

· 第七次上机题解

by CWD、郭晨旭

A 酸奶想成为魔法少女6

本题简单，只需要注意输入的温度是浮点数就行了

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

int main()
{
    double temp;
    char ch;
    scanf("%c",&ch);
    scanf("%lf",&temp);
    if(ch=='C')
        printf("%.2f",32+1.8*temp);
    else printf("%.2f",(temp-32)/1.8);
    return 0;
}
```

B CWD的脱单计划

本题考加权平均数计算，根据hint提示的公式和读入方式解决，要注意浮点数计算请一律用double，float会造成精度丢失的。

```
#include <stdio.h>

int main()
{
    int n;
    scanf("%d",&n);
    int sum=0,sumy=0;
    while(n--){
        int x,y;
        scanf("%d%d",&x,&y);
        sumy+=y;
        sum+=x*y;
    }
    printf("%.2f",(double)sum/sumy);
    return 0;
}
```

C 向量的内积

本题考察大家对循环和一维数组的应用情况。

```
#include <stdio.h>

int main()
{
    int n;
```

```

int a[15],b[15];
scanf("%d",&n);
for(int i=1;i<=n;i++)
    scanf("%d",a+i);
for(int i=1;i<=n;i++)
    scanf("%d",b+i);
int sum=0;
for(int i=1;i<=n;i++)
    sum+=a[i]*b[i];
printf("%d",sum);
return 0;
}

```

D Ange的爱心数字

本题考察基本的输入输出和转义符(本题借用出题人的代码)

```

#include <stdio.h>
#include <assert.h>

char one[] = "  _  \n /  \\  \n \\/) )  \n | |  \n | |  \n | |  \n _)"
(_ \n \\___/  \n";
char two[] = " _____ \n/  _  )\n\\\\ /  ) | \n   /  ) \n _/  / \n /  _/  \n(
(_/\n\n\\_____/ \n";
char three[] = " _____ \n/  _  \\ \n\\ /  \\ \n \\ \n  _  ) /\n ( _ ( \n   )
\\ \n/\n___/  /\n\\_____/ \n";
char zero[] = " _____ \n(  _  ) \n| (  ) | \n| | /  | \n| (/ /) | \n| / | | \n| ( _ )
| \n(_____) \n";

void prt(int x) {
    char *now;
    switch(x) {
        case 0: now = zero; break;
        case 1: now = one; break;
        case 2: now = two; break;
        case 3: now = three; break;
        default:
            assert(x >= 0 && x<=3);
            break;
    }
    printf("%s",now);
}

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

```

E lx学复变

本题考察多重循环，要注意的是i-j的下标必须控制在定义域范围内，不能越界，这个题要说明一下，在处理多组数据的时候最好每次都初始化一下使用过的变量和数组，本题向大家介绍memset函数，在<string.h>中，memset(数组名，你要置的数，sizeof(数组名))就可以给一个数组进行赋值操作，但是memset可以用来置0和-1，不能给它置1，这一点要切记！具体的可以百度memset的原理。

```

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

int f[1000],g[1000];
int y[1000];
int main()
{
    int t;
    scanf("%d",&t);
    while(t--){
        int n,m;
        scanf("%d%d",&n,&m);
        memset(y,0,sizeof(y));
        for(int i=0;i<n;i++)
            scanf("%d",f+i);
        for(int i=0;i<m;i++)
            scanf("%d",g+i);
        for(int i=0;i<=n+m-2;i++)
        {
            for(int j=0;j<m;j++){
                if(i-j>=0&&i-j<n)
                    y[i]+=f[i-j]*g[j];
            }
        }
        for(int i=0;i<=n+m-2;i++)
            printf("%d ",y[i]);
        printf("\n");
    }
    return 0;
}

```

bzb写密码学实验

- 关于二分法，直接用题目中所给的0和1作为开始时的左边界和右边界，利用单调递增函数这个条件，不断二分，直到误差小于 10^{-8}

```

#include <stdio.h>
#include <math.h>
double eps=1e-8;
double f(double x){//将右侧部分定义成一个函数
    return pow(x,2.5)+exp(x)+sin(x)+atan(x);
}
int main(){
    double y,r,l,mid,dif=1.0;
    l=0.0;
    r=1.0;
    scanf("%lf",&y);
    mid=(l+r)/2;
    dif=f(mid)-y;
    while(fabs(dif)>=eps){
        if(dif>0) r=mid;
        else l=mid;
        mid=(l+r)/2;
        dif=f(mid)-y;
    }
    printf("%.6lf",mid);
    return 0;
}

```

Count 327

- 一次性读入整个字符串，然后循环判断就好了

```
#include<stdio.h>
#include <string.h>
char str[100010];
int main(){
    gets(str);
    char a327[]="327";
    int i=0;
    int num=0;
    int len=strlen(str);
    for(i=0;i<=len-3;i++){//注意i的循环上界
        if(str[i]=='3'&&str[i+1]=='2'&&str[i+2]=='7') num++;
        //if(strncmp(str+i,a327,3)==0) num++;或者可以用strncmp函数，需要指针的知识，具体
        使用方法请百度
    }
    printf("%d",num);
    return 0;
}
```

寻根溯源

- 题目给出的数据范围并不大，而且只需要查找一次，我们可以把这个关系表存下来，然后首先查找当前这个人的父亲，将这个过程递归下去，直到查找不到当前这个人的父亲，我们就找到了辈分最大的人

```
#include <stdio.h>
#include <string.h>
char father[1510][30];
char son[1510][30];
int main(){
    int n,i;
    scanf("%d",&n);
    for(i=0;i<n;i++){
        scanf("%s%s",father[i],son[i]);
    }
    char name[30];
    scanf("%s",name);//用来记录待查找的名字
    int flag=1;//用来标志是否找到了当前的name的父亲
    int num=0;//记录查找次数
    while(flag){
        flag=0;
        for(i=0;i<n;i++){
            if(strcmp(name,son[i])==0){
                strcpy(name,father[i]);
                num++;
                flag=1;
            }
        }
    }
    printf("%s %d",name,num);
    return 0;
}
```

Ausar的弱点

- 首先存下总架势值，通过仔细读题，我们需要归类出三种情况来讨论
- 第一种情况攻击力为3，直接将原有总架势值减去90
- 第二种情况完美格挡，这时架势值减去基础值，如果架势值已经减为0，则不做任何改变（保证架势值大于等于0）
- 第三种情况就是架势值减去（攻击力%技巧值+基础值）
- 再通过判断架势值是否小于0来判断是否死亡
- 如果在最后架势值大于等于0，则输出剩余的架势值。

```
#include <stdio.h>
int main(){
    int n,level,e1e,sum;
    scanf("%d%d%d%d",&n,&level,&e1e,&sum);
    int i=0,attack;
    for(i=0;i<n;i++){
        scanf("%d",&attack);
        if(attack==3) sum-=90;
        else{
            if(attack%level==0){
                if(sum<e1e) sum=0;
                else sum-=e1e;
            }
            else{
                sum-=(attack%level+e1e);
            }
        }
        if(sum<0) {
            printf("DEATH\nKILL BY:%d",attack);
            return 0;//输出后可以退出程序
        }
    }
    printf("SHINOBI EXECUTION\nREMAIN:%d",sum);
    return 0;
}
```