

# 上机考试练习题

20021程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat (0<repeat<10)，做repeat次下列运算：

输入 x，计算并输出下列分段函数 f(x) 的值(保留1位小数)。

当 x 不等于0时， $y = f(x) = 1/x$ ，当 x 等于0时， $y = f(x) = 0$ 。

输入输出示例：括号内是说明

输入

2 (repeat=2)

10 (x=10)

0 (x=0)

输出

f(10.00) = 0.1

f(0.00) = 0.0

```
#include <stdio.h>
int main(void)
{
    int repeat, ri;
    double x, y;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%lf", &x);
        /*-----*/
        if(x!=0) y=1/x;
        else y=0;
        printf("f(%.2f) = %.1f\n", x, y);
    }
}
```

20022程序填空，不要改变与输入输出有关的语句。

输入华氏温度，输出对应的摄氏温度。计算公式： $c = 5*(f-32)/9$ ，式中：c表示摄氏温度，f表示华氏温度。

输入输出示例：括号内为说明

输入

150 (fahr=150)

输出

celsius = 65

```
#include <stdio.h>
int main(void)
```

```

{
    int celsius, fahr;
    /*-----*/

scanf("%d",&fahr);

celsius=5.0*(fahr-32)/9;

    printf("celsius = %d\n", celsius);
}

```

20023程序填空，不要改变与输入输出有关的语句。

输入存款金额 money、存期 year 和年利率 rate，根据下列公式计算存款到期时的利息 interest(税前)，输出时保留2位小数。

$$\text{interest} = \text{money}(1 + \text{rate})^{\text{year}} - \text{money}$$

输入输出示例：括号内为说明

输入

1000 3 0.025      (money = 1000, year = 3, rate = 0.025)

输出

interest = 76.89

```

#include <stdio.h>
#include <math.h>
int main(void)
{
    int money, year;
    double interest, rate;
    /*-----*/

scanf("%d%d%lf",&money,&year,&rate);

interest=money*pow((1+rate),year)-money;

    printf("interest = %.2f\n", interest);
}

```

20024程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat (0<repeat<10)，做repeat次下列运算：

输入 x，计算并输出下列分段函数 f(x) 的值(保留2位小数)，请调用 sqrt 函数求平方根，调用 pow 函数求幂。

当  $x \geq 0$  时， $f(x) = x^{0.5}$ ，当  $x$  小于 0 时， $f(x) = (x+1)^2 + 2x + 1/x$ 。

输入输出示例：括号内是说明

输入

3            (repeat=3)

10

-0.5

0

输出

$f(10.00) = 3.16$

$f(-0.50) = -2.75$

$f(0.00) = 0.00$

```
#include <stdio.h>
#include <math.h>
int main(void)
{
    int repeat, ri;
    double x, y;
    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        /*-----*/

        scanf("%lf", &x);

        if(x>=0) y=sqrt(x);

        else y=pow((x+1),2)+2*x+1/x;

        printf("f(%.2f) = %.2f\n", x, y);
    }
}
```

20025程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat ( $0 < \text{repeat} < 10$ )，做repeat次下列运算：

输入实数 x，计算并输出下列分段函数  $f(x)$  的值，输出时保留1位小数。

当 x 不等于10时， $y = f(x) = x$ ，当 x 等于10时， $y = f(x) = 1/x$ 。

输入输出示例：括号内是说明

输入

2            (repeat=2)

10

234

输出

$f(10.0) = 0.1$

$f(234.0) = 234.0$

```
#include <stdio.h>
int main(void)
{
    int repeat, ri;
    double x, y;
```

```

scanf("%d", &repeat);
for(ri = 1; ri <= repeat; ri++){
    /*-----*/

    scanf("%lf", &x);

    if(x!=10) y=x;

    else y=1/x;

    printf("f(%.1f) = %.1f\n", x, y);
}
}

```

20026程序填空，不要改变与输入输出有关的语句。

输入2个整数 num1 和 num2，计算并输出它们的和、差、积、商与余数。  
输出两个整数的余数可以用 printf("%d %% %d = %d\n", num1, num2, num1%num2);

输入输出示例：括号内是说明

输入

5 3            (num1=5, num2=3)

输出

5 + 3 = 8

5 - 3 = 2

5 \* 3 = 15

5 / 3 = 1

5 % 3 = 2

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

```
int main(void)
```

```
{
```

```
    int num1, num2;
```

```
    /*-----*/
```

```
    scanf("%d%d", &num1,&num2);
```

```
    printf("%d + %d = %d\n", num1, num2,
```

```
num1+num2);
```

```
    printf("%d - %d = %d\n", num1, num2, num1-num2);
```

```
    printf("%d * %d = %d\n", num1, num2,
```

```

num1*num2);

    printf("%d / %d = %d\n", num1, num2, num1/num2);

    printf("%d %% %d = %d\n", num1, num2,
num1%num2);

    return 0;
}

```

20031程序填空，不要改变与输入输出有关的语句。

计算表达式  $1 + 2 + 3 + \dots + 100$  的值。

输出示例：

sum = 5050

```

#include <stdio.h>
int main(void)
{
    int i, sum;
    /*-----*/

    sum=0;

    for(i=1;i<=100;i++)

        sum=sum+i;

    printf("sum = %d\n", sum);
}

```

20032程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat ( $0 < \text{repeat} < 10$ )，做repeat次下列运算：

输入一个正整数m ( $0 \leq m \leq 100$ )，计算表达式  $m + (m+1) + (m+2) + \dots + 100$  的值。

输入输出示例：括号内为说明

输入

```

3      (repeat=3)
0      (计算0+1+2+...+100)
10     (计算10+11+12+...+100)
50     (计算50+51+52+...+100)

```

输出

```

sum = 5050
sum = 5005
sum = 3825

```

```

#include <stdio.h>
int main(void)
{
    int i, m, sum;
    int repeat, ri;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &m);
        /*-----*/

        sum=0;

        for(i=m;i<=100;i++)

            sum=sum+i;

        printf("sum = %d\n", sum);
    }
}

```

20033程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat (0<repeat<10)，做repeat次下列运算：

输入2个正整数 m 和 n(m<=n)，计算表达式  $1/m + 1/(m+1) + 1/(m+2) + \dots + 1/n$  的值，输出时保留3位小数。

输入输出示例：括号内为说明

输入

```

3          (repeat=3)
5 15      (计算1/5+1/6+1/7+...+1/15)
10 20     (计算1/10+1/11+1/12+...+1/20)
1 3       (计算1+1/2+1/3)

```

输出

```

sum = 1.235
sum = 0.769
sum = 1.833

```

```

#include <stdio.h>
int main(void)
{
    int i, m, n;
    int repeat, ri;
    double sum;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){

```

```

scanf("%d%d", &m, &n);
/*-----*/

sum=0;

for(i=m;i<=n;i++)

    sum=sum+1.0/i;

    printf("sum = %.3f\n", sum);
}
}

```

20034程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat (0<repeat<10)，做repeat次下列运算：

输入一个正整数 n，计算表达式  $1 + 1/3 + 1/5 + \dots$  的前 n 项之和，输出时保留6位小数。

输入输出示例：括号内为说明

输入

```

2      (repeat=2)
5      (计算1+1/3+1/5+1/7+1/9)
23     (计算1+1/3+1/5+...+1/45)

```

输出

```

sum = 1.787302
sum = 2.549541

```

```

#include <stdio.h>
int main(void)
{
    int i, n;
    int repeat, ri;
    double sum;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &n);
        /*-----*/

        sum=0;

        for(i=1;i<=n;i++)

            sum=sum+1.0/(2*i-1);

        printf("sum = %.6f\n", sum);
    }
}

```

20035程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat (0<repeat<10)，做repeat次下列运算：

读入一个正整数 n，计算 $1-1/4+1/7-1/10+\cdots$ 的前 n 项之和，输出时保留3位小数。

输入输出示例：括号内是说明

输入

2 (repeat=2)

3

10

输出

sum = 0.893

sum = 0.819

```
#include <stdio.h>
int main(void)
{
    int flag, i, n, t;
    int repeat, ri;
    double item, sum;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &n);
        /*-----*/

        sum=0;flag=1;item=0;t=1;

        for(i=1;i<=n;i++)

            { item=flag*1.0/t;

                sum=sum+item;

                flag=-flag;

                t=t+3;

            }

        printf("sum = %.3f\n", sum);
    }
}
```

20036程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat (0<repeat<10)，做repeat次下列运算：

读入2个整数 lower 和 upper，输出一张华氏—摄氏温度转换表，华氏温度的取



值范围是[lower, upper], 每次增加2F。

计算公式:  $c = 5 * (f - 32) / 9$ , 其中: c表示摄氏温度, f表示华氏温度。

输出请使用语句 `printf("%3.0f %6.1f\n", fahr, celsius);`

输入输出示例: 括号内是说明

输入

```
2          (repeat=2)
32 35      (lower=32, upper=35)
40 30      (lower=40, upper=30)
```

输出

```
fahr  celsius
  32    0.0
  34    1.1
fahr  celsius
```

```
#include <stdio.h>
int main(void)
{
    int lower, upper;
    int repeat, ri;
    double celsius, fahr;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d%d", &lower, &upper);
        printf("fahr  celsius\n");

        /*-----*/

        for(fahr=lower;fahr<=upper;fahr=fahr+2){

            celsius=5 * (fahr- 32) / 9;

            printf("%3.0f %6.1f\n", fahr, celsius);

        }
    }
}
```

20037程序填空, 不要改变与输入输出有关的语句。

输入一个正整数repeat (0<repeat<10), 做repeat次下列运算:

输入2 个正整数 m 和 n, 计算  $m!+n!$ 。

输入输出示例: 括号内是说明

输入:

```
2          (repeat=2)
```

1 4 (m=1, n=4)

3 8 (m=3, n=8)

输出:

1! + 4! = 25

3! + 8! = 40326

```
#include "stdio.h"
int main(void)
{
    int i, m, n;
    int repeat, ri;
    double fm, fn;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++) {
        scanf("%d%d", &m, &n);
        /*-----*/

        fm=fn=1;

        for(i=1;i<=m;i++)

            fm=fm*i;

        for(i=1;i<=n;i++)

            fn=fn*i;

        printf("%d! + %d! = %.0f\n", m, n, fm+fn);
    }
}
```

20038程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat (0<repeat<10)，做repeat次下列运算：

读入1 个实数x和正整数 n(n<=50)，计算并输出 x 的 n 次幂(保留2位小数)，不允许调用pow函数求幂。

输入输出示例：括号内是说明

输入

2 (repeat=2)

1.5 2 (x=1.5, n=2)

2 7 (x=2, n=7)

输出

2.25

128.00

```
#include <stdio.h>
int main(void)
{
    int i, n;
    int repeat, ri;
    double mypow, x;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++) {
        scanf("%lf%d", &x, &n);
        /*-----*/

        mypow=1;

        for(i=1;i<=n;i++)

            mypow = mypow*x;

        printf("%.2f\n", mypow);
    }
}
```

20041程序填空，不要改变与输入输出有关的语句。

输入一个正整数n，生成一张3的乘方表，输出 $3^0 \sim 3^n$ 的值，可调用幂函数计算3的乘方。

输出使用语句 `printf("pow(3,%d) = %.0f\n", i, mypow);`

输入输出示例：括号内是说明

输入

3            (n=3)

输出

`pow(3,0) = 1`

`pow(3,1) = 3`

`pow(3,2) = 9`

`pow(3,3) = 27`

```
#include <stdio.h>
#include <math.h>
int main(void)
{
    int i, n;
    double mypow;
```

```

scanf("%d", &n);
/*-----*/

for(i=0;i<=n;i++){

    mypow=pow(3,i);

    printf("pow(3,%d) = %.0f\n", i, mypow);

}

return 0;
}

```

20042程序填空，不要改变与输入输出有关的语句。

输入一个正整数n，生成一张阶乘表，输出  $1! \sim n!$  的值，要求定义和调用函数 fact(n) 计算  $n!$ ，函数类型为double。

输出使用语句 `printf("%d! = %.0f\n", i, myfact);`;

输入输出示例：括号内是说明

输入

3            (n=3)

输出

1! = 1

2! = 2

3! = 6

```

#include <stdio.h>
int main(void)
{
    int i, n;
    double myfact;
    double fact(int n);

    scanf("%d", &n);
    /*-----*/

    for(i=1;i<=n;i++){

        myfact=fact(i);

        printf("%d! = %.0f\n", i, myfact);

    }

    return 0;
}

```

```

}
/*-----*/
double fact(int n)
{
    int i;

    double f=1;

    for(i=1;i<=n;i++)
        f=f*i;

    return f;
}

```

20043程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat (0<repeat<10)，做repeat次下列运算：

输入2个正整数 m 和 n(m<=n)，计算  $n! / (m! * (n-m)!)$  。

要求定义并调用函数fact(n)计算n的阶乘，其中 n 的类型是 int，函数类型是 double。

例：括号内是说明

输入：

2        (repeat=2)

2 7      (m=2, n=7)

5 12     (m=5, n=12)

输出：

result = 21

result = 792

```

#include "stdio.h"
double fact(int n);
int main(void)
{
    int m, n;
    int repeat, ri;
    double s;

```

```

scanf("%d", &repeat);
for(ri = 1; ri <= repeat; ri++){
    scanf("%d%d", &m, &n);
    /*-----*/

    s= fact(n)/(fact(m)*fact(n-m));

    printf("result = %.0f\n", s);
}

return 0;
}
/*-----*/

double fact(int n)
{
    int i;

    double f=1;

    for(i=1;i<=n;i++)
        f=f*i;

    return f;
}

```

20044程序填空，不要改变与输入输出有关的语句。

计算  $100^{0.5} + 101^{0.5} + \dots + 1000^{0.5}$  的值 (保留2位小数)，可调用sqrt函数计算平方根。

输入输出示例：括号内是说明

输出

sum = 20435.99

```

#include <stdio.h>
#include <math.h>
int main(void)
{

```

```

    int i;
    double sum;

/*-----*/

    sum=0;;

    for(i=100;i<=1000;i++)

        sum=sum+sqrt(i);

    printf("sum = %.2f\n", sum);
}

```

30001程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat (0<repeat<10)，做repeat次下列运算：

输入参数a,b,c，求一元二次方程 $a*x*x+b*x+c=0$ 的根，结果保留2位小数。

输出使用以下语句：

```

printf("参数都为零，方程无意义!\n");
printf("a和b为0，c不为0，方程不成立\n");
printf("x = %.2f\n", -c/b);
printf("x1 = %.2f\n", (-b+sqrt(d))/(2*a));
printf("x2 = %.2f\n", (-b-sqrt(d))/(2*a));
printf("x1 = %.2f+%.2fi\n", -b/(2*a), sqrt(-d)/(2*a));
printf("x2 = %.2f-%.2fi\n", -b/(2*a), sqrt(-d)/(2*a));

```

输入输出示例：括号内为说明

输入：

```

5                (repeat=5)
0 0 0            (a=0, b=0, c=0)
0 0 1            (a=0, b=0, c=1)
0 2 4            (a=0, b=2, c=4)
2.1 8.9 3.5      (a=2.1, b=8.9, c=3.5)
1 2 3            (a=1, b=2, c=3)

```

输出：

参数都为零，方程无意义！

a和b为0，c不为0，方程不成立

x = -2.00

x1 = -0.44

x2 = -3.80

x1 = -1.00+1.41i

x2 = -1.00-1.41i

```

#include <stdio.h>

```

```

#include <math.h>
int main(void)
{
    int repeat, ri;
    double a, b, c, d;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%lf%lf%lf", &a, &b, &c);
        /*-----*/
d=b*b-4*a*c;

if(a==0)
    if(b==0)
        if(c==0)
            printf("参数都为零，方程无意义!\n");
        else printf("a和b为0，c不为0，方程不成立\n");
    else printf("x = %0.2f\n", -c/b);
else if(d>=0){
    printf("x1 = %0.2f\n", (-b+sqrt(d))/(2*a));
    printf("x2 = %0.2f\n", (-b-sqrt(d))/(2*a));
    }
else
    { printf("x1 = %0.2f+%0.2fi\n", -b/(2*a),
sqrt(-d)/(2*a));
        printf("x2 = %0.2f-%0.2fi\n", -b/(2*a),
sqrt(-d)/(2*a));
    }

```



```

    }
}

```

30002程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个整数 x，计算并输出下列分段函数 sign(x) 的值。

$$y = \text{sign}(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}$$

输入输出示例：括号内是说明

输入

3 (repeat=3)

10 (x=10)

0 (x=0)

-98 (x=-98)

输出

sign(10) = 1 (x = 10时 y = 1)

sign(0) = 0 (x = 0时 y = 0)

sign(-98) = -1 (x = -98时 y = -1)

```

#include <stdio.h>
int main(void)
{
    int repeat, ri;
    int x, y;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &x);

        /*-----*/
        if(x>0) y=1;

        else if(x==0) y=0;

        else y=-1;

        printf("sign(%d) = %d\n", x, y);
    }

    return 0;
}

```

```
}
```

30003程序填空，不要改变与输入输出有关的语句。

输入10个字符，统计其中英文字母、空格或回车、数字字符和其他字符的个数。

输入输出示例：括号内是说明

输入

Reold 123?

输出

letter = 5, blank = 1, digit = 3, other = 1

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

```
int main(void)
```

```
{
```

```
    char c;
```

```
    int blank, digit, i, letter, other;
```

```
    blank = digit = letter = other = 0;
```

```
    for(i = 1; i <= 10; i++){
```

```
        c = getchar();
```

```
        /*-----*/
```

```
        if((c >= 'a' && c <= 'z' ) || ( c >= 'A' && c <= 'Z'))
```

```
            letter ++;
```

```
        else if(c >='0'&&c<='9')
```

```
            digit ++;
```

```
        else if(c == ' ' || c == '\n')
```

```
            blank ++;
```

```
        else other ++;
```

```
    }
```

```
    printf("letter = %d, blank = %d, digit = %d, other = %d\n", letter,  
blank, digit, other);
```

```
    return 0;
```

```
}
```

30004程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入五级制成绩(A—E)，输出相应的百分制成绩(0—100)区间，要求使用switch语句。

五级制成绩对应的百分制成绩区间为：A(90—100)、B(80—89)、C(70—79)、D(60—69)和E(0—59)，如果输入不正确的成绩，显示“Invalid input”。

输出使用以下语句：

```
printf("90-100\n");
printf("80-89\n");
printf("70-79\n");
printf("60-69\n");
printf("0-59\n");
printf("Invalid input\n");
```

输入输出示例：括号内是说明

输入

6ABCDEj (repeat=6，输入的五级成绩分别为A、B、C、D、E和无效的字符j)

输出

90-100

80-89

70-79

60-69

0-59

Invalid input (输入数据不合法)

```
#include <stdio.h>
int main(void)
{
    char ch;
    int repeat, ri;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        ch = getchar();
        /*-----*/

        switch(ch) {

        case 'A': printf("90-100\n");break;

        case 'B': printf("80-89\n");break;

        case 'C' : printf("70-79\n");break;
```

```

        case 'D' : printf("60-69\n");break;

        case 'E' : printf("0-59\n"); break;

        default: printf("Invalid input\n"); break;

    }

}

return 0;

}

```

30005程序填空，不要改变与输入输出有关的语句。

查询水果的单价。有4种水果，苹果(apples)、梨(pears)、桔子(oranges)和葡萄(grapes)，单价分别是3.00元/公斤，2.50元/公斤，4.10元/公斤和10.20元/公斤。

在屏幕上显示以下菜单(编号和选项)，用户可以连续查询水果的单价，当查询次数超过5次时，自动退出查询；不到5次时，用户可以选择退出。

当用户输入编号1~4，显示相应水果的单价(保留1位小数)；输入0，退出查询；输入其他编号，显示价格为0。

输入输出示例：括号内是说明

输入

3 (oranges的编号)

0 (退出查询)

输出

[1] apples

[2] pears

[3] oranges

[4] grapes

[0] Exit

price = 4.1

[1] apples

[2] pears

[3] oranges

[4] grapes

[0] Exit

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

```
int main(void)
```

```
{
```

```

int choice, i;
double price;

for(i = 1; i <= 5; i++) {
    printf("[1] apples\n");
    printf("[2] pears\n");
    printf("[3] oranges\n");
    printf("[4] grapes\n");
    printf("[0] Exit\n");
    scanf("%d", &choice);
    if(choice == 0)
        break;
    else{
        /*-----*/

        switch (choice) {

            case 1: price=3.0; break;

            case 2: price=2.5; break;

            case 3: price=4.1; break;

            case 4: price=10.2; break;

            default: price=0.0; break;

        }

        printf("price = %0.1f\n", price);
    }

return 0;
}

```

30006 程序填空，不要改变与输入输出有关的语句。

输入 5 个学生的数学成绩，判断他们的成绩是否及格。如果成绩低于 60，输出 "Fail"，否则，输出 "Pass"。

输入输出示例：括号内是说明

输入

61

59

92

40

60

输出

Pass  
Fail  
Pass  
Fail  
Pass

```
#include <stdio.h>
int main(void)
{
    int i, mark;

    for(i = 1; i <= 5; i++) {
        scanf("%d", &mark);
        /*-----*/

        if(mark>=60)    printf("Pass\n");

        else printf("Fail\n");
    }
}
```

30007程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入三角形的3条边 a, b, c，如果能构成一个三角形，输出面积 area 和周长 perimeter(保留2位小数)；否则，输出"These sides do not correspond to a valid triangle"。

在一个三角形中，任意两边之和大于第三边。三角形面积计算公式： $area = (s(s-a)(s-b)(s-c))^{0.5}$ ，其中  $s = (a+b+c)/2$

输出使用以下语句：

```
printf("area = %.2f, perimeter = %.2f\n", area, perimeter);
printf("These sides do not correspond to a valid triangle\n");
```

输入输出示例：括号内是说明

输入

```
2          (repeat=2)
5 5 3      (a=5, b=5, c=3)
1 4 1      (a=1, b=4, c=1)
```

输出

```
area = 7.15, perimeter = 13.00
```

```
These sides do not correspond to a valid triangle
```

```

#include <stdio.h>
#include <math.h>
int main(void)
{
    int a, b, c;
    int repeat, ri;
    double area, perimeter, s;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d%d%d", &a, &b, &c);
        /*-----*/
        if(a+b>c&&b+c>a&&c+a>b)

        {    perimeter=a+b+c;

            s= perimeter/2;

            area=sqrt(s*(s-a)*(s-b)*(s-c));

            printf("area = %.2f, perimeter = %.2f\n",area,
perimeter);

        }

        else

            printf("These sides do not correspond to a valid
triangle\n");

        }

        return 0;
    }
}

```

30008程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个职工的月薪 salary，输出应交的个人所得税 tax(保留2位小数)。

计算公式：tax = rate \* (salary - 850)

当 salary ≤850 时，rate = 0

当  $850 < \text{salary} \leq 1350$  时,  $\text{rate} = 5\%$   
当  $1350 < \text{salary} \leq 2850$  时,  $\text{rate} = 10\%$   
当  $2850 < \text{salary} \leq 5850$  时,  $\text{rate} = 15\%$   
当  $5850 < \text{salary}$  时,  $\text{rate} = 20\%$

输入输出示例：括号内是说明

输入

5 (repeat=5)  
1010.87 (salary=1010.87)  
32098.76 (salary=32098.76)  
800 (salary=800)  
4010 (salary=4010)  
2850 (salary=2850)

输出

tax = 8.04  
tax = 6249.75  
tax = 0.00  
tax = 474.00  
tax = 200.00

```
#include <stdio.h>
int main(void)
{
    int repeat, ri;
    double rate, salary, tax;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%lf", &salary);
        /*-----*/

        if(salary<=850) rate=0;

        else if(salary<=1350) rate=0.05;

        else if(salary<=2850) rate=0.1;

        else if(salary<=5850) rate=0.15;

        else rate=0.2;

        tax = rate * (salary - 850);
    }
}
```



```

        printf("tax = %0.2f\n", tax);
    }
}

```

30009程序填空，不要改变与输入输出有关的语句。

输入一个正整数 n，再输入 n 个学生的百分制成绩，统计各等级成绩的个数。

成绩等级分为五级，分别为A(90-100)、B(80-89)、C(70-79)、D(60-69)和E(0-59)。

输入输出示例：括号内是说明

输入

5 (n=5)

77 54 92 73 60

输出

Number of A(90-100): 1

Number of B(80-89): 0

Number of C(70-79): 2

Number of D(60-69): 1

Number of E(0-59): 1

```

#include <stdio.h>
int main(void)
{
    int i, mark, n;
    int na, nb, nc, nd, ne;

    scanf("%d", &n);
    na = nb = nc = nd = ne = 0;
    for(i = 1; i <= n; i++) {
        scanf("%d", &mark);
        /*-----*/
        if(mark >=90) na++;

        else if(mark >=80) nb++;

        else if(mark >=70) nc++;

        else if(mark >=60) nd++;

        else ne++;
    }
    printf("Number of A(90-100): %d\n", na);
    printf("Number of B(80-89): %d\n", nb);

```

```

    printf("Number of C(70-79): %d\n", nc);
    printf("Number of D(60-69): %d\n", nd);
    printf("Number of E(0-59): %d\n", ne);
}

```

30010程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat (0<repeat<10)，做repeat次下列运算：

输入一个形式如“操作数 运算符 操作数”的表达式，对2个整数进行乘、除和求余运算。

输出使用以下语句：

```

printf("%d * %d = %d\n", x, y, x * y);
printf("%d / %d = %d\n", x, y, x / y);
printf("%d Mod %d = %d\n", x, y, x % y);
printf("Invalid operator\n");

```

输入输出示例：括号内是说明

输入

4 (repeat=4)

21\*8

21/8

21%8

21!8

输出

21 \* 8 = 168

21 / 8 = 2

21 Mod 8 = 5

Invalid operator

```

#include <stdio.h>
int main(void)
{
    char sign;
    int x, y;
    int repeat, ri;

    scanf("%d",&repeat);
    for(ri = 1;ri <= repeat; ri++){
        scanf("%d%c%d", &x, &sign, &y);
        /*-----*/

```

**switch(sign){**

**case “\*”: printf("%d \* %d = %d\n", x, y, x \***

```

y);break;

        case '/': printf("%d / %d = %d\n", x, y, x / y);
break;

        case '%': printf("%d Mod %d = %d\n", x, y, x %
y); break;

        default: printf("Invalid operator\n"); break;

    }

}

return 0;
}

```

40011程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入两个正整数m和n，输出它们的最小公倍数和最大公约数。

输入输出示例：括号内为说明

输入：

```

3          (repeat=3)
3 7        (m=3, n=7)
24 4       (m=24, n=4)
24 18      (m=24, n=18)

```

输出：

21 is the least common multiple of 3 and 7, 1 is the greatest common divisor of 3 and 7.

24 is the least common multiple of 24 and 4, 4 is the greatest common divisor of 24 and 4.

72 is the least common multiple of 24 and 18, 6 is the greatest common divisor of 24 and 18.

```

#include <stdio.h>
int main(void)
{
    int gcd, lcm, m, n;
    int repeat, ri;

```

```

scanf("%d", &repeat);
for(ri = 1; ri <= repeat; ri++){
    scanf("%d", &m);
    scanf("%d", &n);
    if(m <= 0 || n <= 0)
        printf("m <= 0 or n <= 0");
    else{
        /*-----*/
        for(lcm=m;lcm%n!=0;lcm=lcm+m)
        ;

        gcd= m*n/lcm;

    }

    printf("%d is the least common multiple of %d and %d, %d is
the greatest common divisor of %d and %d.\n", lcm, m, n, gcd, m, n);
}

return 0;
}

```

40012程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat (0<repeat<10)，做repeat次下列运算：

读入1个正实数 eps，计算并输出下式的值，精确到最后一项的绝对值小于eps(保留6位小数)。请使用 while 语句实现循环。

求 $1 - 1/4 + 1/7 - 1/10 + 1/13 - 1/16 + \dots$ 。

输入输出示例：括号内是说明

输入

2 (repeat=2)

1E-4 (eps=1E-4)

0.1 (eps=0.1)

输出

sum = 0.835699

sum = 0.869780

```

#include <stdio.h>
#include <math.h>
int main(void)
{
    int denominator, flag;
    int repeat, ri;

```

```

double eps, item, sum;

scanf("%d", &repeat);
for(ri = 1; ri <= repeat; ri++) {
    scanf("%le", &eps);
    /*-----*/

sum=0; denominator =1;flag=1;item=1;

while(fabs(item)>=eps)

{ item= flag*1.0/ denominator;

sum=sum+item;

denominator = denominator +3;

flag=-flag;

}

    printf("sum = %.6f\n", sum);
}
}

```

40013程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat (0<repeat<10)，做repeat次下列运算：

读入一批正整数(以零或负数为结束标志)，求其中的奇数和。请使用while语句实现循环。

输入输出示例：括号内是说明

输入

2 (repeat=2)

1 3 90 7 0

8 7 4 3 70 5 6 101 -1

输出

The sum of the odd numbers is 11.

The sum of the odd numbers is 116.

```

#include <stdio.h>
int main(void)
{
    int x, sum;
    int repeat, ri;

```

```

scanf("%d", &repeat);
for(ri = 1; ri <= repeat; ri++){
    scanf("%d", &x);
    /*-----*/

    sum=0;

    while(x>0)

    {

        if(x%2) sum=sum+x;

        scanf("%d", &x);

    }

    printf("The sum of the odd numbers is %d.\n", sum);
}
}

```

40014程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个整数 in，求它的位数。例如123的位数是3。请使用do-while语句实现循环。

输入输出示例：括号内是说明

输入

4 (repeat=4)

12345 (in=12345)

-100 (in=-100)

-1 (in=-1)

1290 (in=1290)

输出

count = 5 (12345的位数是5)

count = 3 (-100的位数是3)

count = 1 (-1的位数是1)

count = 4 (99的位数是2)

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

```
int main(void)
```

```
{
```

```
    int count, in;
```

```
    int repeat, ri;
```

```
    scanf("%d", &repeat);
```

```

for(ri = 1; ri <= repeat; ri++){
    scanf("%d", &in);
    /*-----*/

    count =0;

    if(in<0) in=-in;

    do{

        count ++;

        in=in/10;

    }while(in);

    printf("count = %d\n", count);
}
}

```

40015程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat (0<repeat<10)，做repeat次下列运算：

输入一个正整数n，再输入n个整数，输出最小值。

输入输出示例：括号内是说明

输入

```

3      (repeat=3)
4 -2 -123 100 0
4 -9 -1 1 -8
3 5 3 1

```

输出

```

min = -123
min = -9
min = 1

```

```

#include <stdio.h>
int main(void)
{
    int i, min, n, x;
    int repeat, ri;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &n);
        scanf("%d", &x);
        /*-----*/

        min=x;

```

```

for(i=1;i<n;i++){

    scanf("%d", &x);

    if(x<min) min=x;}

```

```

    printf("min = %d\n", min);
}
}

```

40016程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个整数 in，求它的位数以及各位数字之和。例如 123 的各位数字之和是 6，位数是 3。

输入输出示例：括号内是说明

输入

```

4          (repeat=4)
0          (in=0)
23456      (in=23456)
-100       (in=-100)
-1         (in=-1)

```

输出

```

count = 1, sum = 0      (0的位数是1，各位数字之和是0)
count = 5, sum = 20     (23456的位数是5，各位数字之和是20)
count = 3, sum = 1      (-100的位数是3，各位数字之和是1)
count = 1, sum = 1      (-1的位数是1，各位数字之和是1)

```

```

#include <stdio.h>
int main(void)
{
    int count, in, sum;
    int repeat, ri;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &in);
        /*-----*/

        count=sum=0;

        if(in<0) in=-in;

```



```

if(in==0) {count=1;sum=0;}

while(in!=0)

{ sum=sum+in%10;

count++;

in=in/10;

}

printf("count = %d, sum = %d\n", count, sum);
}
}

```

40017程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个正整数 n，输出  $2/1+3/2+5/3+8/5+\cdots$  前n项之和，保留2位小数。  
(该序列从第2项起，每一项的分子是前一项分子与分母的和，分母是前一项的分子)

输入输出示例：括号内是说明

输入

```

3      (repeat=3)
1      (n=1)
5      (n=5)
20     (n=20)

```

输出

```

sum = 2.00      (第1项是2.00)
sum = 8.39      (前5项的和是8.39)
sum = 32.66     (前20项的和是32.66)

```

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

```
int main(void)
```

```
{
```

```
    int i, n;
```

```
    int repeat, ri;
```

```
    double denominator, numerator, sum, temp;
```

```
    scanf("%d", &repeat);
```

```
    for(ri = 1; ri <= repeat; ri++){
```

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

```
        /*-----*/
```

```
        denominator=1; numerator =2;sum=0;
```

```

        for(i=0;i<n;i++)
        { sum=sum+ numerator / denominator;

          temp= numerator;

          numerator = denominator + numerator;

          denominator =temp;

        }

        printf("sum = %.2f\n", sum);
    }
}

```

40018程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入2个正整数 a 和 n，求  $a+aa+aaa+aa\cdots a$  (n个a) 之和。例如，输入 2 和 3，输出 246 (2 + 22 + 222)。

输入输出示例：括号内是说明

输入

```

3          (repeat=3)
2 3        (a=2, n=3)
5 4        (a=5, n=4)
1 1        (a=1, n=1)

```

输出

```

sum = 246          (2+22+222)
sum = 6170         (5+55+555+5555)
sum = 1            (1)

```

```

#include <stdio.h>
int main(void)
{
    int a, i, n, sn, tn;
    int ri, repeat;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%ld%d", &a, &n);
        /*-----*/
    }
}

```

```

    sn=0;  tn=1;

    for(i=0;i<n;i++) {

        sn=sn+tn;

        tn=tn*10+1;

    }

    sn=sn*a;

    printf("sum = %d\n", sn);
}
}

```

40019程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

皮球从 height（米）高度自由落下，触地后反弹到原高度的一半，再落下，再反弹，……，如此反复。问皮球在第 n 次落地时，在空中一共经过多少距离？

第 n 次反弹的高度是多少？（输出保留1位小数）

输入输出示例：括号内是说明

输入

```

3                (repeat=3)
10 2            (height=10m, n=2)
4 1             (height=2m, n=1)
100 8          (height=100m, n=8)

```

输出

distance=20.0, height=2.5 (第2次落地时，空中经过距离为20米，第2次反弹高度为2.5米)

distance=4.0, height=2.0 (第1次落地时，空中经过距离为4米，第1次反弹高度为2米)

distance=298.4, height=0.4 (第8次落地时，空中经过距离为298.4米，第8次反弹高度为0.4米)

```

#include <stdio.h>
int main(void)
{
    int i, n;
    int repeat, ri;
    double distance, height;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){

```

```

scanf("%lf%d", &height, &n);
/*-----*/

    distance = height ; height = height /2;

    for(i=2;i<=n;i++){

        distance = distance +2* height;

        height = height /2;

    }

    printf("distance = %.1f, height = %.1f\n", distance, height);
}
}

```

40021程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入1个正整数n，计算下式的前n项之和(保留4位小数)。要求使用嵌套循环。

$$e = 1 + 1/1! + 1/2! + \dots + 1/n!$$

输入输出示例：括号内为说明

输入：

2           (repeat=2)

2           (n=2)

10          (n=10)

输出：

e = 2.5000

e = 2.7183

```

#include "stdio.h"
int main(void)
{
    int i, j, n;
    int repeat, ri;
    double e, product;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &n);
        /*-----*/

        e=1;

        for(i=1;i<=n;i++){

            product =1;

```

```

for(j=1;j<=i;j++)

    product = product *j;

    e=e+1.0/ product ;

}

```

```

    printf("e = %0.4f\n", e);
}
}

```

40022程序填空，不要改变与输入输出有关的语句。

输入一个正整数repeat (0<repeat<10)，做repeat次下列运算：

输入2个正整数 m 和 n(1<=m, n<=500)，输出 m 和 n 之间的所有素数，每行输出6个。素数就是只能被1和自身整除的正整数，1不是素数，2是素数。

输出语句：printf("%d ", i);

输入输出示例：括号内为说明

输入：

```

2                (repeat=2)
1 35             (m=1, n=35)
2 10             (m=2, n=10)

```

输出：

```

primes:          (1到35之间的素数)
2 3 5 7 11 13
17 19 23 29 31
primes:          (2到10之间的素数)
2 3 5 7

```

```

#include "stdio.h"
#include "math.h"
int main(void)
{
    int count, i, j, k, m, n;
    int ri,repeat;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat;ri++){
        scanf("%d%d", &m, &n);
        printf("primes:\n");
        /*-----*/

        if(m==1) m=m+1;

        count =0;

```

```

    for (i =m ; i <=n ; i ++)  

    {   k =sqrt (i );  

        for (j=2;j<=k ;j++)  

            if (i %j ==0) break;  

        if (j>k ) {  

            printf("%d ", i);  

            count++;  

            if(count%6==0) printf("\n");  

        }  

    }  

    printf("\n");  

}

```

40023程序填空，不要改变与输入输出有关的语句。？

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

将一笔零钱（大于8分，小于1元，精确到分）换成5分、2分和1分的硬币。

输入金额，问有几种换法？针对每一种换法，输出各种面额硬币的数量和硬币的总数量，要求每种硬币至少有一枚。先输出硬币总数量少的换法。

输出使用语句：printf("fen5:%d, fen2:%d, fen1:%d, total:%d\n", fen5, fen2, fen1, fen5+fen2+fen1);

输入输出示例：括号内为说明

输入：

2	(repeat=2)
10	(money=10分)
13	(money=13分)

输出：

fen5:1, fen2:2, fen1:1, total:4	
fen5:1, fen2:1, fen1:3, total:5	
count = 2	(10分有2种换法)
fen5:2, fen2:1, fen1:1, total:4	
fen5:1, fen2:3, fen1:2, total:6	
fen5:1, fen2:2, fen1:4, total:7	
fen5:1, fen2:1, fen1:6, total:8	
count = 4	(13分有4种换法)

```

#include "stdio.h"
int main(void)
{
    int count, fen1, fen2, fen5, money;
    int repeat, ri;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &money);
        /*-----*/

        count=0;

        for(fen1=1; fen1<=money; fen1++)

        for(fen2=1; fen2<=money/2; fen2++)

        for(fen5=1; fen5<=20/5; fen5++)

        if(fen1*1+ fen2*2+ fen5*5==money)

        {count++;

        printf("fen5:%d,fen2:%d,fen1:%d,total:%d\n",fen5,
fen2, fen1, fen5+fen2+fen1);

}

        printf("count = %d\n", count);
    }
}

```

```

#include "stdio.h"
int main(void)
{
    int count, fen1, fen2, fen5, money;
    int repeat, ri;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &money);

```

```

count=0;
for(fen5= money/5; fen5>=1; fen5--)
    for(fen2= money/2; fen2>=1; fen2--)
        for(fen1=1; fen1< money ; fen1++)
            if(fen1*1+ fen2*2+ fen5*5==money) {
                count++;
printf("fen5:%d, fen2:%d, fen1:%d, total:%d\n", fen5, fen2, fen1,
fen5+fen2+fen1);
}
    printf("count = %d\n", count);
}
}

```

40024程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入两个正整数 m 和 n (1<=m, n<=1000)，输出 m 到 n 之间的所有满足各位数字的立方和等于它本身的数。

例如153的各位数字的立方和是  $1^3+5^3+3^3=153$ 。

输出使用语句：printf("%d\n", i);

输入输出示例：括号内为说明

输入：

```

2                (repeat=2)
100 400          (m=100, n=400)
1 100            (m=1, n=100)

```

输出：

result:

```

153              (1*1*1+5*5*5+3*3*3=153)
370              (3*3*3+7*7*7=370)
371              (3*3*3+7*7*7+1*1*1=371)

```

result:

```

1                (1*1*1=1)

```

```
#include "stdio.h"
```

```
int main(void)
```

```
{
```

```
    int i, digit, m, n, number, sum;
```

```
    int repeat, ri;
```

```
    scanf("%d", &repeat);
```

```
    for(ri = 1; ri <= repeat; ri++) {
```

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

```
        printf("result:\n");
```

```
        /*-----*/

```



```

        for(i=m;i<=n; i++)
        {
            sum=0;number=i;

            while (number)
            {
                digit =number %10;

                sum =sum +digit *digit *digit ;

                number = number /10;

            }

            if(i==sum)  printf(“%d\n”,i );

        }

    }
}

```

40025程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入两个正整数 m 和 n(1<=m, n<=1000)，输出 m 到 n之间的所有完数，并输出其因子。一个数如恰好等于它的因子之和，这个数称为完数，例如，6=1+2+3，其中1、2、3为因子，6为因子和。

输出使用以下语句：

```

printf(“%d = 1”, number);
printf(“ + %d”, factor);
printf(“\n”);

```

输入输出示例：括号内为说明

输入：

```

2                (repeat=2)
1 30             (m=1, n=30)
400 500          (m=400, n=500)

```

输出

result:

```

1 = 1
6 = 1 + 2 + 3
28 = 1 + 2 + 4 + 7 + 14

```

result:

```

496 = 1 + 2 + 4

```

```

#include <stdio.h>
int main(void)
{
    int factor, m, n, number, sum;
    int repeat, ri;

    scanf("%d",&repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d%d", &m, &n);
        printf("result:\n");
        /*-----*/
for(factor=m;factor<=n;factor++)
{
        sum=1;
for(number=2;number<=factor/2;number++)
    if(factor%number==0) sum+=number;
    if(sum==factor)
    {
        printf("%d = 1",factor);
for(number=2;number<=factor/2;number++)
    if(factor%number==0) printf("
+ %d",number);
    printf("\n");
    }
}
}
}
}

```

50001 程序填空，不要改变与输入输出有关的语句。

从键盘输入一个正整数 n，计算 n! 的值。要求定义和调用函数 fact(n)，计算 n!，函数形参 n 的类型是 int，函数类型是 double。

输入输出示例：括号内是说明

5            (n=5)  
5! = 120.000000

```

#include <stdio.h>
int main(void)
{
    int n;
    double factorial;
    double fact(int n);

    scanf ("%d", &n);
    /*-----*/

```

```

    factorial=fact(n);
    printf("%d! = %f\n", n, factorial);
}

/*-----*/
double fact(int n)
{ int i;double s=1;
  for(i=1;i<=n;i++)
    s=s*i;
  return s;
}

```

50002 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入 1 个整数 x，若 x 大于 0，sign(x) = 1；若 x 等于 0，sign(x) = 0；否则，sign(x) = -1，最后输出 sign(x) 的值。

要求定义和调用函数 sign(x) 实现该分段函数，函数形参 x 的类型是 int，函数类型是 int。

输入输出示例：括号内是说明

输入

```

3          (repeat=3)
10         (x=10)
-5         (x=-5)
0          (x=0)

```

输出

```

sign(10) = 1      (x=10 时 sign(x) 的值为 1)
sign(-5) = -1     (x=-5 时 sign(x) 的值为 -1)
sign(0) = 0       (x=0 时 sign(x) 的值为 0)

```

```

#include <stdio.h>
int sign(int x);
int main(void)
{
    int x, y;

```

```

    int repeat, ri;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d",&x);
    /*-----*/

        y= sign(x);

        printf("sign(%d) = %d\n", x, y);
    }
}

/*-----*/

int sign(int x)
{ if(x>0) return 1;
    else if(x==0) return 0;
    else return -1;
}

```

50003 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一批正整数（以零或负数为结束标志），求其中的奇数和。

要求定义和调用函数 even(n) 判断数的奇偶性，当 n 为偶数时返回 1，否则返回 0，函数形参 n 的类型是 int，函数类型是 int。

输入输出示例：括号内是说明

输入

```

2      (repeat=2)
12 9 7 18 3 11 20 0
11 8 6 17 2 10 19 -1

```

输出

```

The sum of the odd numbers is 30.
The sum of the odd numbers is 47.

```

```

#include <stdio.h>
int even(int n);

```

```

int main(void)
{
    int n, sum;
    int ri, repeat;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d",&n);
        /*-----*/

        sum=0;

        while(n>0){

            if(even(n)==0)

                sum=sum+n;

                scanf("%d",&n);

        }

        printf("The sum of the odd numbers is %d.\n", sum);
    }
}

/*-----*/

int even(int n)

{ if(n%2==0) return 1;

else return 0;

}

```

50004 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

给定平面任意两点坐标 (x1, y1) 和 (x2, y2)，求这两点之间的距离(保留 2 位小数)。

要求定义和调用函数 dist(x1, y1, x2, y2) 计算两点间的距离，函数形参 x1、y1、x2 和 y2 的类型都是 double，函数类型是 double。

输入输出示例：括号内是说明

输入

1 (repeat=1)

10 10 (x1=10, y1=10)

200 100 (x2=200, y2=100)

输出

Distance = 210.24

```
#include <stdio.h>
#include <math.h>
double dist(double x1, double y1, double x2, double y2);
int main(void)
{
    int repeat, ri;
    double distance, x1, y1, x2, y2;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%lf%lf%lf%lf", &x1, &y1, &x2, &y2);
        /*-----*/

        distance=dist(x1,x2,y1,y2);

        printf("Distance = %.2f\n", distance);
    }
}

/*-----*/

double dist(double x1, double x2, double y1, double
y2)
{

    return sqrt(pow((x1-x2),2)+pow((y1-y2),2));

}
```

50005 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入 2 个正整数 m 和 n (1<=m,n<=500)，统计并输出 m 到 n 之间的素数的个数以及这些素数的和。素数就是只能被 1 和自身整除的正整数，1 不是素数，2 是素数。

要求定义并调用函数 prime(m) 判断 m 是否为素数，当 m 为素数时返回 1，否则返回 0，函数形参 m 的类型是 int，函数类型是 int。

输入输出示例：括号内是说明

输入：

1 (repeat=1)

1 10 (m=1, n=10)

输出:

Count = 4, sum = 17 (1 到 10 之间有 4 个素数: 2, 3, 5, 7)

```
#include "stdio.h"
#include "math.h"
int main(void)
{
    int count, i, m, n, sum;
    int repeat, ri;
    int prime(int m);

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++) {
        scanf("%d%d", &m, &n);
        /*-----*/

        count=sum=0;

        for (i=m ; i<=n ; i ++)

            if(prime(i)) {count++;sum=sum+i;}

        printf("Count = %d, sum = %d\n", count, sum);
    }
}

/*-----*/

int prime(int n )

{ int i ;

if(n==1) return 0;

for (i=2;i<=sqrt (n );i++)

if(n%i ==0) return 0 ;

return 1;

}
```

50006 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

读入 1 个整数 in,再输入一个数字 digit ( $0 \leq \text{digit} < 10$ ),统计并输出整数 in 中数字 digit 的个数。

要求定义并调用函数 countdigit(number,digit), 它的功能是统计整数 number 中数字 digit 的个数,函数形参 number 和 digit 的类型是 int,函数类型是 int。

例如, countdigit(10090,0)的返回值是 3。

输入输出示例: 括号内是说明

输入

```
2          (repeat=2)
21252 2    (number=21252, digit=2)
-1111 9    (number=-1111, digit=9)
```

输出

```
Number 21252 of digit 2: 3      (21252 中有 3 个 2)
Number -1111 of digit 9: 0      (-1111 中有 0 个 9)
```

```
#include "stdio.h"
int main(void)
{
    int count, digit, in;
    int repeat, ri;
    int countdigit(int number, int digit);

    scanf("%d",&repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d%d", &in, &digit);
        /*-----*/

        count= countdigit(in, digit);

        printf("Number %d of digit %d: %d\n", in, digit, count);
    }
}

/*-----*/

int countdigit(int number, int digit)

{ int n=0,m;

    if(number<0) number=-number;

    while(number)

    { m=number%10;

        if(digit==m) n++;
```



```

        number=number/10;

    }

    return n;

}

```

50007 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入 2 个正整数 m 和 n(1<=m,n<=1000)，输出 m 到 n 之间的所有水仙花数。

水仙花数是指各位数字的立方和等于其自身的数。

要求定义并调用函数 is(number)判断 number 的各位数字之立方和是否等于其自身，若相等则返回 1，否则返回 0，函数形参 number 的类型是 int，函数类型是 int。

输出使用语句：printf("%d\n", i);

输入输出示例：括号内是说明

输入：

```

2          (repeat=2)
100 400    (m=100, n=400)
1 100      (m=1, n=100)

```

输出：

```

result:    (100 到 400 之间的水仙花数)
153        (1*1*1+5*5*5+3*3*3=153)
370        (3*3*3+7*7*7=370)
371        (3*3*3+7*7*7+1*1*1=371)
result:    (1 到 100 之间的水仙花数)
1          (1*1*1=1)

```

```

#include "stdio.h"
int main(void)
{
    int i, m, n;
    int repeat, ri;
    int is(int number);

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d%d", &m, &n);
        printf("result:\n");
    }
}

```

```

/*-----*/
for(i=m ; i<=n ; i++)
    if(is(i)) printf("%d\n", i);
}
}

/*-----*/
int is(int number)
{
    int s=0,m,digit=number;
    while(digit)
    {
        m=digit%10;
        s=s+m*m*m;
        digit =digit/10;
    }
    if(s==number) return 1;
    else return 0;
}

```

50008 程序填空，不要改变与输入输出有关的语句。

输入一个正整数  $m$  ( $0 < m < 10$ )，求  $1! + 2! + \dots + m!$ ，要求定义并调用函数  $\text{fact}(n)$  计算  $n!$ ，函数形参  $n$  的类型是 `int`，函数类型是 `double`。

输入输出示例：括号内是说明

输入

5           (m=5)

输出：

$1!+2!+\dots+5! = 153.000000$

```

#include <stdio.h>
double fact(int n);

```

```

int main(void)
{
    int i, m;
    double sum;

    scanf("%d", &m);
    /*-----*/

    sum=0;

    for(i=1;i<=m;i++)

        sum=sum+fact(i);

    printf("1!+2!+...+%d! = %f\n", m, sum);
}

/*-----*/

double fact(int n)
{ int i;double s=1;
    for(i=1;i<=n;i++)
        s=s*i;

    return s;
}

```

60001 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入精度 eps，根据下式求 PI 的值，直到最后一项小于 eps。

$$PI/2 = 1 + 1/(1*3) + 2!/(1*3*5) + 3!/(1*3*5*7) + \dots + n!/(1*3*5*\dots*(2n+1))$$

要求定义并调用函数 fact(n) 计算 n!，函数形参 n 的类型是 int，函数类型是 double；定义并调用函数 multi(n) 计算 1\*3\*5\*…\*n，函数形参 n 的类型是 int，函数类型是 double。

输入输出示例：括号内是说明

输入：

2 (repeat=2)

1E-6 (eps=1E-6)

1E-5            (eps=1E-5)

输出示例:

PI = 3.14159

PI = 3.14158

```
#include <stdio.h>
double fact(int n);
double multi(int n);
int main(void)
{
    int i;
    int repeat, ri;
    double eps, sum, item;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%le", &eps);
/*-----*/

        sum=1;item=1;i=1;

        while(item>=eps){

            item=fact(i)/multi(i);

            sum=sum+item;

            i++;

        }
        printf("PI = %0.5f\n", 2 * sum);
    }

/*-----*/

double fact(int n)

{ int i;double s=1;

    for(i=1;i<=n;i++)

        s=s*i;
```

```

    return s;

}

/*-----*/

double multi(int n)
{ int i;double s=1;
  for(i=1;i<=n;i++)
    s=s*(2*i+1);
  return s;
}

```

60002 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

依次以十进制、八进制和十六进制形式输入三个整数，输出每个整数的十进制、八进制和十六进制形式。

输入输出示例：括号内是说明

输入：

1 (repeat=1)

31 11 1a (十进制数 31，八进制数 11，十六进制数 1a)

输出：

The decimal is 31, the octal is 37, the hexadecimal is 1f. (十进制数 31 的十进制、八进制和十六进制形式是 31, 37, 1f)

The decimal is 9, the octal is 11, the hexadecimal is 9. (八进制数 11 的十进制、八进制和十六进制形式是 9, 11, 9)

The decimal is 26, the octal is 32, the hexadecimal is 1a. (十六进制数 1a 的十进制、八进制和十六进制形式是 26, 32, 1a)

```

#include <stdio.h>
int main(void)
{
    int decimal, octal, hexadecimal;
    int repeat, ri;

```

```

scanf("%d", &repeat);
for(ri = 1; ri <= repeat; ri++){
    scanf("%d%o%x", &decimal, &octal, &hexadecimal);

/*-----*/

    printf("The decimal is %d, the octal is %o, the
hexadecimal is %x.\n", decimal, decimal, decimal);

    printf("The decimal is %d, the octal is %o, the
hexadecimal is %x.\n", octal, octal, octal);

    printf("The decimal is %d, the octal is %o, the
hexadecimal is %x.\n", hexadecimal, hexadecimal,
hexadecimal);
}
}

```

60003 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一行字符，以回车结束，统计出其中的英文字母、空格、数字和其他字符的个数。

输入输出示例：括号内是说明

输入

2 (repeat=2)

Reold building room 123.

Programming is fun!

输出

letter = 17, blank = 3, digit = 3, other = 1 ("Reold building room 123." 中的英文字母、空格、数字和其他字符的个数)

letter = 16, blank = 2, digit = 0, other = 1 ("Programming is fun!" 中的英文字母、空格、数字和其他字符的个数)

```

#include <stdio.h>
int main(void)
{

```

```

char c;
int blank, digit, letter, other;
int ri, repeat;

scanf("%d", &repeat);
getchar();
for(ri = 1; ri<=repeat; ri++){
    c = getchar();
/*-----*/

    letter =digit =blank =other =0;

    while (c!='\n'){

        if(c>='A'   &&   c<='Z' || c>='a'   &&   c<='z')

letter++;

        else if(c>='0' && c<='9')   digit++;

        else if(c==' ')   blank++;

        else   other++;

        c = getchar();

    }

    printf("letter = %d, blank = %d, digit = %d, other = %d\n", letter,
blank, digit, other);
}
}

```

60006 程序填空，不要改变与输入输出有关的语句。

验证哥德巴赫猜想:任何一个大于6的偶数均可表示为两个素数之和。例如 6=3+3, 8=3+5, …, 18=7+11。素数就是只能被1和自身整除的正整数，1不是素数，2是素数。

输入两个正整数 m 和 n (6≤m≤n≤100)，将 m 到 n 之间的偶数表示成两个素数之和，打印时一行打印5组。

要求定义并调用函数 prime(m) 判断 m 是否为素数，当 m 为素数时返回1，否则返回0，函数形参 m 的类型是 int，函数类型是 int。

输出使用语句: printf("%d=%d+%d ", number, i, number - i);

输入输出示例：括号内为说明

输入：

89 100 (m=89, n=100)

输出：

90=7+83 92=3+89 94=5+89 96=7+89 98=19+79  
100=3+97

```
#include "stdio.h"
#include "math.h"
int main(void)
{
    int count, i, m, n, number;

    int prime(int m);

    scanf("%d%d", &m, &n);
    if(m % 2 != 0) m = m + 1;
    if(m >= 6) {
        /*-----*/

        for(number=m; number <= n; number = number
+2 ){

            for(i = 3;i<= number/2;i=i+2)

                if(prime(i)==1&& prime(number-i)==1){

                    printf("%d=%d+%d ",number,i,number-i);

                    count++;

                    if(count%5==0) printf("\n");

                    break;

                }

            }

        }

    }
}
/*-----*/

int prime(int m)

{    int i, n;
```



```

        if(m == 1) return 0;

        n = sqrt(m);

        for(i = 2; i <= n; i++)

            if(m%i == 0) return 0;

        return 1;

    }

```

60007 程序填空，不要改变与输入输出有关的语句。

输入 2 个整数，分别将其逆向输出。

要求定义并调用函数 fun(n)，它的功能是返回 n 的逆向值，函数形参 n 的类型是 int，函数类型是 int。例如，fun(123)的返回值是 321。

输入输出示例：括号内是说明

输入：

123

-910

输出：

123 的逆向是 321

-910 的逆向是-19

```

#include <stdio.h>
int fun(int n);
int main(void)
{
    int m1,m2;

    scanf("%d%d", &m1, &m2);
    printf("%d 的逆向是%d\n", m1, fun(m1));
    printf("%d 的逆向是%d\n", m2, fun(m2));

}

```

/\*-----\*/

```

int fun(int n)
{
    int s=0;int digit,flag=1;

```

```

    if(n<0) {n=-n;flag=-1;}

    while(n)
    {
        digit=n%10;

        s=s*10+digit;

        n=n/10;
    }

    return flag*s;
}

```

70011 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat ( $0 < \text{repeat} < 10$ )，做 repeat 次下列运算：

输入一个正整数 n ( $0 < n \leq 9$ ) 和一组 (n 个) 有序的整数，再输入一个整数 x，把 x 插入到这组数据中，使该组数据仍然有序。

输入输出示例：括号内为说明

输入：

4	(repeat=4)
5	(数据的个数 n=5)
1 2 4 5 7	(5 个有序整数)
3	(待插入整数 x=3)
4	(数据的个数 n=4)
1 2 5 7	(4 个有序整数)
-10	(待插入整数 x=-10)
3	(数据的个数 n=3)
1 2 4	(3 个有序整数)
100	(待插入整数 x=100)
5	(数据的个数 n=5)
1 2 4 5 7	(5 个有序整数)
4	(待插入整数 x=4)

输出：

```

1 2 3 4 5 7
-10 1 2 5 7
1 2 4 100
1 2 4 4 5 7

```

```

#include <stdio.h>
int main(void)
{
    int i, j, n, x;
    int repeat, ri;
    int a[10];

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++) {
        scanf("%d", &n);
        for(i = 0; i < n; i++)
            scanf("%d", &a[i]);
        scanf("%d", &x);
        /*-----*/

        j=n;

        for(i = 0; i < n; i++)

            if(x<=a[i]) {j=i;break;}

        for(i= n-1; i>=j; i--)

            a[i+1]=a[i];

        a[j]=x;

        for(i = 0; i < n + 1; i++)
            printf("%d ", a[i]);
        putchar('\n');
    }
}

```

70012 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个正整数 n (1<n<=10)，再输入 n 个整数，输出平均值(保留 2 位小数)。

输入输出示例：括号内为说明

输入

2 (repeat=2)

3 (n=3)

1 2 -6

5 (n=5)

12 2 5 4 0

输出

```
average = -1.00
average = 4.60
```

```
#include <stdio.h>
int main(void)
{
    int i, n, sum;
    int repeat, ri;
    int a[10];
    double aver;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &n);
        for(i = 0; i < n; i++)
            scanf("%d", &a[i]);
        /*-----*/

        sum=0;

        for(i=0; i<n; i++)

            sum+=a[i];

        aver=(double)sum/n;

        printf("average = %.2f\n", aver);
    }
}
```

70013 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个正整数 n (1<n<=10)，再输入 n 个整数，存入数组 a 中，先将数组 a 中的这 n 个数逆序存放，再按顺序输出数组中的 n 个元素。

输入输出示例：括号内为说明

输入

2 (repeat=2)

4 (n=4)

10 8 1 2

5 (n=5)

1 2 5 4 0

输出

2 1 8 10

0 4 5 2 1

```
#include <stdio.h>
int main(void)
{
    int i, n, temp;
    int repeat, ri;
    int a[10];

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++) {
        scanf("%d", &n);
        for(i = 0; i < n; i++)
            scanf("%d", &a[i]);
        /*-----*/
        for(i=0;i<n/2;i++)
        { temp=a[i];a[i] =a[n-1-i]; a[n-1-i]=temp;}

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

70014 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat ( $0 < \text{repeat} < 10$ )，做 repeat 次下列运算：

输入一个正整数 n ( $1 < n \leq 10$ )，再输入 n 个整数，输出最大值及其对应的最小下标，下标从 0 开始。

输入输出示例：括号内为说明

输入

3 (repeat=3)

3 (n=3)

1 6 4

4 (n=4)

10 8 1 9

5 (n=5)

1 2 0 4 5

输出

max = 6, index = 1 (最大值 6 的下标是 1)

max = 10, index = 0 (最大值 10 的下标是 0)

max = 5, index = 4      (最大值 5 的下标是 4)

```
#include <stdio.h>
int main(void)
{
    int i, index, n;
    int ri, repeat;
    int a[10];

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++) {
        scanf("%d", &n);
        for(i = 0; i < n; i++)
            scanf("%d", &a[i]);
/*-----*/
        index=0;

        for(i=1; i<n; i++)

            if(a[i]>a[index]) index=i;

        printf("max = %d, index = %d\n", a[index], index);
    }
}
```

70015 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个正整数 n (1<n<=10)，再输入 n 个整数，将最小值与第一个数交换，  
最大值与最后一个数交换，然后输出交换后的 n 个数。

输入输出示例：括号内为说明

输入

3            (repeat=3)

5            (n=5)

8 2 5 1 4

4            (n=4)

1 5 6 7

5            (n=5)

5 4 3 2 1

输出

After swap: 1 2 5 4 8

After swap: 1 5 6 7

After swap: 1 4 3 2 5

```
#include <stdio.h>
int main(void)
{
    int i, index, n, t;
    int repeat, ri;
    int a[10];

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++) {
        scanf("%d", &n);
        for(i = 0; i < n; i++)
            scanf("%d", &a[i]);
/*-----*/
        index=0;

        for(i=1; i<n; i++)
            if(a[i]<a[index]) index=i;

        t=a[0];a[0]=a[index];a[index]=t;

        index=0;

        for(i=1; i<n; i++)
            if(a[i]>a[index]) index=i;

        t=a[n-1];a[n-1]=a[index];a[index]=t;

        printf("After swap: ");
        for(i = 0; i < n; i++)
            printf("%d ", a[i]);
        printf("\n");
    }
}
```

70016 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个正整数 n (1<n<=10)，再输入 n 个整数，将它们从大到小排序后输出。

输入输出示例：括号内为说明

输入

3 (repeat=3)

4 (n=4)

5 1 7 6

3 (n=3)

1 2 3

5 (n=5)

5 4 3 2 1

输出

After sorted: 7 6 5 1

After sorted: 3 2 1

After sorted: 5 4 3 2 1

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

```
int main(void)
```

```
{
```

```
    int i, index, k, n, temp;
```

```
    int repeat, ri;
```

```
    int a[10];
```

```
    scanf("%d", &repeat);
```

```
    for(ri = 1; ri <= repeat; ri++){
```

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

```
        for(i = 0; i < n; i++)
```

```
            scanf("%d", &a[i]);
```

```
/*-----*/
```

```
    for(i=0;i<n-1;i++)
```

```
    { index=i;
```

```
        for(k=i+1;k<n;k++)
```

```
            if(a[k]>a[index]) index=k;
```

```
            temp=a[i];a[i]= a[index]; a[index]=temp;
```

```
    }
```

```
    printf("After sorted: ");
```



```

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

```

70017 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个正整数 n (1<n<=10)，再输入 n 个整数，将它们存入数组 a 中，再输入一个整数 x，然后在数组 a 中查找与 x 相同的元素，如果找到，输出 x 在数组 a 中对应元素的最小下标，如果没有找到，输出相应信息。

输入输出示例：括号内为说明

输入：

```

2                (repeat=2)
6                (n=6)
1 3 5 7 9 5
5                (x=5)
4                (n=4)
1 3 5 7
2                (x=2)

```

输出：

```

5: a[2]
2: not found

```

```

#include <stdio.h>
int main(void)
{
    int flag, index, i, n, x;
    int repeat, ri;
    int a[10];

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &n);
        for(i = 0; i < n; i++)
            scanf("%d", &a[i]);
        scanf("%d", &x);
        /*-----*/

flag=0;

for(i=0; i<n; i++)

```

**if(a[i]==x) {index=i;flag=1;break;}**

```
if(flag != 0)
    printf( "%d: a[%d]\n", x, index);
else
    printf( "%d: not found\n", x);
```

```
}
```

```
}
```

70021 程序填空，不要改变与输入输出有关的语句。

输入 2 个正整数 m 和 n ( $1 \leq m \leq 6$ ,  $1 \leq n \leq 6$ )，然后输入矩阵 a (m 行 n 列) 中的元素，分别求出各行元素之和，并输出。

输出使用语句：printf("sum of row %d is %d\n", i, sum);

输入输出示例：括号内为说明

输入：

3 2 (m=3, n=2)

6 3

1 -8

3 12

输出：

sum of row 0 is 9

sum of row 1 is -7

sum of row 2 is 15

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

```
int main(void)
```

```
{
```

```
    int i, j, m, n, sum;
```

```
    int a[6][6];
```

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

```
    for(i = 0; i < m; i++)
```

```
        for(j = 0; j < n; j++)
```

```
            scanf("%d", &a[i][j]);
```

```
/*-----*/
```

```
    for(i = 0; i < m; i++){
```

```
        sum=0;
```

```
        for(j = 0; j < n; j++)
```

```

        sum=sum+a[i][j];

        printf("sum of row %d is %d\n", i, sum);

    }
}

```

}

70022 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat ( $0 < \text{repeat} < 10$ )，做 repeat 次下列运算：

读入 1 个正整数  $n$  ( $1 \leq n \leq 6$ )，再读入  $n$  阶方阵  $a$ ，计算该矩阵除副对角线、最后一列和最后一行以外的所有元素之和。副对角线为从矩阵的右上角至左下角的连线。

输入输出示例：括号内为说明

输入：

```

1          (repeat=1)
4          (n=4)
2 3 4 1
5 6 1 1
7 1 8 1
1 1 1 1
sum = 35   (2+3+4+5+6+7+8=35)

```

```

#include "stdio.h"
int main(void)
{
    int i, j, n, sum;
    int repeat, ri;
    int a[6][6];

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &n);
        for(i = 0; i < n; i++)
            for(j = 0; j < n; j++)
                scanf("%d", &a[i][j]);
        /*-----*/

        sum=0;

        for(i=0;i<n;i++){

            for(j=0;j<n;j++){

                if(i+j==n-1) continue;

```

```

        if(i==n-1) continue;

        if(j==n-1) continue;

        sum+=a[i][j];

    }

}

printf("sum = %d\n", sum);
}

return 0;
}

```

70023 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个正整数 n(0<n<10)，输出九九乘法表的前 n×n 项。

提示：将乘数、被乘数和乘积放入一个二维数组中，再输出该数组。

输入输出示例：括号内为说明

输入：

1 (repeat=1)

3 (n=3)

输出：

\* 1 2 3

1 1

2 2 4

3 3 6 9

```

#include "stdio.h"
int main(void)
{
    int i, j, n;
    int a[10][10];
    int repeat, ri;

    scanf("%d",&repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &n);
        /*-----*/

        for(i=0;i<=n;i++){

```

```

        a[i][0]=i;

        a[0][i]=i;

    }

    for(i=1;i<=n;i++)

        for(j=1;j<=n;j++)

            a[i][j]=a[0][j]*a[i][0];

    for(i = 0; i <= n; i++) {
        for(j = 0; j <= n; j++)
            if(i == 0 && j == 0) printf("%-4c", '*');
            else if(i == 0 || j <= i) printf("%-4d", a[i][j]);
        printf("\n");
    }
}

```

70024 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat ( $0 < \text{repeat} < 10$ )，做 repeat 次下列运算：

输入 1 个正整数 n ( $1 \leq n \leq 6$ ) 和 n 阶方阵 a 中的元素，如果 a 是上三角矩阵，输出"YES"，否则，输出"NO"。上三角矩阵指主对角线以下的元素都为 0 的矩阵，主对角线为从矩阵的左上角至右下角的连线。

输入输出示例：括号内为说明

输入：

2 (repeat=2)

3 (n=3)

1 2 3

0 4 5

0 0 6

2 (n=2)

1 0

-8 2

输出：

YES

NO

```
#include "stdio.h"
```

```
#include "math.h"
```

```

int main(void)
{
    int flag, i, j, n;
    int a[6][6];
    int repeat, ri;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &n);
        for(i = 0; i < n; i++)
            for(j = 0; j < n; j++)
                scanf("%d", &a[i][j]);
/*-----*/

        flag=1;

        for(i=0;i<n;i++)

            for(j=0;j<n;j++)

                if(i>j) if(a[i][j]!=0) {flag=0;break;}

        if(flag != 0) printf("YES\n");
        else printf("NO\n");
    }
}

```

70025 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入日期(年、月、日)，输出它是该年的第几天。

要求定义与调用函数 day\_of\_year(year, month, day)，计算并返回 year(年)、month(月)和 day(日)对应的是该年的第几天，函数形参 year、month 和 day 的类型是 int，函数类型也是 int。

输入输出示例：括号内为说明

输入：

2 (repeat=2)

1981 3 1 (1981 年 3 月 1 日)

2000 3 1 (2000 年 3 月 1 日)

输出：

days of year: 60 (1981 年 3 月 1 日是该年的第 60 天)

days of year: 61 (2000 年 3 月 1 日是该年的第 61 天)

```

#include "stdio.h"
int main(void)
{
    int day, day_year, month, year;
    int repeat, ri;

    int day_of_year(int year, int month, int day);

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d%d%d", &year, &month, &day);
        /*-----*/
        day_year= day_of_year(year,month,day);
        printf("days of year: %d\n", day_year);
    }
}

/*-----*/

int day_of_year(int year, int month, int day)
{ int k, leap;
int tab[2][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}
};
leap    =    (year%4==0&&year%100!=0)    ||
year %400==0;
for(k=1; k<month; k++)
    day = day + tab[leap][k];
return day;
}

```

70031 程序填空，不要改变与输入输出有关的语句。

输入一个以回车结束的字符串 (少于 80 个字符)，将它的内容逆序输出。如"ABCD"的逆序为"DCBA"。

输入输出示例：括号内为说明

输入：

Welcome to you!

输出：

!uoy ot emocleW

```
#include <stdio.h>
int main(void)
{
    int i, j, temp;
    char str[80];

    i = 0;
    while((str[i] = getchar( )) != '\n')
        i++;
    str[i] = 0;
    /*-----*/

    j=i;

    for(i = 0; i<j/2; i++)

        { temp=str[i];str[i]=str[j-1-i]; str[j-1-i]=temp;}

    for(i = 0; str[i] != 0; i++)
        putchar(str[i]);
}
```

70032

程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个字符，再输入一个以回车结束的字符串 (少于 80 个字符)，在字符串中查找该字符，如果找到，输出该字符在字符串中所对应的最大下标 (下标从 0 开始)；否则输出"Not Found"。

输入输出示例：括号内为说明

输入：

2 (repeat=2)

m (字符'm')

programming (字符串"programming")

a (字符'a')



1234                   (字符串"1234")  
输出:  
index = 7           ('m' 在"programming"中对应的最大下标是 7)  
Not Found           ("1234"中没有'a')

```
#include "stdio.h"
#define MAXLEN 80
int main(void)
{
    char cc, ch;
    char str[MAXLEN];
    int count, flag, i, index;
    int repeat, ri;

    scanf("%d", &repeat);
    getchar();
    for(ri = 1; ri <= repeat; ri++){
        cc = getchar();
        getchar();

        i = 0;
        while((ch = getchar()) != '\n'){
            str[i++] = ch;
        }
        str[i] = 0;
/*-----*/

        flag=0;

        for(i=0;str[i]!='\0';i ++)

            if(str[i]==cc) {index=i;flag=1;}

        if(flag != 0)
            printf("index = %d\n", index);
        else
            printf("Not Found\n");
    }
}
```

70033 程序填空，不要改变与输入输出有关的语句。  
输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个以回车结束的字符串（少于 80 个字符），统计并输出其中大写辅音字母的个数。大写辅音字母：除'A'，'E'，'I'，'O'，'U'以外的大写字母。

输入输出示例：括号内为说明

输入：

2 (repeat=2)

HELLO

group

输出：

count = 3 ("HELLO"中有 3 个大写辅音字母)

count = 0 ("group"中没有大写辅音字母)

```
#include "stdio.h"
```

```
#define MAXLEN 80
```

```
int main(void)
```

```
{
```

```
    char ch;
```

```
    char str[MAXLEN];
```

```
    int count, i;
```

```
    int repeat, ri;
```

```
    scanf("%d", &repeat);
```

```
    getchar();
```

```
    for(ri = 1; ri <= repeat; ri++){
```

```
        i = 0;
```

```
        while((ch = getchar()) != '\n'){
```

```
            str[i++] = ch;
```

```
        }
```

```
        str[i] = 0;
```

```
/*-----*/
```

```
    count=0;i=0;
```

```
    while(str[i]!='\0'){
```

```
        if(str[i]>='A'&&str[i]<='Z')
```

```
            if(str[i]!='A'&&str[i]!='E'&&str[i]!='I'&&str[i]!='O' &&
```

```
str[i]!='U')
```

```
                count++;
```

```

        i++;
    }

    printf("count = %d\n", count);
}
}

```

70034 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个以回车结束的字符串（少于 80 个字符），将其中的大写字母用下面列出的对应大写字母替换，其余字符不变，输出替换后的字符串。

原字母    对应字母

A            Z

B            Y

C            X

D            W

.....

X            C

Y            B

Z            A

输入输出示例：括号内为说明

输入：

2            (repeat=2)

A flag of USA

1+2=3

输出：

After replaced: Z flag of FHZ

After replaced: 1+2=3

```

#include "stdio.h"
#include "string.h"
#define MAXLEN 80
int main(void)
{
    char ch;
    char str[MAXLEN];
    int i;
    int repeat, ri;

    scanf("%d", &repeat);

```

```

    getchar();
    for(ri = 1; ri <= repeat; ri++){
        i = 0;
        while((ch = getchar()) != '\n'){
            str[i] = ch;
            i++;
        }
        str[i] = 0;
    }
    /*-----*/

```

**i=0;**

**while(str[i]!='\0'){**

**if(str[i]>='A'&&str[i]<='Z')**

**str[i]='Z'-str[i]+'A';**

**i++;**

**}**

```

printf("After replaced: ");
for(i = 0; str[i] != 0; i++)
    putchar(str[i]);
putchar('\n');

```

```

    }
}

```

70035 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个以回车结束的字符串（少于 80 个字符），滤去所有的非十六进制字符后，组成一个新字符串（十六进制形式），然后将其转换为十进制数后输出。

输入输出示例：括号内为说明

输入：

2 (repeat=2)

10

+A

输出：

Dec = 16

Dec = 10

```

#include "stdio.h"
#include "string.h"
#define MAXLEN 80
int main(void)
{
    char ch;
    char str[MAXLEN], num[MAXLEN];

    int i, j, k;    //加 j
    int repeat, ri;
    long number;

    scanf("%d", &repeat);
    getchar();
    for(ri = 1; ri <= repeat; ri++){
        i = 0;
        while((ch = getchar()) != '\n'){
            str[i]=ch;
            i++;
        }
        str[i] = 0;

/*-----*/

        i=j=0;

        while(str[i]!='\0'){

if((str[i]>='A'&&str[i]<='F')||(str[i]>='a'&&str[i]<='f') ||
(str[i]>='0'&&str[i]<='9'))
            num[j++]=str[i];
            i++;
}
        num[j]='\0';
        number=0;
        for(i=0;num[i]!='\0';i++)

```

```

{
    if(num[i]>='0'&&num[i]<='9')
        number=number*16+(num[i]-'0');
    if(num[i]>='A'&&num[i]<='F')
        number=number*16+10+num[i]-'A';
    if(num[i]>='a'&&num[i]<='f')
        number=number*16+10+num[i]-'a';
}

    printf("Dec = %ld\n", number);
}
}

```

70036 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个以回车结束的字符串（少于 80 个字符），把字符串中的所有数字字符（'0'～'9'）转换为整数，去掉其他字符。例如，字符串"3a56bc"转换为整数后是 356。

输入输出示例：括号内为说明

输入：

2 (repeat=2)

82jeep5

free

输出：

825

0

```

#include <stdio.h>
#include <string.h>
int main(void)
{
    char str[80];
    int i, s;
    int repeat, ri;

```

```

scanf("%d", &repeat);
getchar();
for(ri = 1; ri <= repeat; ri++){
    i = 0;
    while((str[i] = getchar( )) != '\n')
        i++;
    str[i] = 0;
}
/*-----*/

```

**s= 0;**

**for(i = 0; str[i] != '\0'; i++)**

**if(str[i] <= '9' && str[i] >= '0')**

**s = s \* 10 + (str[i] - '0');**

```

printf("%d\n", s);
}

```

}

80011 程序填空，不要改变与输入输出有关的语句。

输入两个正整数 n 和 m ( $1 < m < n \leq 10$ )，再输入 n 个整数，将这些数排成一行，向右循环移动 m 个位置（从右边移出的数再从左边移入），最后输出移动后的 n 个整数。

要求定义并调用函数 mov(x, n, m) 实现上述循环移动的功能，函数形参 x 的类型是整型指针，形参 n 和 m 的类型是 int，函数的类型是 void。

输入输出示例：括号内为说明

输入：

5 3 (n=5, m=3)

1 2 3 4 5 (5 个整数)

输出：

After move: 3 4 5 1 2

```

#include <stdio.h>
void mov(int *x, int n, int m);
int main(void)
{
    int i, m, n;
    int a[80];

```

```

scanf("%d%d", &n, &m);
for(i = 0; i < n; i++)
    scanf("%d", &a[i]);
/*-----*/

mov(a,n,m);

printf("After move: ");
for(i = 0; i < n; i++)
    printf("%d ", a[i]);
printf("\n");
}

```

```
/*-----*/
```

**void mov(int \*x, int n, int m)**

```

{   int i,j,k;

    for( i=0;i<m;i++) {

        k=x[n-1];

        for(j=n-1 ; j>0; j--)    x[j]=x[j-1];

        x[0]=k;

    }

}

```

80012 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个正整数 n (1<n<=10)，然后输入 n 个整数存入数组 a 中，再输入一个整数 x，在数组 a 中查找 x，如果找到则输出相应元素的最小下标，否则输出"Not found"。

要求定义并调用函数 search(list, n, x)，它的功能是在数组 list 中查找元素 x，若找到则返回相应元素的最小下标，否则返回-1，函数形参 list 的类型是整型指针，形参 n 和 x 的类型是 int，函数的类型是 void。

输入输出示例：括号内为说明

输入

2 (repeat=2)

3 (n=3)

1 2 -6

2 (x=2)



5 (n=5)  
1 2 2 5 4  
0 (x=0)  
输出  
index = 1  
Not found

```
#include <stdio.h>
int main(void)
{
    int i, index, n, res, x;
    int repeat, ri;
    int a[10];

    int search(int list[], int n, int x);

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++) {
        scanf("%d", &n);
        for(i = 0; i < n; i++)
            scanf("%d", &a[i]);
        scanf("%d", &x);
        /*-----*/

        res=search(a,n,x);

        if(res != -1)
            printf("index = %d\n", res);
        else
            printf("Not found\n");
    }
}
/*-----*/
```

**int search(int list[], int n, int x)**

```
{ int i;

for(i=0; i<n; i++)

if(list[i]==x) return i;

return -1;
```

}

80013 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat ( $0 < \text{repeat} < 10$ )，做 repeat 次下列运算：

输入一个正整数 n ( $1 < n \leq 10$ )，再输入 n 个整数存入数组 a 中，用选择法将数组 a 中的元素按升序排列，最后输出排序后的数组元素。

要求定义并调用函数 sort(a, n)，它的功能是采用选择排序算法，将数组 a 的前 n 个元素按从小到大的顺序排序，函数形参 a 的类型是整型指针，形参 n 的类型是 int，函数的类型是 void。

输入输出示例：括号内为说明

输入

3 (repeat=3)

4 (n=4)

5 1 7 6

3 (n=3)

1 2 3

5 (n=5)

5 4 3 2 1

输出

After sorted: 1 5 6 7

After sorted: 1 2 3

After sorted: 1 2 3 4 5

```
#include <stdio.h>
void sort(int a[], int n);
int main(void)
{
    int i, n;
    int repeat, ri;
    int a[10];

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &n);
        for(i = 0; i < n; i++)
            scanf("%d", &a[i]);
    }
    /*-----*/
```

**sort(a,n);**

printf("After sorted: ");

```

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

/*-----*/

void sort(int a[],int n)
{ int i,k,index,temp;
    for(i=0;i<n-1;i++)
    { index=i;
        for(k=i+1;k<n;k++)
            if(a[k]<a[index]) index=k;
                temp=a[i];a[i]= a[index]; a[index]=temp;
        }
    }

```

80014 程序填空，不要改变与输入输出有关的语句。

输入两个正整数  $n$  和  $m$  ( $1 < m < n \leq 50$ )，有  $n$  个人围成一圈，按顺序从 1 到  $n$  编号。从第一个人开始报数，报数  $m$  的人退出圈子，下一个人从 1 开始重新报数，报数  $m$  的人退出圈子。如此循环，直到留下最后一个人。请按退出顺序输出退出圈子的人的编号，以及最后一个人的编号。

提示：将每个人的编号存入数组，从第一个人开始报数，输出报数  $m$  的人的编号，并将该编号清除为 0，重复这样的操作直至只剩下一个不为 0 的数，该数就是最后一个人的编号。

输出使用语句：printf("No%d: %d\n", no, \*p);

输入输出示例：括号内为说明

输入：

5 (n 个人报数，n=5)

3 (报数 m=3)

输出：

No1: 3 (第 1 个退出圈子的人编号是 3)

No2: 1 (第 2 个退出圈子的人编号是 1)

No3: 5 (第 3 个退出圈子的人编号是 5)

No4: 2 (第 4 个退出圈子的人编号是 2)

Last No is: 4 (最后一个人的编号是 4)

```

#include<stdio.h>
int main(void)
{
    int count, i, m, n, no, k;
    int num[50];
    int *p;

    scanf("%d%d", &n, &m);
    for(i = 0; i < n; i++)
        num[i] = i + 1;
    p = num;
    /*-----*/

        no=1;

        while(no<n)
        {
            k=0;

            while(1)
            {
                if(*p!=0){
                    k++;

                    if(k==m)
                        break;
                }

                p++;

                if(p==num+n)
                    p=num;
            }
        }

```

```

        }

        printf("No%d: %d\n",no,*p);

        *p=0;

        no++;

    }

    p = num;
    while(*p == 0)
        p++;
    printf("Last No is: %d\n", *p);
}

```

80021 程序填空，不要改变与输入输出有关的语句。

输入 5 个字符串，输出其中最大的字符串。

输入输出示例：括号内为说明

输入：

peach

pear

melon

orange

berry

输出：

Max is: pear

```

#include <stdio.h>
#include <string.h>
int main(void)
{
    char str[80], max[80];
    int i;

    scanf("%s", str);
    /*-----*/

```

**strcpy(max,str);**

```

for(i = 1; i < 5; i++){
    scanf("%s", str);
    if(strcmp(str,max)>0)
        strcpy(max,str);
}
    printf("Max is: %s\n", max);
}

```

80022 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个正整数 n (1<n<=10)，然后输入 n 个字符串，输出其中最长的字符串，  
如果字符串的长度相同，则输出先输入的字符串。

输入输出示例：括号内为说明

输入：

```

3                (repeat=3)
5                (n=5)
melon peach pear strawberry orange
3                (n=3)
pear berry orange
4                (n=4)
melon peach pear apple

```

输出：

```

The longest is: strawberry
The longest is: orange
The longest is: melon

```

```

#include <stdio.h>
#include <string.h>
int main(void)
{
    char sx[80], longest[80];
    int i, n;
    int repeat, ri;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d", &n);

```

```

scanf("%s", sx);
/*-----*/

strcpy(longest ,sx);

for(i = 1; i < n; i++){

    scanf("%s", sx);

    if(strlen(sx)>strlen(longest))

        strcpy(longest ,sx);

}

printf("The longest is: %s\n", longest);
}

```

80023 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个字符串 str，再输入一个字符 c，将字符串 str 中出现的所有字符 c 删除。

要求定义并调用函数 delchar(str,c)，它的功能是将字符串 str 中出现的所有 c 字符删除，函数形参 str 的类型是字符指针，形参 c 的类型是 char，函数类型是 void。

输入输出示例：括号内为说明

输入

```

3                (repeat=3)
happy new year  (字符串"happy new year")
a                (待删除的字符' a' )
bee              (字符串"bee")
e                (待删除的字符' e' )
111211           (字符串"111211")
1                (待删除的字符' 1' )

```

输出

```

result: hppy new yer    (字符串"happy new year"中的字符' a' 都被删除)
result: b                (字符串"bee"中的字符' e' 都被删除)
result: 2                (字符串"111211"中的字符' 1' 都被删除)

```

```

#include<stdio.h>
void main()
{

```

```

char c;
char str[80];
int repeat, ri;
void delchar(char *str, char c);

scanf("%d", &repeat);
getchar();
for(ri = 1; ri <= repeat; ri++){
    gets(str);
    scanf("%c", &c);
    getchar();
}
/*-----*/

    delchar(str,c);
    printf("result: ");
    puts(str);
}
}

/*-----*/

void delchar(char *str, char c)
{ char *p,*q;
    p=str;
    while(*p!='\0')
    { if(*p==c)
        { q=p;
            while(*q!='\0')
                { *q=*(q+1); q++;}
                *q='\0';
            }
            if(*p!=c) p++;
        }
    }

```



}

80024 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat ( $0 < \text{repeat} < 10$ )，做 repeat 次下列运算：

输入一个字符串 t 和一个正整数 m，将字符串 t 中从第 m 个字符开始的全部字符复制到字符串 s 中，再输出字符串 s。

要求定义并调用函数 strmcpy(s,t,m)，它的功能是将字符串 t 中从第 m 个字符开始的全部字符复制到字符串 s 中，函数形参 s 和 t 的类型是字符指针，形参 m 的类型是 int，函数类型是 void。

输入输出示例：括号内为说明

输入：

3 (repeat=3)

happy new year

7

happy

1

new

4

输出：

new year (从“happy new year”第 7 个字符开始组成的新字符串为“new year”)

happy (从“happy”第 1 个字符开始组成的新字符串为“happy”)

error input (“new”的长度小于 4)

```
#include<stdio.h>
#include<string.h>
void main()
{
    char s[80], t[80];
    int m;
    int repeat, ri;
    void strmcpy(char *s, char *t, int m);

    scanf("%d", &repeat);
    getchar();
    for(ri = 1; ri <= repeat; ri++){
        gets(t);
        scanf("%d", &m);
        getchar();
        if(strlen(t) < m)
```

```

        printf("error input");
    else{

        strncpy(s,t,m);

        /*-----*/
        puts(s);
    }
}

/*-----*/

void strncpy(char *s, char *t, int m)
{
    int i;

    for(i=m-1;t[i]!='\0';i++)

        s[i-m+1]=t[i];

        s[i-m+1]='\0';

}

```

80025 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个字符串(长度不超过 80)，判断该字符串是否为“回文”。“回文”是指顺读和倒读都一样的字符串，如“XYZYX”和“xyzzyx”。

要求定义并调用函数 mirror(p)判断字符串 p 是否为“回文”，如果 p 是回文字符串，返回 1，否则，返回 0，函数形参 p 的类型是字符指针，函数类型是 int。

输入输出示例：括号内为说明

输入：

2 (repeat=2)

abcedcba

abcedcb

输出：

YES

NO

```

#include <stdio.h>
void main()

```

```

{
    char s[80];
    int repeat, ri;
    int mirror(char *p);

    scanf("%d", &repeat);
    getchar();
    for(ri = 1; ri <= repeat; ri++){
        gets(s);
        if(mirror(s) != 0)
            printf("YES\n");
        else
            printf("NO\n");
    }
}

```

/\*-----\*/

```

int mirror(char *p)
{    int i,length,k;
        for(i = 0; p[i]!='\0'; i++);
        length=i;
        for(k = 0; k < length/2; k++)
            if(p[k]!= p[length-1-k])
                return 0;
        return 1;
}

```

或

```

int mirror(char *p)
{
    int i,j,n,count;
    n=strlen(p);

```

```

        count=0;

        for(i=0,j=n-1;i<j;i++,j--){

            if(*(p+i)!=*(p+j))

                return 0;

            return 1;

        }

    }

```

80026 程序填空，不要改变与输入输出有关的语句。

输入一行字符(不超过 80 个)，统计其中的大写字母、小写字母、空格、数字以及其他字符的个数。

输入输出示例：

输入：

bFaE3+8 =1R

输出：

upper: 3 lower: 2 blank: 1 digit: 3 other: 2

```

#include<stdio.h>
void main()
{
    char s[80];
    char *p;
    int blank, digit, lower, other, upper;

    gets(s);
    upper = lower = blank = digit = other = 0;

    /*-----*/

    for(p=s;*p!='\0';p++){

        if(*p >='A' && *p <='Z') upper++;

        else if(*p >='a' && *p <='z')  lowerr++;

        else if(*p >='0' && *p <='9')  digit++;

```

```

else if(*p == ' ') blank++;
else other++;
}

```

```

printf("upper: %d lower: %d blank: %d digit: %d other: %d\n",
upper, lower, blank, digit, other);
}

```

90001 程序填空，不要改变与输入输出有关的语句。

输入一个正整数  $n$  ( $3 \leq n \leq 10$ )，再输入  $n$  个雇员的信息，包括姓名、基本工资、浮动工资和支出，输出每人的姓名和实发工资，实发工资=基本工资+浮动工资-支出。

输入输出示例：括号内为说明

输入：

3 (n=3)

zhao 240 400 75

qian 360 120 50

zhou 560 0 80

输出：

zhao 实发工资: 565.00

qian 实发工资: 430.00

zhou 实发工资: 480.00

```

#include <stdio.h>
int main (void)
{
    int i, n;
    double x;
    struct emp{
        char name[10];
        double jbg;
        double fdg;
        double zc;
    } s[10];

```

```

    scanf("%d", &n);
    /*-----*/

```

```

for(i=0;i<n;i++)

```

```

scanf("%s%lf%lf%lf",s[i].name,&s[i].jbg,&s[i].fdg,&
s[i].zc);
    for (i = 0; i < n; i++)
        printf ("%5s 实发工资: %7.2f\n", s[i].name, s[i].jbg + s[i].fdg
- s[i].zc);
}

```

90002 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个时间数值，再输入秒数 n，输出该时间再过 n 秒后的时间值，时间的表示形式为时:分:秒，超过 24 时从 0 时重新开始计时。

输入输出示例：括号内为说明

输入：

```

3          (repeat=3)
0:0:1
59         (秒数 n=59)
11:59:40
30         (秒数 n=30)
23:59:40
301        (秒数 n=301)

```

输出：

```

time: 0:1:0      (0:0:01 加上 59 秒的新时间)
time: 12:0:10    (11:59:40 加上 30 秒的新时间)
time: 0:4:41     (23:59:40 加上 301 秒的新时间)

```

```

#include <stdio.h>
int main(void)
{
    int n;
    int repeat, ri;
    struct time{
        int hour, minute, second;
    }time;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d:%d:%d", &time.hour, &time.minute, &time.second);
        scanf("%d",&n);
        /*-----*/

```

```

        time.second=time.second+n;

```

```

        if(time.second>=60){
            time.minute=time.minute+time.second/60;
            time.second=time.second%60;
            if(time.minute>=60){
                time.hour=time.hour+time.minute/60;
                time.minute=time.minute%60;
                if(time.hour>=24)
                    time.hour=time.hour%24;
            }
        }
        printf("time: %d:%d:%d\n", time.hour, time.minute,
time.second);
    }
}

```

90003 程序填空，不要改变与输入输出有关的语句。

输入整数 n(n<10)，再输入 n 个学生的基本信息，包括序号、姓名和成绩，要求计算并输出他们的平均成绩(保留 2 位小数)。

输入输出示例：括号内为说明

输入：

```

3          (n=3)
1  zhang  70
2  wang   80
3  qian   90

```

输出：

```

average: 80.00

```

```

#include <stdio.h>
int main(void)
{
    int i, n;
    double average, sum;

```

```

    struct student{
        int num;
        char name[10];
        int score;
    }s[10];

    scanf("%d", &n);
    /*-----*/

    sum=0;

    for(i=0;i<n;i++){

scanf("%d%s%d",&s[i].num,s[i].name,&s[i].score);

sum=sum+s[i].score;

average=sum/n;

}

    printf("average: %.2f\n", average);
}

```

90004 程序填空，不要改变与输入输出有关的语句。

输入 4 个整数 a1, b1, a2, b2，分别表示两个复数的实部与虚部，求两个复数之积  $(a1+b1i)*(a2+b2i)$ ，乘积的实部为：  $a1*a2-b1*b2$ ，虚部为：  $a1*b2+a2*b1$ 。

输入输出示例：括号内为说明

输入：

3 4 5 6

输出：

$(3+4i) * (5+6i) = -9 + 38i$

```

#include <stdio.h>
int main(void)
{
    struct complex{
        int real;
        int imag;
    }product, x, y;

```



```

scanf("%d%d%d%d", &x.real, &x.imag, &y.real, &y.imag);
/*-----*/

product.real=x.real*y.real-x.imag*y.imag;

product.imag=x.real*y.imag+x.imag*y.real;

printf("(%d+%di) * (%d+%di) = %d + %di\n", x.real, x.imag, y.real,
y.imag, product.real, product.imag);
}

```

90005 程序填空，不要改变与输入输出有关的语句。

编写程序，从键盘输入 n (n<10) 本书的名称和定价并存入结构数组中，查找并输出其中定价最高和最低的书的名称和定价。

输入输出示例：括号内为说明

输入：

3 (n=3)

Programming in C

21.5

Programming in VB

18.5

Programming in Delphi

25.0

输出：

highest price: 25.0, Programming in Delphi

lowest price: 18.5, Programming in VB

```

#include <stdio.h>
int main (void)
{
    int i, max_index, min_index, n;
    double x;
    struct book{
        char name[50];
        double price;
    }book[10];

    scanf("%d", &n);
    getchar();
    for(i = 0; i < n; i++){
        gets(book[i].name);
        scanf("%lf", &x);
        getchar();
    }
}

```

```

        book[i].price = x;
    }
/*-----*/

    max_index=0;

    min_index=0;

    for(i=1;i<n;i++){

        if(book[i].price>book[max_index].price)

            max_index=i;

        if(book[i].price<book[min_index].price)

            min_index=i;

    }

    printf("highest price: %.1f, %s\n", book[max_index].price,
book[max_index].name);
    printf("lowest price: %.1f, %s\n", book[min_index].price,
book[min_index].name);
}

```

90006 程序填空，不要改变与输入输出有关的语句。

输入 n(n<10) 个朋友的信息，包括姓名、生日、电话号码，按照年龄从大到小的顺序依次输出通讯录。

输入输出示例：括号内为说明

输入：

3           (n=3)

zhang 19850403 13912345678

wang 19821020 0571-88018448

qian 19840619 13609876543

输出：

wang 19821020 0571-88018448

qian 19840619 13609876543

zhang 19850403 13912345678

```

#include <stdio.h>
int main (void)
{
    int i, index, j, n;
    struct address_list{
        char name[20];
        long birthday;
        char phone[20];
    }temp, friends[10];

    scanf("%d", &n);
    for(i = 0; i < n; i++)
        scanf("%s%ld%s", friends[i].name, &friends[i].birthday,
friends[i].phone);
    /*-----*/

        for(i=0;i<n-1;i++){

            index=i;

            for(j=i+1;j<n;j++)

                if(friends[j].birthday<friends[index].birth
day){

                    temp=friends[index];

                    friends[index]=friends[j];

                    friends[j]=temp;

                }

            }

        for(i = 0; i < n; i++)
            printf("%s %ld %s\n", friends[i].name, friends[i].birthday,
friends[i].phone);
}

```

90007 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

编写程序，输入一个日期(年、月、日)，计算并输出该日是该年中的第几天。

要求定义并调用函数 `day_of_year(p)` 计算某日是该年的第几天，函数形参 `p` 的类型是结构指针，指向表示日期的结构变量，函数类型是 `int`，注意区分闰年。

输入输出示例：括号内为说明

输入：

2 (repeat=2)

2006 3 5

2000 3 5

输出：

64 (2006 年 3 月 5 日是该年的第 64 天)

65 (2000 年 3 月 5 日是该年的第 65 天)

```
#include <stdio.h>
struct date{
    int year;
    int month;
    int day;
};
int day_of_year(struct date *p);

int main(void)
{
    int yearday;
    int repeat, ri;
    struct date date;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d%d%d", &date.year, &date.month, &date.day);
        /*-----*/
        yearday=day_of_year(&date);
        printf("%d\n", yearday);
    }
}

/*-----*/

int day_of_year(struct date *p)
{
    int k,leap,day;
```

```

        int
tab[2][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}};
        leap=(p->year%4==0&& p->year%100!=0||p->ye
ar%400==0);
        day=0;
        for(k=1;k<p->month;k++)
            day=day+tab[leap][k];
        day+=p->day;
        return day;
    }

```

90008 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个时间数值，再输入秒数 n，输出该时间再过 n 秒后的时间值，时间的表示形式为时:分:秒，超过 24 时从 0 时重新开始计时。

要求定义并调用函数 timecal(p, n)实现时间换算，函数形参 p 的类型是结构指针，指向表示时间的结构变量，形参 n 的类型是整型，表示秒数，函数类型是 void。

输入输出示例：括号内为说明

输入：

3 (repeat=3)

0:0:1

59 (秒数 n=59)

11:59:40

30 (秒数 n=30)

23:59:40

301 (秒数 n=301)

输出：

time: 0:1:0 (0:0:01 加上 59 秒的新时间)

time: 12:0:10 (11:59:40 加上 30 秒的新时间)

time: 0:4:41 (23:59:40 加上 301 秒的新时间)

```

#include <stdio.h>
struct time{
    int hour, minute, second;
};
void timecal(struct time *p, int n);

int main(void)
{
    int n;
    int repeat, ri;
    struct time time;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++){
        scanf("%d:%d:%d", &time.hour, &time.minute, &time.second);
        scanf("%d",&n);
    }
    /*-----*/

```

**timecal(&time,n);**

```

    printf("time: %d:%d:%d\n", time.hour, time.minute,
time.second);
}
}
/*-----*/

```

**void timecal(struct time \*p, int n)**

**{**

**p->second=p->second+n;**

**while(p->second>=60){**

**p->second=p->second-60;**

**p->minute++;**

**}**

**while(p->minute>=60){**

**p->minute=p->minute-60;**

```

        p->hour++;

    }

    if(p->hour>=24)p->hour=p->hour-24;

    return *p;

}

```

90009 程序填空，不要改变与输入输出有关的语句。

输入一个正整数  $n$  ( $3 < n \leq 10$ )，再输入  $n$  个学生的信息，包括学号、姓名和 3 门课程的成绩，输出总分最高学生的姓名和总分。

输入输出示例：括号内为说明

输入：

```

5          (n=5)
1 黄岚 78  83  75
2 王海 76  80  77
3 沈强 87  83  76
4 张枫 92  88  78
5 章盟 76  81  75

```

输出：

总分最高的学生是：张枫，258 分

```

#include <stdio.h>
int main (void)
{
    int i, j, k, max_index, n;
    struct student{
        int number;
        char name[20];
        int score[3];
        int sum;
    }stu[10];

    scanf("%d", &n);
    /*-----*/

    for(i=0;i<n;i++){

```

```

scanf("%d%s",&stu[i].number,stu[i].name);

stu[i].sum=0;

for(j=0;j<3;j++){

    scanf("%d",&stu[i].score[j]);

    stu[i].sum+=stu[i].score[j];

}

}

```

```

max_index=0;

for(i=1;i<n;i++){

    if(stu[i].sum>stu[max_index].sum)

        max_index=i;

}

```

```

printf("总分最高的学生是: %s, %d 分\n", stu[max_index].name,
stu[max_index].sum);
}

```

10012 程序填空，不要改变与输入输出有关的语句。

编写一个函数，利用参数传入一个 3 位数 number，找出 101~number 之间所有满足下列两个条件的数：

它是完全平方数，又有两位数字相同，如 144、676 等，函数返回找出这样的数据的个数。请同时编写主函数。

例：（括号内为说明）

输入

3           (repeat=3)

150

500

999

输出

count=2

count=6

count=8



```

#include <stdio.h>
#include <math.h>
int search(int n);
int main(void)
{
    int number,ri,repeat;

    scanf("%d",&repeat);
    for(ri=1;ri<=repeat;ri++) {
        do{
            scanf("%d",&number);
        }while(number<101||number>999);
        printf("count=%d\n",search(number));
    }
}
/*-----*/

int search(int n)
{

    int num1,num2,num3,num,count=0;
for(num=101;num<n;num++){
    num1=num/100;
    num2=(num%100)/10;
    num3=num%10;
    if(sqrt(num)==(int) sqrt(num))

if(num1==num2||num2==num3||num1==num3)
        count++;
}

```

**return count;**

**}**

10014 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个整数 n (n>=0) 和一个双精度浮点数 x，输出函数 p(n, x) 的值(保留 2 位小数)。

$$p(n, x) = \begin{cases} 1 & (n=0) \\ x & (n=1) \\ ((2*n-1)*p(n-1, x) - (n-1)*p(n-2, x))/n & (n>1) \end{cases}$$

例：括号内是说明

输入

3 (repeat=3)  
0 0.9 (n=0, x=0.9)  
1 -9.8 (n=1, x=-9.8)  
10 1.7 (n=10, x=1.7)

输出

p(0, 0.90)=1.00  
p(1, -9.80)=-9.80  
p(10, 1.70)=3.05

```
#include <stdio.h>
double p(int n, double x);
int main(void)
{
    int repeat, ri;
    int n;
    double x, result;

    scanf("%d", &repeat);
    for(ri = 1; ri <= repeat; ri++)
    {
        scanf("%d%lf", &n, &x);
        result = p(n, x);
        printf("p(%d, %.2lf)=%.2lf\n", n, x, result);
    }
}
/*-----*/
```

**double p(int n, double x)**

```

{
    double y;
    if(n==0)
        y=1;
    else if(n==1)
        y=x;
    else
        y=((2*n-1)*p(n-1,x)-(n-1)*p(n-2,x))/n;
    return y;
}

```

10015 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat ( $0 < \text{repeat} < 10$ )，做 repeat 次下列运算：

输入两个整数 m 和 n ( $m \geq 0$  且  $n \geq 0$ )，输出函数 Ack(m, n) 的值。

在  $m \geq 0$  和  $n \geq 0$  时 Ack(m, n) 定义为：

$\text{Ack}(0, n) = n + 1$

$\text{Ack}(m, 0) = \text{Ack}(m - 1, 1)$

$\text{Ack}(m, n) = \text{Ack}(m - 1, \text{Ack}(m, n - 1))$   $m > 0$  且  $n > 0$

例：括号内是说明

输入

4 (repeat=4)

0 3

2 0

2 3

0 0

输出

Ackerman(0, 3)=4

Ackerman(2, 0)=3

Ackerman(2, 3)=9

Ackerman(0, 0)=1

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

```

int Ack(int m, int n);
int main(void)
{
    int m,n;
    int result;
    int i,repeat;

    scanf("%d",&repeat);
    for(i=1; i<=repeat; i++)
    {
        scanf("%d%d", &m, &n);
        result = Ack(m,n);
        printf("Ackerman(%d,%d)=%d\n", m, n, result);
    }
}
/*-----*/

```

```

int Ack(int m, int n)
{

    int y;

    if(m==0)

        y=n+1;

    else if(n==0)

        y= Ack(m-1,1);

    else

        y=Ack(m-1, Ack(m,n-1));

    return y;

}

```

10016 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入 1 个正整数 n，将其转换为二进制后输出。要求定义并调用函数 dectobin(n)，它的功能是输出 n 的二进制。

例如，调用 dectobin(10)，输出 1010。

输出语句: printf("%d");

例: 括号内是说明

输入:

3 (repeat=3)

15

100

0

输出:

1111

1100100

0

```
#include "stdio.h"
int main(void)
{
    int ri, repeat;
    int i, n;
    void dectobin(int n);

    scanf("%d", &repeat);
    for(ri=1; ri<=repeat; ri++) {
        scanf("%d", &n);
        dectobin(n);
        printf("\n");
    }
}
/*-----*/
```

**void dectobin(int n)**

**{**

**if(n<2)**

**printf("%d", n);**

**else{**

**dectobin(n/2);**

**printf("%d", n%2);**

}

}

10017 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat ( $0 < \text{repeat} < 10$ )，做 repeat 次下列运算：

用递归方法编写求 Fibonacci 数列项的函数，返回值为长整型，并写出相应的主函数。Fibonacci 数列的定义为：

$f(n) = f(n-2) + f(n-1) \quad (n \geq 2)$  其中  $f(0) = 0, f(1) = 1$ 。

例：(括号内为说明)

输入

3 (repeat=3)

0

1

6

输出

fib(0)=0

fib(1)=1

fib(6)=8

```
#include <stdio.h>
long fib(int n);
int main(void)
{
    int n,ri,repeat;

    scanf("%d",&repeat);
    for(ri=1;ri<=repeat;ri++){
        scanf("%d",&n);
        printf("fib(%d)=%ld\n",n,fib(n));
    }
}
/*-----*/
```

**long fib(int n)**

**{**

**int y;**

```

if(n==1||n==2)
    y=1;
else if(n==0)
    y=0;
else
    y=fib(n-1)+fib(n-2);
return y;
}

```

10018 程序填空，不要改变与输入输出有关的语句。  
用递归方法实现对一个整数进行逆序输出。

例：（括号内为说明）

输入

3           (repeat=3)

567

911

8

输出

765

119

8

```

#include <stdio.h>
void reverse(int n);
int main(void)
{
    int n,ri,repeat;

    scanf("%d",&repeat);
    for(ri=1;ri<=repeat;ri++){
        scanf("%d",&n);
        reverse(n);
        printf("\n");
    }
}

```

```

}
/*-----*/

void reverse(int n)

{

    printf("%d",n%10);


    if((n/10)!=0)

        reverse(n/10);

}

```

10019 程序填空，不要改变与输入输出有关的语句。  
用递归函数计算  $x^n$  ( $n \geq 1$ ) 的值。

例：（括号内为说明）

输入

2 3      (x=2, n=3)

输出

Root = 8.00

```

#include <stdio.h>
double fun(int n, double x);
int main(void)
{
    int n;
    double x, root;

    scanf("%lf%d", &x,&n);
    root = fun(n, x);
    printf("Root = %0.2f\n", root);
}
/*-----*/

double fun(int n, double x)

{

    double r;

```



```

        if(n==0)

            r=1;

        else

            r=x*fun(n-1,x);

        return r;

    }

```

10021 程序填空，不要改变与输入输出有关的语句。

输入若干有关颜色的英文单词，以#作为输入结束标志，对这些单词升序排列后输出。其中颜色的英文单词数小于 20 个，颜色的英文单词长度均不超过 10 个字符。

输入输出示例：括号内为说明

输入：

```

red
blue
yellow
green
purple
#

```

输出：

```

blue green purple red yellow

```

```

#include <stdio.h>
#include<stdlib.h>
#include<string.h>
void main()
{
    int i,j, n = 0;
    char *color[20], str[10], *temp;

    scanf("%s", str);
    while(str[0] != '#') {
        color[n] = (char *)malloc(sizeof(char)*(strlen(str)+1));
        strcpy(color[n], str);
    }
}

```

```

        n++;
        scanf("%s", str);
    }
    /*-----*/

    for(i=1;i<n;i++)
        for(j=0;j<n-i;j++)
            if(strcmp(color[j],color[j+1])>0){
                temp=color[j];
                color[j]=color[j+1];
                color[j+1]=temp;
            }

    for(i = 0; i < n; i++)
        printf("%s ", color[i]);
    printf("\n");
}

```

10022 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

编写程序，输入一个月份，输出对应的英文名称，要求用指针数组表示 12 个月的英文名称。

若输入月份错误，输出提示信息。

输入输出示例：括号内为说明

输入：

3 (repeat=3)

5

9

14

输出：

May

September

Wrong input!

```

#include<stdio.h>
void main()
{

```

```

    int ri,repeat;
    int month;
    char
*month_name[]={ "", "January", "February", "March", "April", "May", "June", "
July", "August", "September", "October", "November", "December"};

    scanf("%d",&repeat);
    for(ri=1;ri<=repeat;ri++){
        scanf("%d",&month);
        /*-----*/

        if(month<=12)

            printf("%s\n",month_name[month]);

        else

            printf("Wrong input!\n");

    }
}

```

10023 程序填空，不要改变与输入输出有关的语句。

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

定义一个指针数组将下表的星期信息组织起来，输入一个字符串，在表中查找，若存在，输出该字符串在表中的序号，否则输出-1。

(表格详见实验教材 P99)

输入输出示例：括号内为说明

输入：

3 (repeat=3)

Tuesday

Wednesday

year

输出：

3

4

-1

```

#include<stdio.h>
#include<string.h>
void main()
{
    int i,ri,repeat;

```

```

char
*date[]={ "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday"};
char str[80];

scanf("%d",&repeat);
getchar();
for(ri=1;ri<=repeat;ri++){
    scanf("%s",str);
    /*-----*/

for(i=0;i<7;i++)

if(strcmp(date[i],str)==0){

printf("%d\n",i+1);

break;

}

if(i>=7)

printf("-1\n");

    }
}

```

10024 程序填空，不要改变与输入输出有关的语句。

编写一个函数 `int max_len(char *s[ ], int n)`，用于计算有  $n(n<10)$  个元素的指针数组 `s` 中最长的字符串的长度，并编写主程序验证。

例：（括号内为说明）

输入

4            (n=4)

blue

yellow

red

green

输出

length=6

```

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

```

```

int max_len(char *s[], int n);
void main()
{
    int i, n;
    char s[10][80], *p[10];

    scanf("%d", &n);
    /*-----*/
    for(i=0; i<n; i++)
    {
        scanf("%s", s[i]);
        p[i]=s[i];
    }
    printf("length=%d\n", max_len(p, n));
}
/*-----*/

int max_len(char *s[], int n)
{
    int max, i;
    max=strlen(s[0]);
    for(i=1; i<n; i++){
        if(strlen(s[i])>max)
            max=strlen(s[i]);
    }

    return max;
}

```

10025 程序填空，不要改变与输入输出有关的语句。

用字符指针实现函数 `str_cat(s, t)`，将字符串 `t` 复制到字符串 `s` 的末端，并且返回字符串 `s` 的首地址，并编写主程序。

例：（括号内为说明）

输入

abc

def

输出

abcdef

```
#include <stdio.h>
#include <string.h>
char *str_cat(char *s, char *t);
void main()
{
    char s[80], t[80];

    gets(s);
    gets(t);
    puts(str_cat(s, t));
}

/*-----*/
```

```
char *str_cat(char *s, char *t)
{
    return strcat(s, t);
}
```

10026 程序填空，不要改变与输入输出有关的语句。

编写一个程序，输入一个字符串后再输入 2 个字符，输出此字符串中从与第一个字符匹配的位置开始到与第二个字符匹配的位置之间的所有字符。用返回字符指针的函数实现。

例：（括号内为说明）

输入

program

r

g

输出

rog

```

#include <stdio.h>
char newstr[80];
void main()
{
    char str[80], c_begin, c_end;
    char *fun(char *p, char c_begin, char c_end);

    scanf("%s", str);
    getchar();
    c_begin=getchar();
    getchar();
    c_end=getchar();
    puts(fun(str, c_begin, c_end));
}
/*-----*/

```

```

char *fun(char *p, char c_begin, char c_end)
{

    char *s;

    while(*p){

        if(*p==c_begin)

            s=p;

            if(*p==c_end){

                *p=c_end;

                *(p+1)=0;

                break;

            }

            else

                *p++;

```

}

**return s;**

}

10027 程序填空，不要改变与输入输出有关的语句。

编写一个解密藏尾诗的程序。输入一首藏尾诗（假设只有 4 句），输出其藏尾的真实含义。用返回字符指针的函数实现。

例：

输入：

悠悠田园风

然而心难平

兰花轻涌浪

兰香愈幽静

输出：

风平浪静

```
#include <stdio.h>
#include <string.h>
char p[16];
void main()
{
    char    *change(char s[4][20]);
    char    s[4][20];
    int     i;
    for(i=0;i<4;i++)
        scanf("%s",s[i]);
    puts(change(s));
}
char    *change(char    s[4][20])
{
    int     i,t,len;
    /*-----*/
    for(i=0;i<4;i++){

        p[2*i]=s[i][strlen(s[i])-2];

        p[2*i+1]=s[i][strlen(s[i])-1];}
```



```

        p[2*i]='\0';
    return (p);
}

```

10028 程序填空，不要改变与输入输出有关的语句。

从键盘上输入一个  $3 \times 3$  的矩阵，求矩阵的主对角线和副对角线之和，要求定义和调用函数 `float matrix(float *a[3], int n)`，提示，函数的形参是指针数组。

例：

输入：

```

8   6   12
5   9   10
7   11   5

```

输出：

```

sum=41.00

```

```

#include <stdio.h>

```

```

/*-----*/

```

```

    float matrix(float*a[3],int n)
    {
        int i,j;

        float sum=0;

        for(i=0;i<3;i++)
            for(j=0;j<3;j++){
                if(i==j)
                    sum=sum+*(a+i+j);

                if(j==2-i&& i!=j)
                    sum=sum+*(a+i+2-i);
            }
    }

```

```

        return sum;

    }

void main()
{
    int i, j;
    float a[3][3], *p[3];

    for(i = 0; i < 3; i++)
        for(j = 0; j < 3; j++) {
            /*-----*/

            scanf("%f",&a[i][j]);

        }

        for(i=0;i<3;i++)

            p[i]=a[i];

    printf("sum=%0.2f \n", matrix(p,3));
}

```