

1001 可能不是碳基生物的签到题

不知道大家看到这题有没有想起前段时间的一些事情？

回到题目，“\”是个关键字，所以需要再加一个转义字符“\\”。

在验题时，我们还发现可以通过用对应ASCII码输出字符的方式绕开这个知识点（也就是注释的做法）。

```
#include<stdio.h>
int main(){
    printf("/\\");
    //printf("/%c",92);
}
```

1002 把握

有两种情况的步数是不可能的：

- 1.即使走最短路线步数也不够用
- 2.最短路线步数与潘子所报的步数之差为奇数，因为走奇数的步数不管怎么绕都不可能回到原位

原题链接 <https://ac.nowcoder.com/acm/contest/19304/H>

```
#include<stdio.h>
#include<math.h>
int main()
{
    int a,b,n;
    while(~scanf("%d %d %d",&a,&b,&n)){
        a=fabs(a);
        b=fabs(b);
        if(a+b<=n&&(a+b-n)%2==0)puts("chigua");
        else puts("haha");
    }
}
```

1003 w学长的签到题

正负数判断，大于0 正数，小于0 负数，注意 0 既不是整数也不是负数，按照题目要求格式输出。

代码：

```
#include <stdio.h>
int n,x;
int main(){
    while(scanf("%d",&n)!=EOF){
        int po_sum=0,ne_sum=0;
        while(n--){
            scanf("%d",&x);
```

```

        if(x<0) ne_sum++;
        if(x>0) po_sum++;
    }
    printf("Positive:%d\n",po_sum);
    printf("Negative:%d\n",ne_sum);
}
}

```

1004 w学长的基础数学

由小学的数学知识知道，两个 a/b ， c/d 分数相加，分子 等于 $a * d + b * c$ ，分母等于 $b * d$ 分子、分母同时除以分子和分母的最大公约数就行了。

如果能除尽输出整数就行。(最大公约数用辗转相除法求，不会的百度一下，算法竞赛一般也不会考原理，只会考怎么用)

代码：

```

#include <stdio.h>
int gcd(int x,int y){
    if(y==0) return x;
    else return gcd(y,x%y);
}
int main(){
    int a,b,c,d;
    while(scanf("%d%d%d%d",&a,&b,&c,&d)!=EOF){
        int den = b*d;
        int mol = a*d+b*c;
        printf("%d %d\n",mol/gcd(den,mol),den/gcd(den,mol));
    }
    return 0;
}

```

1005 w学长想挣钱

- 方法一

直接暴力枚举哪天取出来，哪天取出去，取一个最大值。

代码：

```

#include <stdio.h>
int main() {
    int n;
    double x, ans, sum;
    double nums[100005];
    while (scanf("%d", &n) != EOF) {
        for (int i = 1; i <= n; i++)
            scanf("%lf", &nums[i]);
    }
}

```

```

    ans = 1;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            sum = 1;
            for (int k = j; k <= i; k++) {
                sum *= nums[k];
            }
            ans = ans > sum ? ans : sum;
        }
    }
    printf("%.21f\n", ans);
}
return 0;
}

```

• 方法二

观察题目给的数据，我们发现，有些利率是小于1的，所以存在钱越存越少情况，如果都小于1了，那我们肯定不从前面几天开始存，我们会选择当天以后的开始存。这就是贪心的思路。

代码：

```

#include <stdio.h>
int main() {
    int n;
    double x, ans, sum;
    while (~scanf("%d", &n)) {
        ans = 1;
        sum = 1;
        while (n--) {
            scanf("%lf", &x);
            sum *= x;
            ans = ans > sum ? ans : sum; //三目运算符，表达式为真执行前面语句，为假执行后面语句
        }
        printf("%.21f\n", ans);
    }
    return 0;
}

```

1006 xt学长的正经签到题

根据题意，我们需要判断这个序列是不是非降序的，我们只需要判断，每一对相邻的两个数，前一个一定不能小于后一个，利用这个性质我们就可以判断了。

代码：

```

#include<stdio.h>
int main(){
    int n;
    while(~scanf("%d",&n)){
        int tmp = 0, x , flag = 0;

```

```

    for(int i = 1;i <= n;i++){
        scanf("%d",&x);//x表示当前的一个数，tmp表示前面一个数
        if(x < tmp) {
            flag = 1;
        }
        tmp = x;
    }
    if(!flag) puts("YES");
    else puts("NO");
}
return 0;
}

```

1007 xt学长喜欢6

这个题主要就考了你怎么把一个数的所有位都分解出来，其次判断是不是6的倍数。看代码。

代码：

```

#include<stdio.h>
int main(){
    int m;
    while(scanf("%d",&m)!=EOF){
        int i,count=0;
        for(i=1;i<=10000;i++){
            if(i%6==0 || (i/10)%10==6 || (i/100)%10==6 || i%10==6 || (i/1000)%10==6)
                count++;
            if(count==m){
                printf("%d\n",i);
                break;
            }
        }
    }
    return 0;
}

```

1008 xt学长移苹果

首先第一个不用移,所以我们肯定移离第一个比较近的，因为这样用的操作次数比较少，这样移完当前位置移下一个，直到当我们的操作数 $x=0$ 的时候就不能移了，这个时候就是最大数量了。这其实也是比较简单的一种小贪心，代码怎么实现呢？

代码：

```

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

int main()

```

```

{
    int n,x;
    while(scanf("%d %d",&n,&x)!=EOF)
    {
        int first,now,nowans;
        scanf("%d",&first);
        for(int i=2;i<=n;i++)
        {
            scanf("%d",&now);
            if(x>=now*(i-1))
                nowans=now;
            else
                nowans=x/(i-1);
            first=first+nowans;
            x-=nowans*(i-1);
        }
        printf("%d\n",first);
    }
    return 0;
}

```

1009 xt学长想让你找一个数

• 方法一

观察题目可以发现题目的数据范围比较小，所以其实我们是可以手动计算到50，然后把所有答案打印出来的。我这里用的数组，如果你没有学数组，那么你其实也可以用if，else打印，或者switch。

1~9不用说，答案肯定就是1-9；

其他的规律，要使数值最小，且长度肯定短，那我们肯定先填更大的。看下面规律：

最后一位肯定是9，他左边一位可以填1~8，目前我们就凑出了1-17；

再往下最后两位固定89，最左边一位可以填1~7，这样又凑出了1-34；

再往西最后三位固定789，最左边的一位可以填1~6，这样凑出了1-40；

再往西最后三位固定6789，最左边的一位可以填1~5，这样凑出了1-45；

所以大于45的都要输出 -1；

以此类推。

代码：

```

#include<stdio.h>

int a[]={
0,1,2,3,4,5,6,7,8,9,19,29,39,49,59,69,79,89,189,289,389,489,589,689,789,1789,278
9,3789,4789,5789,6789,16789,26789,36789,
46789,56789,156789,256789,356789,456789,1456789,2456789,3456789,13456789,2345678
9,123456789,-1,-1,-1,-1,-1
};

```

```

int main()
{
    int n;
    while(~scanf("%d",&n)){
        printf("%d\n",a[n]);
    }
    return 0;
}

```

• 方法二

我们手动计算的时候，其实算着算着就能发现这个题目的规律的。按照这个规律我们可以模拟一下。

```

#include<stdio>
#include<math.h>

int main()
{
    int n;
    while(~scanf("%d",&n)){
        int ans=0,wei=0;
        if(n>45){
            puts("-1");
            continue;
        }
        else {
            for(int i=9;i>0&& n>0;i--){
                if(n>=i){
                    ans+=(i*pow(10,wei));//pow(a,b),计算a的b次方，不会可以学一下
                    n-=i;
                    wei++;
                }
            }
            printf("%d\n",ans);
        }
    }
}

```