

LESSON_10 参考试题

一、 选择题

1. （样题）枚举算法的主要特点是（ ）。
A. 以空间换时间
B. 逐个尝试所有可能的解
C. 动态规划
D. 贪心策略

【答案】B

二、 判断题

1. （2023年9月）著名的哥德巴赫猜想：任一大于2的偶数都可写成两个素数之和。
我们可以通过枚举法来证明它。

【答案】错误

2. （样题）枚举算法是一种暴力求解方法，逐个尝试所有可能的解。

【答案】正确

三、 编程题

1. 楼层编号

【题目描述】

小林在 NOIP 比赛期间住在“新世界”酒店。和其他酒店不一样的是，这个酒店每天都有一个高能的数字 t ，这个数字在楼层中是不会出现的，以 $t=3$ 为例，则 3、13、31、33 等楼层是不存在的，楼层编号为 1,2,4,5, ... 所以实际上的 4 楼才是 3 楼。已知小林预定了编号为 m 层的房间，并且当天高能数字是 t ，现在他想知道房间所在真实楼层是多少。

【输入描述】

一行两个整数 m 和 t ， $1 \leq m \leq 100000$ ， $0 \leq t \leq 9$ ，保证 m 对 t 合法。

【输出描述】

一行一个整数，表示真实楼层。

【输入样例】

14 3

【输出样例】

12

【参考代码】

```

#include <iostream>
using namespace std;
int main() {
    int m,t;
    cin>>m>>t;
    int ans=0;
    for(int i=1;i<=m;i++){
        int x=i;
        bool flag=true;
        while(x>0){
            if(x%10==t) flag=false;
            x/=10;
        }
        if(flag) ans++;
    }
    cout<<ans;
    return 0;
}

```

2. 百钱买百鸡

【题目描述】

我国古代数学家张丘建在《算经》一书中提出的数学问题：鸡翁一值钱五，鸡母一值钱三，鸡雏三值钱一。百钱买百鸡，问鸡翁、鸡母、鸡雏各几何？

意思是：公鸡 1 只需要 5 钱，母鸡 1 只需要 3 钱，小鸡 3 只需要 1 钱，100 钱买 100 只鸡，公鸡、母鸡、小鸡各多少只？

【输入描述】

无

【输出描述】

一行中分别是公鸡、母鸡、小鸡的数量，用空格隔开，如果有多组答案，一组答案占一行，公鸡数量少的答案排在前面。

【参考代码】

```

#include <iostream>
using namespace std;
int main(){
    for(int x=0;x<=20;x++)
        for(int y=0;y<=33;y++)
            for(int z=0;z<=100;z++)
                if(x+y+z==100&&5*x+y*3+z/3.0==100)
                    cout<<x<<" "<<y<<" "<<z<<endl;
    return 0;
}

```

3. 小猫分鱼 (2023年12月)

【题目描述】

海滩上有一堆鱼，N 只小猫来分。第一只小猫把这堆鱼平均分为 N 份，多了 $i < N$ 个，

这只小猫把多的 i 个扔入海中，拿走了一份。第二只小猫接着把剩下的鱼平均分成 N 份，又多了 i 个，小猫同样把多的 i 个扔入海中，拿走了一份。第三、第四、.....，第 N 只小猫仍是最终剩下的鱼分成 N 份，扔掉多了的 i 个，并拿走一份。

编写程序，输入小猫的数量 N 以及每次扔到海里的鱼的数量 i ，输出海滩上最少的鱼数，使得每只小猫都可吃到鱼。例如：两只小猫来分鱼 $N=2$ ，每次扔掉鱼的数量为 $i=1$ ，为了每只小猫都可吃到鱼，可令第二只小猫需要拿走 1 条鱼，则此时待分配的有 3 条鱼。第一只小猫待分配的鱼 $3*2+1=7$ 条。

【输入描述】

总共 2 行。第一行一个整数 N ，第二行一个整数 i 。保证 $0 < N < 10 ; i < N$ 。

【输出描述】

一行一个整数，表示满足要求的海滩上最少的鱼数。

【输入样例 1】

2

1

【输出样例 1】

7

【输入样例 2】

3

1

【输出样例 2】

25

【参考代码】

```

#include <iostream>
using namespace std;
int main(){
    int n,i;
    cin>>n>>i;
    int ans=0;
    int k=1;
    while(true){
        bool flag=true;
        ans=k*n+i;
        for(int j=1;j<n;j++){
            if(ans%(n-1)!=0){
                flag=false;
                break;
            }
            ans=ans/(n-1)*n+i;
        }
        if(flag) break;
        k++;
    }
    cout<<ans;
    return 0;
}

```