数据结构 排序 (更新中)
- 面试算法备急方第三卷：高级编程题 高级算法题 KMP 动态规划 ACMER 之类的 （准备中）

基础差的同学以抄写代码为主，然后再去理解代码，基础好的同学可以直接做习题。

题目来源：邢文鹏 牛客网 计蒜课 刘汝佳 算法竞赛入门经典

[每天学一点数据结构和算法的博客](#)

[华为机试真题博客](#)

【应该背过的东西】：ASCII码表中：‘A’~‘Z’的值对应 65~90；‘a’~‘z’的ASCII值为 97~122；因此大小写之间差 32。91~96 分别代表：[\] ^ _ ` (~号下面)。数字 0 为 48；

1、十进制数据按照二进制形式输出

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

int main(){
    int ten;
    cin>>ten;
    int num[32];
    int i = 0;
    while(ten)
    {
        num[i]= ten % 2;
        ten = ten /2;
        i++;
    }
    for(i = i-1;i>=0;i--)
    {
        cout<<num[i];
    }

    system("pause");
}
```

2、十进制按照十六进制输出

```

#include<iostream>
using namespace std;

int main(){
    int ten;
    cin>>ten;
    int num[32];
    int i = 0;
    while(ten)
    {
        num[i]= ten % 16;
        ten = ten /16;
        i++;
    }
    for(i = i-1;i>=0;i--)
    {
        if(num[i]<10)
        {
            cout<<num[i];
        }else {
            switch(num[i])
            {
                case 10:
                    cout<<"A";
                    break;
                case 11:
                    cout<<"B";
                    break;
                case 12:
                    cout<<"C";
                    break;
                case 13:
                    cout<<"D";
                    break;
                case 14:
                    cout<<"E";
                    break;
                case 15:
                    cout<<"F";
                    break;
            }
        }
    }

    system("pause");
}

```

3、统计一个自然数的二进制表示形式中有多少个1

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

int main(){
    int num;
    cin>>num;
    int count = 0;
    while(num)
    {
        num = num & (num -1);
        count++;
    }

    cout<<count<<endl;
    system("pause");
}
```

4、编写程序，输入4个整数找出最大值和最小值。

Enter four integers: 10 8 49 35

Largest:49

Smallest:8

```
#include <stdio.h>

int main(void)
{
    int a, b, c, d, max, min;
    printf("请输入4个整数: \n");
    scanf("%d %d %d %d", &a, &b, &c, &d);

    max = (a > b) ? a : b;
    max = (max > c) ? max : c;
    max = (max > d) ? max : d;
    printf("四个数中最大值为: %d\n", max);

    min = (a < b) ? a : b;
    min = (min < c) ? min : c;
    min = (min < d) ? min : d;
    printf("最小值为: %d\n", min);

    return 0;
}
```

5、利用switch语句编写一个程序，把用数字表示的成绩转化为字母表示的等级。

Enter numerical grade:84

Letter grade:B

```
#include <stdio.h>

int main(void)
{
    int num;
    int grade;

    printf("请输入考试分数: \n");
    scanf("%d", &num);

    grade = num / 10;
    switch(grade){
        case 10:
        case 9:
            printf("成绩为A\n");
            break;
        case 8:
            printf("成绩为B\n");
            break;
        case 7:
            printf("成绩为C\n");
            break;
        case 6:
            printf("成绩为D\n");
            break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0:
            printf("成绩为F\n");
            break;
        default:
            printf("成绩必须为0~100之间的自然数\n");
    }

    return 0;
}
```

6、编写程序实现整数的加减乘除四则运算（使用switch/case结构）

```

#include <stdio.h>

int main(void)
{
    int a, b;
    char c;

    printf("Input Formula e.g. 3 + 4\n");
    scanf("%d %c %d", &a, &c, &b);

    switch(c){
        case '+':
            printf("The answer is %d\n", (a + b));
            break;
        case '-':
            printf("The answer is %d\n", (a - b));
            break;
        case '*':
            printf("The answer is %d\n", (a * b));
            break;
        case '/':
            printf("The answer is %d\n", (a / b));
            break;
        case '%':
            printf("The answer is %d\n", (a % b));
            break;
        default:
            printf("只能输入+ - * /这四种运算符\n");
    }

    return 0;
}

```

7、编写程序，确定一个数的位数：

Enter a number:374

The number 347 has 3 digits

```

#include<iostream>
using namespace std;

int main(){
    int num;
    cin>>num;
    int count = 0;
    while(num)
    {
        num = num /10;
        count++;
    }

    cout<<count<<endl;
    system("pause");
}

```

8、判断1-100的数中，共有多少个9

```

#include <stdio.h>

int main(void)
{
    int num, i = 0;
    int count = 0;

    printf("请输入1~100之间任意一个数:\n");
    scanf("%d", &num);

    for(i = 0; i <= num; i++){
        if(i % 10 == 9)
            count++;
        if((i / 10) == 9)
            count++;
    }
    #if 0
        for(i = 1, count = 0; i <= num; i++)
            for(tmp = i; tmp != 0; tmp /= 10)
                if(tmp % 10 == 9)
                    count++;
    #endif
    printf("1 ~ %d之间的数中共有%d个9\n", num, count);

    return 0;
}

```

9、编写程序，找出用户输入的一串数中的最大值，程序需要提示用户一个一个地输入数值，当用户输入0或负数时，程序显示已输入的最大非负数。

```
Enter a number:60
Enter a number:38
Enter a number:100
Enter a number:75
Enter a number:0
The largest number entered was 100.
```

```
#include <iostream>
using namespace std;
int main(void)
{
    int num = 0;
    int max =0;
    while(cin>>num)
    {
        if(num <= 0)
        {
            break;
        }
        if(num > max)
        {
            max =num;
        }
    }

    cout<<max<<endl;
    system("pause");
    return 0;
}
```

10、将1-200之间不能被3整除的数输出（要求每行输出10个数字）


```

#include <iostream>
using namespace std;
int main(void)
{
    int count =0;
    for(int i=1;i<200;i++)
    {
        if(i%3 !=0)
        {
            cout<<i<<" ";
            count++;
        }
        if(count % 10 == 0)
        {
            cout<<endl;
        }
    }
    system("pause");
    return 0;
}

```

11、打印如下半个菱形； 11b.c可以打印整个菱形。

11.c / 11a.c / 11b.c / 11d.c

```

*
***
*****
*****
*****
***
*

```

```

#include <iostream>
using namespace std;
int main(void)
{
    int num ;
    cin>>num;
    for(int i =1;i<=num;i=i+2)
    {
        for(int j=0;j<i;j++)
        {
            cout<<"*";
        }
        cout<<endl;
    }

    for(int i=num-2;i>=1;i=i-2)
    {
        for(int k=0;k<i;k++)
        {
            cout<<"*";
        }
        cout<<endl;
    }
    system("pause");
    return 0;
}

```

12、求1-100的素数(也叫质数)。

12.c

【总结】：规定：1不是素数，2是最小的素数。

```

#include <iostream>
using namespace std;
int main(void)
{
    int i,j;
    for(i=2;i<=100;i++)
    {
        for(j=2;j<=i;j++)
        {
            if(i%j == 0)
                break;

        }
        if(j>=i)
        {
            cout<<i<<" ";
        }
    }
    system("pause");
    return 0;
}

```

13、寻找一维数组中最大、小值及其坐标位置并打印输出。

```

#include <iostream>
using namespace std;
int main(void)
{
    int arr[] = {1,2,4,6,7,9,34,45,3,5};
    int max = 0,max_index =0;
    int len = sizeof(arr)/sizeof(arr[0]);

    for(int i = 0;i<len;i++)
    {
        if(max<arr[i])
        {
            max = arr[i];
            max_index = i;
        }
    }

    cout<<max<<max_index<<endl;
    system("pause");
    return 0;
}

```

14、编写程序实现以下功能，在字符串str中找出最大的字符并放在第一个位置上，并将该字符前的原字符往后顺序移动。

如str字符串内容为“chyab”,程序执行完毕，str中的内容将成为“ychab”

【总结】：首先循环字符数组找出最大字母，并记录该字母下标。另用一个循环，将最大字母以前的字符串顺序向后移动（以后的字符串，不管它）`str[i+1] = str[i]`,最后将`str[0]`赋值为最大字母即可。

```
#include <iostream>
#include <string>
using namespace std;
int main(void)
{
    string str;
    cin>>str;
    int len = str.length();
    char max=str[0] ;
    int max_index = 0;
    for(int i =0 ;i<len ;i++)
    {
        if(max < str[i])
        {
            max = str[i];
            max_index = i;
        }
    }
    cout<<str[max_index]<<max_index<<endl;
    for(int j = max_index ;j>0;j--)
    {
        str[j] = str[j-1];
    }
    str[0] = max;

    cout<<str<<endl;
    system("pause");
    return 0;
}
```

15、将二维数组转置。（行变列、列变行）

15.c / 15a.c

```
num[5][5] = {
{1,6,2,3,7},
{2,7,3,2,2},
{3,2,5,2,6},
{3,2,6,4,2},
{4,3,7,4,8}};
```

【总结】：首先，进行转换的数组必须行列数相等。使用2重循环，三杯水方法，关键是交换的时候，循环初始条件：行循环初始*i* = 0；但是列循环初始*j* = *i*+1；

可以画图理解出：交换的过程是以00 11 22 33 44为对角线交换的，这几个值不会变。

```
#include <stdio.h>

#define ROW 5
#define COL 5
int main(void)
{
    int i, j, tmp;
    int num[ROW][COL] = {
        {1, 6, 2, 3, 7},
        {2, 7, 3, 2, 2},
        {3, 2, 5, 2, 6},
        {3, 2, 6, 4, 2},
        {4, 3, 7, 4, 8}};

    for(i = 0; i < ROW ; i++){
        for(j = 0; j < COL; j++){
            printf("%2d ", num[i][j]);
        }
        printf("\n");
    }

    for(i = 0; i < ROW; i++){
        for(j = i+1; j < ROW; j++){
            tmp = num[i][j];
            num[i][j] = num[j][i];
            num[j][i] = tmp;
        }
    }
    printf("转化以后: \n");
    for(i = 0; i < ROW ; i++){
        for(j = 0; j < COL; j++){
            printf("%2d ", num[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

16、给定某年某月某日，将其转换成这一年的第几天并输出

【总结】：这个用switch也能做，不过使用二维数组更为简单。判断闰年的方法：能被400整除，或者能被4但不能被100整除的年份：if(!(year % 100)&&(year % 4) || (year % 400))

将12个月列到2维数组中，一行显示非闰年，一行显示闰年。天数sum = day再 += 各月之和。%4d 表示4位的年， %02d表示2位的月 %02d 表示2位的day

```
#include <stdio.h>

int main(void)
{
    int days_of_year[][13] = {
        {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},
        {0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}
    };

    int i, year, month, day, leapyear, sum;
    int flag;

    do{
        flag = 0;
        printf("请输入要查询的年月日，如：2010-10-10: \n");
        scanf("%d-%d-%d", &year, &month, &day);
        if((year < 0)|| (month < 0)|| (month > 12)|| (day < 0)|| (day > 31)){
            printf(" 【警告】：输入错误！请重新输入! \n");
            flag = 1;
        }
    }while(flag);

    leapyear = ((year % 4 == 0)&&(year % 100 != 0)) || (year % 400 == 0);

    for(i = 1, sum = day; i < month; i++)
        sum += days_of_year[leapyear][i];

    printf("%4d-%02d-%02d 是这一年中的第 %d 天! \n", year, month, day, sum);

    return 0;
}
```

17、编写程序检查某一个整数中是否有重复的数字，如检查2822中存在重复数字2

```

int main(void)
{
    int num;
    cin>>num;
    int a[10]={0};

    while(num)
    {
        int sub = num %10;
        a[sub]++;
        num = num /10;
    }

    for(int i =0 ;i<10;i++)
    {
        cout<<a[i]<<" ";
    }
    system("pause");
    return 0;
}

```

18、改写上题，使其可以显示出哪些数字有重复。

```

Enter a number: 939577
Repeated digit(s) : 7 9

```

【总结】：因为数字都是由0~9组成，所以可以初始化 `int num[10]={0};` 来分别代表这个十个数的位置。得到这个数，`%10`得到个位的数字，得到的数字是几，就将 `num[几]++`;

然后`/=10`，再`%10`得到的是十位数，同样，得到几就将`num[几]++`; 然后`/=10`，再`%10`.....直到`/=10` 为0，`while`循环跳出。

这时，数组里，第几位上的数，就表示数字几，出现了多少次。仍为0的表示一次也没出现过。

19、改写上题，使其打印一个列表，显示出每个数字在数中出现的次数：Enter a number:

```

Digit :      0  1  2  3  4  5  6  7  8  9
Occurrences: 1  2  2  0  4  0  0  1  0  1

```

【总结】：把上面的用两个循环分别输出。

20、统计一下某字符串中某指定字符出现的次数（字符串和字符都从键盘输入）

【总结】：定义一个字符数组，定义一个字符，循环==即可。

```

#include <stdio.h>

int main(void)
{
    char str[30], a;
    int i, count = 0;

    printf("Enter a String and a char: e.g. happy p\n");
    scanf("%s %c", str, &a);

    for(i = 0; str[i] != '\0'; i++){
        if(str[i] == a)
            count++;
    }
    printf("The '%c' appeared %d times in \"%s\"!\n", a, count, str);

    return 0;
}

```

20a、实现数组元素(字符串、数字)的反向输出。

【总结】：正着存，倒着取。

```

int main(void)
{
    string str;
    cin>>str;

    int len = str.length();

    for(int i =len-1 ;i>=0;i--)
    {
        cout<<str[i]<<" ";
    }

    system("pause");
    return 0;
}

```

21、使用数组实现对栈的操作。

【总结】：


```

#include <stdio.h>

#define N 1024

typedef char int8_t;

int8_t stack[N];
int top = 0;

void push(int8_t n)
{
    stack[top++] = n;
}

int8_t pop(void)
{
    return stack[--top];
}

int is_empty(void)
{
    return (top == 0); // 1: empty
}
#ifdef 0

int is_full(void)
{
    return (top == N); // 1: full
}

void delete(void)
{
    top = 0;
}

#endif

int main(void)
{
    char s[] = "hello aka!";
    int i = 0;

    while(s[i] != '\0'){
        push(s[i]);
        i++;
    }

    while(is_empty() != 1){
        printf("%c", pop());
    }
}

```

```

    printf("\n");

    return 0;
}

```

22、使用指针操作数组

- 1)、对int num[10]数组进行赋值
- 2)、求数组元素之和

【总结】： 1) num是一个“地址”，因此 num+10 是10个连续的地址。所以for的条件可以这样写：（p = num; p < num + 10; p++）

2) 和1) 同，不过上面是赋值，这里是取值。

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void)
{
    int num[10], *p;
    int sum = 0;

    srand(time(NULL));
    for(p = num; p < num + 10; p++){
        printf("%d ", *p = rand() % 5);
    }
    printf("\n");
    #if 0
    int i;
    for(p = num, i = 0; p + i < num + 10; i++){
        sum += *(p + i);
        printf(" sum = %d\n", sum);
    }
    #endif
    for(p = num; p < num + 10; p++){
        sum += *p;
        printf("sum = %d\n", sum);
    }
    return 0;
}

```

23、定义char str[2][20]，从键盘输入两个字符串保存在此数组中，求这两个字符串中的最大字符

23.c

【总结】： 定义一个char max = str[0][0]; 使用2重循环，第一重 i < 2; 第二重 s t r [i] [j] != '\0'; if (max < str[i][j]) {max = str[i][j]}

```

#include <stdio.h>

int main(void)
{
    char str[2][20];
    int i, j;

    printf("Enter 2 String:\n");
    scanf("%s %s", str[0], str[1]);

    char max = str[0][0];
    for(i = 0; i < 2; i++)
        for(j = 0; str[i][j] != '\0'; j++)
            if(max < str[i][j])
                max = str[i][j];

    printf("The biggest char is : %c\n", max);
    return 0;
}

```

24、使用命令行参数接收运算符和运算值，实现四则运算。

【总结】：int main(int argc, char *argv[]) :argc 表示 输入的总参数个数，argv[i]表示输入的参数内容。如：编译通过后输入：./24a a + b

这里：*argv[0] = ./24a（因为argv[0]是 char* 型指针）； *argv[1] = a; *argv[2] = '+'; *argv[3] = b;

另，switch()的条件为 *argv[2]，而 atoi()函数 参数则输入 arg[],因为该参数的类型是 char* 的，所以和 switch 不同；

```

#include <stdio.h>

int main(void)
{
    char operator;
    int a, b;
    printf("请输入一个运算符: \n");
    scanf("%c", &operator);

    printf("请输入参加运算的两个数: \n");
    scanf("%d %d", &a, &b);

    switch(operator){
        case '+':
            printf("运算结果为: %d\n", a+b);
            break;
        case '-':
            printf("运算结果为: %d\n", a-b);
            break;
        case '*':
            printf("运算结果为: %d\n", a*b);
            break;
        case '/':
            printf("运算结果为: %d\n", a/b);
            break;
    }
    return 0;
}

```

25、实现itoa: char *myitoa(int n, char *p)

【总结】：重点是如果出现负数的情况。首先判断若这个数是负数，要 $n = -n$ ；将 $p[i++] = '-'$ ；把首地址存成负号。然后 % 10 得到各个位上的数存入数组，最后存如 $'\0'$ ，

拿 -7926483 为例，在内存中，存放的顺序依次为： $'-' | '3' | '8' | '4' | '6' | '2' | '9' | '7' | '\0'$ 所以，输出时，下标 0 和 * (最后一位) 不用动，

重要一次将 3~7、8~9、2~4 调换位置即可。使用 $\text{for}(--i; j \leq i; i--, j++)$ 这里 i 已经在之前存每个字符的时候增到 8，j 根据输入数是否为负数来进行判断，

如果为负数那从 1 开始计，如果不为负数从 0 开始计。 $\text{tmp} = p[j]; p[j] = p[i]; p[i] = \text{tmp};$

```

#include <stdio.h>
#include <time.h>
#include <stdlib.h>

char *myitoa(int n, char *p)
{
    char temp;
    int i = 0, j = 0;

    if(n < 0){ //如果为负数，那么首地址应该存负号
        n = -n;
        p[i++] = '-';
        j = 1;
    }
    do{
        p[i++] = n % 10 + '0';
    }while(n /= 10);
    p[i] = '\0';

    for(--i; j <= i; i--,j++){
        temp = p[j];
        p[j] = p[i];
        p[i] = temp;
    }

    return p;
}

int main(void)
{
    int num;
    char str[10];

    printf("Enter a number:\n");
    scanf("%d", &num);
    printf("output String : %s\n", myitoa(num, str));

    return 0;
}

```

26、用循环结构将一个字符数组里的值复制到另外一个字符数组里(main函数里实现)。

[26.c](#) / [26a.c](#) / [26_1.c](#) / [26_2.c](#)

[/26_2a.c](#) / [26_3.c](#)

- 1)、练习使用strcpy函数
- 2)、将上题设计思想整理成my_strcpy函数，并在main函数里调用、测试。
- 3)、部分复制，例如从第3个字符开始复制

【总结】：1)该函数的返回值是 `char*` 类型的，因此`dest`代表一个数组，对应`%s`，而使用在`scanf`中的时候不需要使用`&`符!!!

2)`for(i = 0; (dest[i] = str[i]); i++)`这里学习`for`循环的第二个参数的用法，另外：`\`写在`printf`中可以将`""`打出在控制台上。`\`起转义的作用

3)`dest`数组的起始字符应该是，要截取的字符串的第`N`个字符，循环截取，结束后要在末尾添加`'\0'`

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    char dest[50];
    int i;
    for(i = 0; (dest[i] = argv[1][i]); i++)
        ;
    printf("%s\n", dest);
    return 0;
}
```

28、编写一个函数`int is_leap_year(int year)`判断参数`year`是不是闰年，如果`year`是闰年则返回1，否则返回0.

如果某年份能被4整除，但不能被100整除，那么这一年是闰年，此外，能被400整除的年份也是闰年。

```
#include <stdio.h>
int is_leap_year(int year){

    if(((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0))
        return 1;
    else
        return 0;
}
int main(void)
{
    int year;
    printf("Enter a year:\n");
    scanf("%d", &year);

    int tmp = is_leap_year(year);
    if(tmp == 1)
        printf("The year \"%d\" is a leapYear!\n", year);
    else
        printf("The year \"%d\" is Not a leapYear!\n", year);

    return 0;
}
```

29、编写一个函数double myround(double x)，输入一个小数，将它四舍五入。

例如myround(-3.51)的值是-4.0，myround(4.49)的值是4.0。可以调用math.h中的库函数ceil和floor实现这个函数。

【总结】：重点： 光使用 `#include <math.h>` 是不行的，在编译的时候必须加 `gcc -lm` 这个命令才可以。

否则无法正确引入这两个函数!!!

`%f`代表“浮点型”，包括double float longdouble longfloat 都用这个表示。

`ceil("天花板")`函数，取大于等于参数 `x` 的最小整数值，结果以double类型返回。 `ceil(3.1415)`
--> 4.000000

`floor("地板")`函数，取小于等于参数 `x` 的最小整数值，结果以double类型返回。 `floor(3.1415)`
--> 3.000000

```
#include <iostream>
using namespace std;
int main(void)
{
    double num ;
    while(cin>>num)
    {
        cout<<int(num+0.5)<<endl;
    }
    system("pause");
    return 0;
}
```

30、求解1+2+...+n的和

【总结】：`i = 1; i <= n; i++`

```

#include <stdio.h>
#include <iostream>
long long sum(long long n)
{
    if(n == 0)
        return 0;
    else
        return (n + sum(n - 1));
}

int main(void)
{
    long long num;

    printf("Enter a num: \n");
    scanf("%lld", &num);

    printf("1+2+3+...+%lld = %lld\n", num, sum(num));
    printf("sum = (%lld+1)*%lld/2 = %lld\n", num, num, (num + 1) * num / 2);

    system("pause");
    return 0;
}

```

31、输入正方形的个数和正方形边的长度 求总面积

```
scanf("%d",&n),n
```

就是当输入一个整形,且这个整形 的数不是0时,条件成立,否则,若输入0,则跳过while里面的内容


```

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

int main()
{
    int n;
    while(scanf("%d",&n),n);
    {
        int x,sum=0;
        for (int i = 0; i < n; ++i)
        {
            scanf("%d",&x);
            sum = sum + x * x;
        }
        printf("%d\n",sum);
    }

    return 0;
}

```

32、输入要罚抄的课文，和抄写遍数，来抄写课文

```

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

int main()
{
    char str[201];
    int cnt;
    gets(str);
    while(scanf("%d",&cnt)==1)
    {
        while(cnt--)
        {
            puts(str);
        }
    }

    return 0;
}

```

33、求连续数的立方和 33.cpp 33a.cpp

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

int main()
{
    int cnt;
    int sum = 0;
    while(scanf("%d",&cnt),cnt)
    {
        sum += cnt*cnt*cnt;
    }
    printf("%d\n", sum);

    return 0;
}
```

34、题目描述:

NowCoder小时候走路喜欢蹦蹦跳跳，他最喜欢在楼梯上跳来跳去。

但年幼的他一次只能走上一阶或者一下子蹦上两阶。

现在一共有N阶台阶，请你计算一下NowCoder从第0阶到第N阶共有几种走法。[提示：利用斐波那契数组]34.cpp

输入描述:

输入包括多组数据。每组数据包括一个整数n, ($1 \leq n \leq 90$)。

输出描述:

对应每个输入包括一个输出。

为redraiment到达第n阶不同走法的数量。

输入例子:

1
2

输出例子:

1
2

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

int allkindstep(int num)
{
    if(num ==1)
    {
        return 1;
    }else if(num ==2)
    {
        return 2;
    }

    else return allkindstep(num-1)+allkindstep(num-2);
}

int main()
{
    int num;
    while(cin>>num)
    {
        cout<<allkindstep(num)<<endl;
    }
}
```

35、二分查找法

```

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

int binary_search(int len,int num[],int request)
{
    int left = 0, right = len-1,mid;
    while(left <= right)
    {
        mid = (left + right)/2;
        if (num[mid] == request)
        {
            return mid;
        }
        else if (request < num[mid])
        {
            right = mid;
        }else if(request > num[mid])
        {
            left = mid +1;
        }
    }
    return -1;
}

int main()
{
    int const LEN = 10000;
    int a[LEN];
    for (int i = 0; i < LEN; ++i)
    {
        a[i] = i - 5000;
    }
    int goal = 0 ;
    int index = binary_search(LEN,a,goal);
    if (index == -1)
    {
        cout<<goal<<"is"<<"error"<<endl;
    }else{
        cout<<goal<<"is"<<index<<endl;
    }
}

```

36、一只成熟的兔子每天能产下一胎兔子。每只小兔子的成熟期是一天。某人领养了一只小兔子，请问第N天以后，他将会得到多少只兔子。

```

#include <iostream>
using namespace std;

int allkindstep(int num)
{
    if(num ==1)
    {
        return 1;
    }else if(num ==2)
    {
        return 2;
    }

    else return allkindstep(num-1)+allkindstep(num-2);
}

int main()
{
    int num;
    while(cin>>num)
    {
        cout<<allkindstep(num)<<endl;

    }
}

```

37、蒜头的数学实在是太差了，于是老师把他关到小黑屋让他闭门修炼。老师跟他一张纸，上面一排写着 1, 2, 3...N 这 N 个数，中间用空白分隔。老师让他在空白处填上加号或者减号。他让蒜头君求出一共有多少种加运算符的方法使得整个表达式的值为 0，并输出所有的方案。比如 N=7 时，1 2 3 4 5 6 7 排成一排，一种插入符号的方案为 1+2-3+4-5-6+7=0。是不是很有趣，快来帮蒜头君解出这题吧(*'▽')//

输入为一行，包含一个整数 N（ $3 \leq N \leq 9$ ）。

输出为所有在每对数字间插入“+”或“-”后能得到和为零的数列，并按照字典（ASCII 码）序排列。如果无解就输出一行 None。

不知道字典序和 ASCII 也不要紧，我们看样例输出就清楚啦，1 到 N 排成一排，先每个位置优先放“+”，再放“-”，这么放的原因是因为“+”的 ASCII 码要比“-”小。

```

#include<stdio>
#include<cstring>
#include<stdlib>
#include<cmath>
#include<algorithm>
using namespace std;
typedef long long LL;
const int MX=1009+5;

int ans[MX],n;

void DFS(int pos,int sum,int d){
    sum+=d?pos:-pos;
    ans[pos]=d;

    if(pos==n){
        if(!sum){
            printf("1");
            for(int i=2;i<=n;i++){
                printf("%c%d",ans[i]?'+':'-',i);
            }
            printf("\n");
        }
        return;
    }

    DFS(pos+1,sum,1);
    DFS(pos+1,sum,0);
}

int main(){
    while(~scanf("%d",&n)){
        DFS(1,0,1);
    }
    return 0;
}

```

38、读入n名学生的姓名、学号、成绩，分别输出成绩最高和成绩最低学生的姓名和学号。

```

#include <stdio.h>
#include <iostream>
int main()
{
    int n=0;
    int score =0;
    int score_hight = 0;
    int score_low = 101;
    char name [10]={0};
    char name_low [10]={0};
    char name_hight [10]={0};
    char id [10]={0};
    char id_low [10]={0};
    char id_hight [10]={0};
    scanf("%d",&n);
    for(int i=0;i<n;++i)
    {
        scanf("%s%s%d",name,id,&score);
        if(score > score_hight )
        {
            score_hight = score ;
            sprintf(name_hight,"%s",name);
            sprintf(id_hight,"%s",id);
        }
        if(score < score_low )
        {
            score_low = score;
            sprintf(name_low,"%s",name);
            sprintf(id_low,"%s",id);
        }

    }

    printf("%s %s\n",name_hight,id_hight);
    printf("%s %s\n",name_low,id_low);

    return 0;
}

```

39、输入一段数字，逆向输出

```

#include <stdio.h>
#include <iostream>
int main()
{
    int a[100],n = 0;
    int num = 0;
    while(scanf("%d",&num),num)
    {
        a[n++] = num;
    }

    for (int i = n - 1; i >= 0; i--)
    {
        printf("%d\n",a[i]);
    }

    return 0;
}

```

40、有N台灯，第一个人开了所有灯，第二个人关了2的倍数的灯，第三个人开了3的倍数的灯。这样下去求房间里最后还开着的灯。

```

#include <stdio.h>
#include <string.h>
#define MAXN 1000+10
int a[MAXN];
int main()
{
    int i, j, n, k, first = 1;
    memset(a, 0, sizeof (a));
    scanf("%d%d",&n, &k);
    for (i=1; i<=k; i++)
    for (j=1; j<=n; j++)
    if (j%i==0) a[j]=!a[j];
    for (i=1; i<=n; i++)
    if (a[i]) {
        if (first) first = 0;
        else printf(" ");
        printf("%d", i);
    }
    printf("%\n");
    return 0;
}

```

41、蛇形填数

输入 4

输出:

10 11 12 1

9 16 13 2

8 15 14 3

7 6 5 4

```
#include <stdio.h>
#include <iostream>
int main()
{
    int a[10][10];
    int n = 0, x = 0, y = 0, tot = 0;
    scanf("%d", &n);
    memset(a, 0, sizeof(a));
    tot = a[x=0][y=n-1]=1;
    while(tot<n*n)
    {
        while(x+1<n&& a[x+1][y]==0) a[++x][y]=++tot;
        while(y-1>=0&& a[x][y-1]==0) a[x][--y]=++tot;
        while(x-1>=0&& a[x-1][y]==0) a[--x][y]=++tot;
        while(y+1<n&& a[x][y+1]==0) a[x][++y]=++tot;
    }
    for (x = 0; x < n; x++)
    {
        for(y=0; y<n; y++)
        {
            printf("%3d", a[x][y]);
        }
        printf("\n");
    }

    return 0;
}
```

42、利用递归求阶层

```

#include <stdio.h>
#include <iostream>

    int f(int n)
    {

        return n==0?1:f(n-1)*n;
    }
int main()
{

    int n,m;
    scanf("%d",&n);
    m=f(n);
    printf("%d",m);

    return 0;
}

```

43、输出不能被3整除的数字

```

#include <stdio.h>
#include <iostream>
using namespace std;

int main()
{
    int i;
    for (i = 1; i <= 20; ++i)
    {
        if (i % 3 ==0)
            continue;
        cout<<i<<" ";
    }
    cout<<endl;
}

```

44、对数组先进行选择排序，然后进行二分查找

```

#include <stdio.h>
#include <iostream>
using namespace std;

int main()
{
    int a[]={1,5,33,53,23,19,34,56,78,22,234};
    int length;
    int t;
    length = sizeof(a)/sizeof(a[0]);

    for (int i = 0; i < length; ++i)
    {
        for (int j = i+1; j < length; ++j)
        {
            if (a[i]>a[j])
            {
                t = a[i];
                a[i] = a[j];
                a[j] = t;
            }
        }
    }

    for (int i = 0; i < length; ++i)
    {
        printf("%d\n", a[i]);
    }

    int low = 0, hight = length - 1;
    int aim;
    scanf("%d",&aim);
    while(low <= hight)
    {
        int mid = (low +hight)/2;

        if (a[mid] == aim)
        {
            cout<<mid<<endl;
            break;
        }else if (a[mid] > aim)
        {
            hight = mid - 1;
        }else
        {
            low = mid + 1;
        }
    }
}

```

45、递归求解汉诺塔问题

```
#include <stdio.h>
#include <iostream>
using namespace std;
void func(int n,string from,string mid,string to)
{
    if (n == 1)
    {
        cout<<"move from "<<from<<" to "<<to<<endl;
    }
    else{
        func(n-1,from,to,mid);
        func(1,from,mid,to);
        func(n - 1,mid,from,to);
    }
}
int main()
{
    int n;
    scanf("%d",&n);
    func(n,"left","mid","right");
}
```

46、卡片游戏

桌上有一叠牌，从第一张牌（即位于顶面的牌）开始从上往下依次编号为1~n。当至少还剩两张牌时进行以下操作：把第一张牌扔掉，然后把新的第一张放到整叠牌的最后。输入n，输出每次扔掉的牌，以及最后剩下的牌。

样例输入：7

样例输出：1 3 5 7 4 2 6

```

#include <iostream>
#include <queue>
using namespace std;
int main()
{
    queue<int> q;
    int n;
    cin>>n;
    for(int i=0;i<n;i++)
        q.push(i+1);
    while(!q.empty()){
        cout<<q.front()<<" ";
        q.pop();
        q.push(q.front());
        q.pop();
    }
    cout<<endl;
    return 0;
}

```

47、6174问题

任取一个四位数,只要四个数字不全相同,按数字递减顺序排列,构成最大数作为被减数;按数字递增顺序排列,构成最小数作为减数,其差就会得6174; 如不是6174,则按上述方法再作减法,至多不过7步就必然得到6174.

```

#include <stdio.h>
#include <string.h>
int next_num(int x)
{
    char s[10];
    int a,b,n;
    sprintf(s,"%d",x);
    n=strlen(s);
    for (int i=0; i<n; i++) {
        for (int j=i; j<n; j++) {
            if (s[i]>s[j]) {
                char t=s[i];
                s[i]=s[j];
                s[j]=t;
            }
        }
    }
    sscanf(s, "%d",&a);
    //printf("%d",a);
    for(int i=0;i<n/2;i++)
    {
        char t = s[i];
        s[i]= s[n-1-i];
        s[n-1-i]= t;
    }
    sscanf(s, "%d",&b);
    // printf("%d",b);

    return b-a;

    printf("%d",b-a);
}

```

```

int main()
{
    int x;
    scanf("%d",&x);
    printf("%d",x);
    while(x != 6174)
    {
        printf("-->");
        printf("%d",next_num(x));
        x=next_num(x);
    }
    printf("\n");
}

```

//sprintf(s,"%d",x); 的目的数将数字x转换成字符串s

//sscanf(s, "%d",&b); 目的是将字符s转换成数字b

```
}
```

48、很多学生学习加法时，进位很容易出错。你的任务是计算两个整数在相加的时候需要进位多少次。

样例输入：

123 456

555 555

123 594

0 0

样例输出：

0

3

1

```
#include <iostream>
using namespace std;
int main(){
    int a;int b;

    while(cin>>a&&cin>>b)
    {
        if(a == 0&&b==0)
        {
            break;
        }
        int ret = 0;
        int c =0;
        while(a&&b)
        {
            if(a%10+b%10+c>9)
            {
                ret++;
                c=1;
            }
            a=a/10;
            b=b/10;
        }

        cout<<ret<<endl;
    }
}
```

49、周期串

如果一个字符串可以由某个长度为K的字符串重复多次得到，我们说该串以K为周期。例如，`abcabcabc`以3位周期（注意，也可以6和12为周期）。输入一个长度不超过80的串，输出它的最小周期。

样例输入: `HoH0H0`

样例输出: `2`

```
#include<iostream>
#include<string>
using namespace std;
int main(){

    string str;
    cin>>str;
    int len = str.length();

    for(int i=1;i<=len;i++){
        if(len%i==0)
        {
            int flag = 0;
            for(int j=i;j<len;j++){
                if(str[j] != str[j%i])
                {
                    flag =1;
                    break;
                }

                if(flag == 0)
                {
                    cout<<i<<endl;
                    break;
                }
            }
        }

    }

    system("pause");
}
```

50、Tex括号

在Tex中，左右双引号是“和”，输入一篇包含双引号的文章，你的任务是把它转换成Tex的格式。

样例输入: `"To be or not to be,"quoth the Bard,"that is the question".`

样例输出: `"To be or not to be,"quoth the Bard,"that is the question".`


```

int main(){

    string str;
    cin>>str;
    int len = str.length();

    for(int i=0;i<len;i++)
    {
        if(str[i] == '"')
        {
            str[i] = '\'';
        }
        if(str[i] == '\'')
        {
            str[i] = '"';
        }

    }

    cout<<str;

    system("pause");

}

```

51.有1、2、3、4个数字，能组成多少个互不相同且无重复数字的三位数？都是多少？

1.程序分析：可填在百位、十位、个位的数字都是1、2、3、4。组成所有的排列后,再去掉不满足条件的排列。

```

#include<iostream>
//#include<string>
//#include<string.h>
using namespace std;

int main()
{
    int arr[100];
    for(int i =0;i<=63;)
    {
        for(int j =1;j<=4;j++)
        {
            for(int k =1;k<=4;k++)
            {
                for(int l =1;l<=4;l++)
                {
                    arr[i++]=j*100+k*10+l;

                }
            }
        }
    }
    for(int i =0;i<=63;i++)
    {
        int ge = arr[i]%10;
        int bai = arr[i]/100;
        int shi = (arr[i] - 100*bai)/10;
        if(ge!=bai && ge != shi && shi!=bai)
        {
            cout<<arr[i]<<endl;
        }
    }

    system("pause");
}

```

52.输入一个字符串，删除其中所有的数字，所有大写字母改成小写，其他不变，并输出

输入描述

一个字符串，保证没有空格，以回车符结束，字符串长度 ≤ 20

输出描述

一个字符串，为题目要求的结果

输入样例

aAbB13A

输出样例

aabba

解题思路：模拟题目要求即可，遇到数字就跳过不输出，遇到大写字母就改成小写。

```
#include<iostream>
#include<string>
//#include<string.h>
using namespace std;

int main()
{
    string s;
    cin>>s;
    for (int i=0;i<s.size();i++) //枚举字符串的每一位
    {
        if (s[i]>='0' && s[i]<='9') continue; //遇到数字就跳过
        if (s[i]>='A' && s[i]<='Z') s[i]=s[i]-'A'+'a'; //遇到大写字母就改成小写
        cout<<s[i];
    }
    cout<<endl;

    system("pause");
}
```

53.输入一个字符串，统计其出现频率最高的字符，并输出。若存在两个字符出现频率相同，则输出字典序较小的那一个

输入描述

一个字符串，保证没有空格，以回车符结束，字符串长度 ≤ 20

输出描述

一个字符

输入样例

aabbaabb

输出样例

a

解题思路：做一个频率数组来统计所有字符的出现频率，机试时候不会有汉字输入，因此只考虑输入是ASCII编码的情况。

```

#include<iostream>
#include<string>
//#include<string.h>
using namespace std;

int main()
{
    string s;
    int biggest=0;
    int ascall[200];
    cin>>s;
    memset(ascall,0,sizeof(ascall));
    for (int i=0;i<s.size();i++) //枚举字符串的每一位
    {
        int num = s[i];
        ascall[num]++;
        if(ascall[num]>biggest)
        {
            biggest = ascall[num];
        }
    }

    for(int i =0;i<127;i++)
    {
        if(biggest == ascall[i])
        {
            cout<<char(i);
            break;
        }
    }

    system("pause");
}

```

54.输入10个数字，按各个位上的和从小到大排序，如果相同，则按数字从小到大排序。

输入描述

10个正整数，保证都在int范围内，用空格隔开

输出描述

10个数字，其从大到小的值，用空格隔开，最后一个数字后不加空格

输入样例

11 3 2 4 5 9 8 7 10 6

输出样例

10 2 11 3 4 5 6 7 8 9

解题思路：调用C++自带的sort函数，重新改写compare函数即可。

```

#include<iostream>
#include<algorithm>
using namespace std;
int cal(int x)
{
    int ans = 0;
    while(x!=0)
    {
        ans = ans +x%10;
        x = x/10;
    }
    return ans;
}
int num[10];
int camp(int x,int y)
{
    if(cal(x)!=cal(y))
    {
        return cal(x)<cal(y);
    }else
    {
        return x<y;
    }
}
int main()
{
    for(int i=0;i<10;i++)
    {
        cin>>num[i];
    }
    sort(num,num+10,camp);
    //sort(num,num+10);
    for(int i=0;i<10;i++)
    {
        cout<<num[i]<<" ";
    }

    system("pause");
    return 0;
}

```

55.你有一个容量为100的箩筐，给你30个物品，每个物品的体积已知问：最多能装多少个物品进箩筐

输入描述

一行30个正整数，用空格隔开，表示每个物品的体积

输出描述

一个数字，为最多能装下的物品数

输入样例（此处用3个物品作为样例，实际读入为30个）

5 59 100

输出样例

2

解题思路：利用性价比对所有物品进行排序，优先装性价比高的，在此题中，性价比就是物品的体积

```
#include<iostream>
#include<algorithm>
using namespace std;
int main()
{
    int num[5] = {0};
    int i = 0;
    for(i = 0 ;i<5;i++)
    {

        cin>>num[i];

    }
    sort(num,num+5);
    int sum =0,count =0;
    for(int i=0;i<5;i++)
    {
        if(sum + num[i]<=100)
        {
            sum = sum + num[i];
            count++;
        }
    }
    cout<<count;
    system("pause");
    return 0;
}
```

56.10个学生考完期末考试评卷完成后，A老师需要划出及格线，要求如下：

- (1) 及格线是10的倍数;
- (2) 保证至少有60%的学生及格;
- (3) 如果所有的学生都高于60分, 则及格线为60分
- (4) 及格线越高越好, 但最高不能超过60

输入: 输入10个整数, 取值0~100

输出: 输出及格线, 10的倍数

输入样例: 61 51 49 30 20 10 70 80 90 99

输出样例: 50

解题思路: 从高到低枚举及格线, 输出第一个满足要求的及格线就是答案

```
#include<iostream>
#include<algorithm>
using namespace std;
int main()
{
    //int num[10] = {61,51,49,30,20,10,70,80,90,99};
    int num[10];
    int i = 0;
    for(i = 0 ; i<10; i++)
    {
        cin>>num[i];
    }
    sort(num,num+10);
    for(i = 0 ; i<10; i++){
        cout<<num[i]<<" ";
    }
    int ret = 0;
    cout<<endl;
    if(num[4] >=60)
        ret = 60;

    for(int start =60; start >=10; start=start-10)
    {
        if(start <=num[4])
        {
            ret = start;
            break;
        }
    }

    cout<<ret;
    system("pause");
    return 0;
}
```

57.输出 9*9 口诀

```

#include <stdio.h>
#include <string.h>
int main()
{
    int i,j,k;
    for (i = 1; i <= 9; ++i)
    {
        for (j = 1; j <= i; ++j)
        {
            k = i * j;
            printf("%d * %d = %d ",i,j,k );
        }
        printf("\n");
    }
}

```

58、古典问题：有一对兔子，从出生后第3个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一对兔子，假如兔子都不死，问每个月的兔子总数为多少？

程序分析： 兔子的规律为数列1,1,2,3,5,8,13,21....

```

#include <stdio.h>
#include <string.h>

int main()
{
    long long a[30];
    long long sum = 0;
    a[0]=1,a[1]=1;
    for(int i=2;i<=30;i++)
    {
        a[i]=a[i-1]+a[i-2];
    }

    for(int j=0;j<=12;j++)
    {
        sum+=a[j];
        printf("the %d month is %lld\n",j+1 ,sum);
    }
}

```

59、判断101-200之间有多少个素数，并输出所有素数。

判断素数的方法：用一个数分别去除2到sqrt(这个数)，如果能被整除，则表明此数不是素数，反之是素数。


```

#include <stdio.h>
#include <string.h>

int main()
{
    for (int i = 101; i < 200; ++i)
    {
        for (int j = 2; j < i; ++j)
        {
            if (i % j == 0)
            {
                break;
            }
            else printf("%d ", i);
            break;
        }
    }
}

```

60、将一个正整数分解质因数。例如：输入90,打印出90=233*5。

程序分析：对n进行分解质因数，应先找到一个最小的质数k，然后按下述步骤完成：(1)如果这个质数恰等于n，则说明分解质因数的过程已经结束，打印出即可。(2)如果 $n \neq k$ ，但n能被k整除，则应打印出k的值，并用n除以k的商,作为新的正整数你n,重复执行第一步。(3)如果n不能被k整除，则用k+1作为k的值,重复执行第一步。

```

#include<iostream>
using namespace std;
int main()
{
    int num;
    cin>>num;

    for(int i=2;i<=num;i++)
    {
        if(num % i ==0)
        {
            cout<<i<<"*";
            num = num /i;
        }

    }

    cout<<num;
    system("pause");
    return 0;
}

```

61、题目：输入两个正整数m和n，求其最大公约数和最小公倍数。

```
int main()
{
    int num1,num2;
    scanf("%d%d",&num1,&num2);
    int x,y,temp;
    if (num1 < num2)
    {
        temp = num1;
        num1 = num2;
        num2 = temp;
    }
    x = num1;
    y = num2;
    while(y != 0)
    {
        temp = x % y ;
        x = y;
        y = temp;
    }
    printf("max is %d ",x);
    printf("min is %d\n", num1 * num2 / x);
}
```

程序分析：利用辗除法。

```

#include<iostream>
using namespace std;
int main()
{
    int a,b;
    cin>>a>>b;

    int max = a>b?a:b;
    int min = a<b?a:b;

    while(min !=0)
    {
        int temp = max %min;
        max = min;
        min = temp;
    }
    cout<<max<<" ";
    cout<<a*b/max;
    system("pause");
    return 0;
}

```

62、求 $s=a+aa+aaa+aaaa+aa\dots a$ 的值，其中 a 是一个数字。例如 $2+22+222+2222+22222$ (此时共有5个数相加)，几个数相加有键盘控制

```

#include<iostream>
#include<math.h>
using namespace std;
int sum(int num,int time)
{
    int sum =0;
    for(int i=0;i<time;i++)
    {
        sum = sum +num;
        num = num*10+num;
    }
    return sum;
}
int main()
{
    int num;
    cin>>num;
    int time;
    while(cin>>time)
    {
        cout<<sum(num,time);
    }
    system("pause");
    return 0;
}

```

63、题目：一个数如果恰好等于它的因子之和，这个数就称为“完数”。例如 $6=1+2+3$ 。编程找出1000以内的所有完数。

```

#include<iostream>
using namespace std;

int main()
{
    for(int i=2;i<1000;i++)
    {
        int s=i;
        for(int j=1;j<i;j++)
        {
            if(i%j == 0)
            {
                s = s-j;
            }
        }

        if(s == 0)
        {
            cout<<i<<" ";
        }
    }
    system("pause");
    return 0;
}

```

64.随机输入一个数，判断它是不是对称数（回文数）（如3，121，12321，45254）。不能用字符串库函数，判断一个字符串是不是回文串

```

#include<iostream>
using namespace std;

int main()
{
    int num;
    cin>>num;
    int s = 0;
    s =num;
    int n_num = 0;
    while(num)
    {
        n_num = n_num*10+num %10;
        num = num/10;
    }

    if(n_num == s)
    {cout<<"yes";
    }
    else
    {cout<<"no";
    }
    system("pause");
    return 0;
}

```

65.写一个程序, 要求功能: 求出用1, 2, 5这三个数不同个数组合的和为100的组合个数。

事实上, 这个题目是一道明显的数学问题, 而不是单纯的编程问题。我的解法如下:

因为 $x+2y+5z=100$

所以 $x+2y=100-5z$, 且 $z \leq 20$ $x \leq 100$ $y \leq 50$

所以 $(x+2y) \leq 100$, 且 $(x+5z)$ 是偶数

```

#include<iostream>
using namespace std;

int main()
{

    int x,y,z;
    for(int i=1;i<=100;i++)
    {
        for(int j=1;j<=50;j++)
        {
            for(int k=1;k<=20;k++)
            {
                x=i;
                y=j;
                z=k;
                if(x+2*y<=100&&(x+5*z)%2 ==0&&x+2*y+5*z==100)
                {
                    cout<<x<<" "<<y<<" "<<z<<" "<<endl;
                }
            }
        }
    }
    system("pause");
}

```

66.输入一句英文句子，翻转句子中单词的顺序，但单词内字符的顺序不变，句子中单词以空格符隔开。

为简单起见，标点符号和普通字母一样处理。

例如：输入“I am a boy.”，则输出“boy. a am I”。

```

#include <iostream>
using namespace std;

    void swap(char *begin,char *end)
{

    char temp = *begin;
    *begin = *end;
    *end = temp;

}

int main()
{
//    char mychar[5000];
//    gets(mychar);

char mychar[]="i am a boy";

char *begin = mychar;
char *end = mychar;
char *ptr = mychar;

while (*ptr++ != '\0') {

    if (*ptr == ' ' || *ptr == '\0') {
        end = ptr-1;
        while(begin < end)
            swap(*begin++, *end--);
        end = begin = ptr+1;
    }

}

begin = mychar,end = ptr-2;

while (begin <end) {
    swap(*begin++, *end--);
}

cout<<mychar;

system("pause");

}

```


67.给定一个字符串，把字符串内的字母转换成该字母的下一个字母，a换成b，z换成a，Z换成A，如aBf转换成bCg，字符串内的其他字符不改变，给定函数，编写函数void Stringchang (const charpu,charoutput) 其中input是输入字符串，output是输出字符串

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

int main()
{

    string str;
    cin>>str;
    int len = str.length();

    for(int i=0;i<len;i++)
    {
        if(str[i]=='z')
            { str[i]='a';
            }
        else if(str[i]=='Z')
            { str[i]='A';}else
            {
                str[i] = str[i]- 'A'+ 'B';
                cout<<str[i];
            }
    }

    system("pause");
}
```

68.求两个字符串的乘积，结果存到字符串中，例如字符串一中存的“657891”，字符串二中存的“521”，分别将字符串中的字符转换成整型数字，进行计算后，再转换成字符类型存储起来
函数为 void mul (char *input1, int n, char *input2, int m, char *output)
其中input1和input2是输入，n是input1的长度，n2是input2的长度。Output是输出

```

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

int main()
{
    string str1;
    cin>>str1;
    string str2;
    cin>>str2;
    int n = str1.length();
    int m = str2.length();

    int num1=0,num2=0;
    int count1 = 0;
    int count2 = 0;
    for(int i=n-1;i>=0;i--,count1++)
    {
        num1 = num1+(str1[i]-'0')*pow(10.0,count1);
    }

    for(int j=m-1;j>=0;j--,count2++)
    {
        num2 = num2+(str1[j]-'0')*pow(10.0,count2);
    }

    long long tar = num1*num2;
    cout<<tar;

    char output[100];

    int k=0;
    while(tar)
    {
        output[k]=tar%10+'0';
        k++;
        tar = tar/10;
    }
    output[k] = '\0';

    for(int l=0;l<k/2;l++)
    {
        char temp = output[l];
        output[l] = output[k-1-l];
        output[k-1-l]= temp;
    }

    cout<<output;
    system("pause");
}

```

