Bucket Filling

Nejiya has a bucket having a capacity of K liters. It is already filled with X liters of water.

Find the maximum amount of extra water in liters that Nejiya can fill in the bucket without overflowing.

Input Format

The first line will contain T - the number of test cases. Then the test cases follow.

The first and only line of each test case contains two space separated integers K and X - as mentioned in the problem.

Output Format

For each test case, output in a single line, the amount of extra water in liters that Nejiya can fill in the bucket without overflowing.

Sample Input 1

```
2
5 4
15 6
```

Sample Output 1

```
1
9
```

C Program

```c
#include <stdio.h>
int main(void) {
        int t;
        scanf("%d",&t);
        while(t--)
        {
           int k,x;
        scanf("%d %d",&k,&x);
        printf("%d\n",k-x);
```

```
        }
        return 0;
}
```

C++ Program

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

int main() {
        int a;
        cin>>a;
        while(a--){
            int b,c;
        cin>>b>>c;
        cout<<b-c<<"\n";
        }
        return 0;
}
```
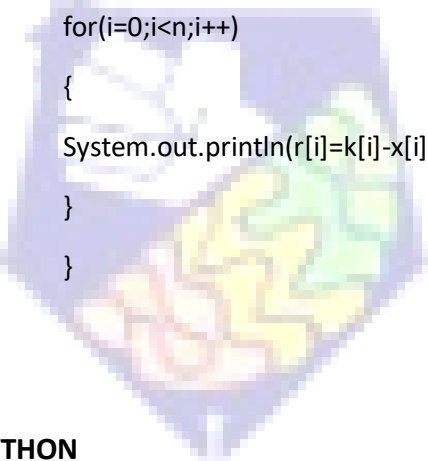
JAVA

```java
import java.util.*;
import java.lang.*;
import java.io.*;
class Main
{
        public static void main (String[] args) throws java.lang.Exception
        {
        int n;
        Scanner sc=new Scanner(System.in);
```

```
        n=sc.nextInt();

        int []k=new int [n];

        int []x=new int [n];

        int []r=new int [n];

        int i;

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

        {

            k[i]=sc.nextInt();

            x[i]=sc.nextInt();

            r[i]=k[i]-x[i];

        }

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

        {

        System.out.println(r[i]=k[i]-x[i]);

        }

        }

}
```

PYTHON

```
r = int(input())

for i in range(r):

k,y = map(int,input().split())

ans = int(k-y)

    a = ans + y

    print(ans)
```