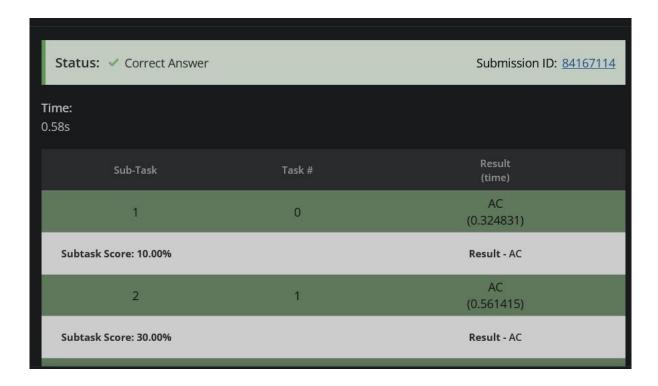
Worksheet 1

Q1. Fire and Ice Code:

```
#include <stdio.h>
#include<inttypes.h>
void multiply(uint64_t F[2][2], uint64_t M[2][2], uint64_t k);
void power(uint64_t F[2][2], uint64_t n,uint64_t k); uint64_t
fib(uint64_t n,uint64_t k)
{
uint64_t F[2][2] = {{1,1},{1,0}};
if (n == 0) return 0; power(F,
n-1,k); return F[0][0];
}
void power(uint64_t F[2][2], uint64_t n,uint64_t k)
{
if( n == 0 | | n == 1)
return;
uint64_t M[2][2] = {{1,1},{1,0}};
power(F, n/2,k); multiply(F,
F,k);
if (n%2 != 0) multiply(F,
M, k);
}
```

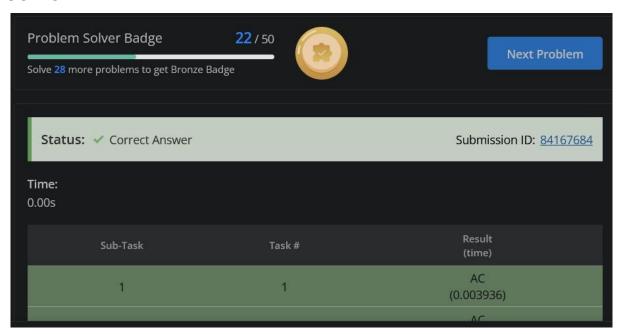
```
void multiply(uint64_t F[2][2], uint64_t M[2][2], uint64_t k)
{
uint64_t x = (F[0][0]*M[0][0] + F[0][1]*M[1][0])%k; uint64_t
y = (F[0][0]*M[0][1] + F[0][1]*M[1][1])%k; uint64_t z =
(F[1][0]*M[0][0] + F[1][1]*M[1][0])%k; uint64_t w =
(F[1][0]*M[0][1] + F[1][1]*M[1][1])%k;
F[0][0] = x; F[0][1]
= y;
F[1][0] = z;
F[1][1] = w;
}
int main()
{
uint64_t n,k,t;
scanf("%llu",&t);
while(t--)
{
scanf("%llu",&n);
scanf("%llu",&k); printf("%llu\n",
(2*fib(n,k))%k);
}
return 0;
}
```



Q2. Gold Mining

Code:

```
if(sum<x)
{
printf("NO\n");
}
else
{
printf("YES\n");
}
return 0;
}</pre>
```

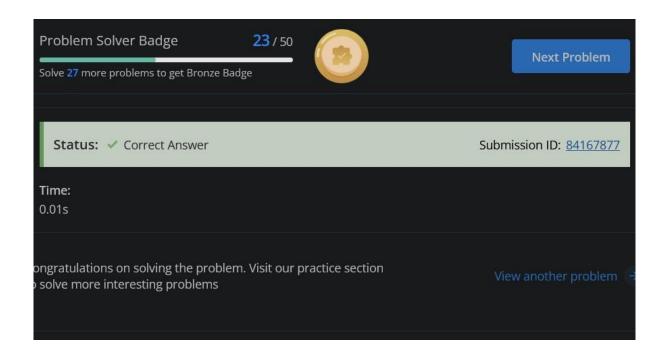


Q3 The Lead Game

Code:

#include<bits/stdc++.h>

```
using namespace std;
typedef long long int lli;
int main(){ int
t,S=0,T=0; cin>>t;
vector<int> v; while(t--
){ int s,t;
cin>>s>>t; S+=s;
T+=t;
v.push_back(S-T);
}
int max=-1,win;
for(int i:v){
if(abs(i)>max){
max=abs(i); if(i>0)
win = 1; else win
= 2;
}}
cout<<win<<' '<<max;
}
```

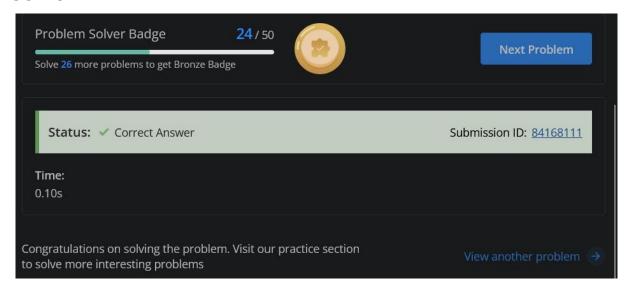


Q4. Sums in a triangle

Code:

```
#include<bits/stdc++.h>
using namespace std;
int main(){ int i,j,t,n;
cin>>t; while(t--){
  cin>>n; int a[n][n];
  for(int i=0;i<n;i++){
  for(j=0;j<=i;j++){
    cin>>a[i][j];
  }}
  fo
  r(
  in
  t
```

```
i=
n-
2;
i>
=
0;
i---
){
fo
r(
j=
0;
j<
=i
;j
+
+)
{
if((a[i][j]+a[i+1][j])>(a[i][j]+a[i+1][j+1]))
a[i][j]=a[i][j]+a[i+1][j]; else
a[i][j]=a[i][j]+a[i+1][j+1];
}
}
cout<<a[0][0]<<endl;
}
return 0;
```



Q5. Small Factorials

```
#include
<bits/stdc++.h>
#include <boost/multiprecision/cpp_int.hpp>
#include <iostream>
using namespace std;
using namespace boost::multiprecision; int
main() {
   // your code goes here
int t; cin>>t;
while(t--)
{ int n; cin>>n;
cpp_int fact=1;
for(int i=n;i>0;i--)
```

```
fact=fact*i;
cout<<fact<<endl;
}
return 0;
}</pre>
```

