1

10055 - Hashmat the Brave Warrior

main.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;
int main() {
    long long a, b;
    while(cin>>a>>b) {
        cout<<abs(a-b)<<endl;
    }
    return 0;
}
```

Explanation: Simple subtraction to determine the difference between Hashmat's army and the opponent's army.

2

10071 - Back to High School Physics

*main.cpp

```cpp
#include <iostream>
using namespace std;
int main() {
    int s, u=0, v, t, a=0;
    while(cin>>v>>t) {
        s= v*t-0.5*a*(t*t);
        cout<<2*s<<endl;
    }
    return 0;
}
```

Explanation: Using formula: S= vt-0.5*a*t*t
Here a=0;

3

10281 - Average Speed

*main.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;
int main() {
    int h, m, s, speed=0;
    double S=0, S_save, res=0;
    char c;
    while(cin>>h>>c>>m>>c>>s) {
        S = (((h*60)*60)+(m*60)+s);
        if (cin.peek() == ' ') {
            if(!speed==0) {
                res+= speed*(((S-S_save)/3600));
            }
            cin>>speed;
        } else {
            res+= speed*(((S-S_save)/3600));
            printf("%02d:%02d:%02d %.2f km\n", h, m, s, res);
        }
        S_save=S;
    }
    return 0;
}
```

Explanation: We find spending time between the given time and calculate the distance with the help of the previously given speed and save the distance value for the next calculation.

4

10469 - To Carry or not to Carry

*main.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;
int main() {
    int a, b, c=0;
    while(cin>>a>>b) {
        bitset<32>x(a);
        bitset<32>y(b);
        bitset<32>z(c);
        for(int i=0; i<32; i++) {
            if(x[i]==y[i]) {
                z[i]=0;
            } else {
                z[i]=1;
            }
        }
        cout<<z.to_ullong()<<endl;
    }
    return 0;
}
```

Explanation:

- Three bitsets (x, y, and z) are created, each with a size of 32 bits, initialized with the values of a, b, and c
- Using for loop for Bitwise XOR Operation (If the bits in x and y are the same, the bit in z is set to 0, otherwise, it's set to 1) as Mofiz did.
- The result, stored in bitset z, is converted to an unsigned long long integer using to_ullong() and then printed to the result

5

10773 - Back to Intermediate Math

*main.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;
int main() {
    double T, t1, t2, P, d, v, u;
    cin>>T;
    for(int i=1; i<=T; i++) {
        cin>>d>>v>>u;
        cout<<"Case "<<i<<": ";
        if(v>=u || d*v*u==0) {
            cout<<"can't determine"<<endl;
        } else {
            t1=d/u;
            t2=d/sqrt(u*u-v*v);
            P=abs(t1-t2);
            printf("%.3f\n", P);
        }
    }
    return 0;
}
```

Explanation: We find time for Fastest Path and Shortest Path using formulas t2 and t1 for crossing the river.

6

11614 - Etruscan Warriors Never Play Chess

*main.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;
int main() {
    long long T, S, a, b, c, res;
    cin>>T;
    while(T--){
        cin>>S;
        a=1;
        b=1;
        c=-2*S;
        res=(-1+sqrt(b*b-4*a*c))/(2*a);
        cout<<res<<endl;
    }
    return 0;
}
```

Explanation: $1+2+3+....+x = \frac{x*(x+1)}{2} = S$

$\Rightarrow x*x + x - 2S=0$

Here: a=1, b=1, c=-2*S

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

7

11723 - Numbering Roads

*main.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;
int main() {
    int R, N, i=1, nS, tS, res;
    while(cin>>R>>N && R*N!=0) {
        cout<<"Case "<<i++<<": ";
        nS=R-N;
        tS=N*26;
        if(nS<=0) {
            cout<<"0"<<endl;
        } else if(nS > tS) {
            cout<<"impossible"<<endl;
        } else if(nS<N) {
            cout<<"1"<<endl;
        } else if(nS%N > 0) {
            cout<<(nS/N)+1<<endl;
        } else {
            cout<<nS/N<<endl;
        }
    }
    return 0;
}
```

Explanation: We determines the minimum number of different suffixes required to name streets, considering a limited number of integers allocated for naming, and outputs the results for multiple cases, indicating "impossible" when necessary. Simple if else statement.

8

11805 - Bafana Bafana

*main.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;
int main() {
    int T, N, K, P, c=1;
    cin>>T;
    while(T--) {
        cin>>N>>K>>P;
        cout<<"Case "<<c++<<": ";
        if((K+P)%N==0) {
            cout<<N<<endl;
        } else {
            cout<<(K+P)%N<<endl;
        }
    }
    return 0;
}
```

Explanation: Using formula (K+P)%N to find who will give the ball back to Parreira.

9

11875 - Brick Game

*main.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;
int main() {
    int T, N, mid, p, res, c=1;
    cin>>T;
    while(T--) {
        cin>>N;
        cout<<"Case "<<c++<<": ";
        mid=(N/2)+1;
        for(int i=1; i<=N; i++) {
            cin>>p;
            if(i==mid) res=p;
        }
        cout<<res<<endl;
    }
    return 0;
}
```

Explanation: Given ages are in strictly increasing order or strictly decreasing order. So we just find the mid-sequence.

10

12149 - Feynman

*main.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;
int main() {
    int N, res, c=1;
    while(cin>>N && N!=0) {
        res=0;
        for(int i=1; i<=N; i++) {
            res+=i*i;
        }
        cout<<res<<endl;
    }
    return 0;
}
```

Explanation: Using simple loop to find the result.

$1^2+2^2+3^2+\cdots+n^2 = \text{Result}$

11

12502 - Three Families

*main.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;
int main() {
    float T, x, y, z, Fhr, Ihr, res;
    cin>>T;
    while(T--) {
        cin>>x>>y>>z;
        Fhr=(x+y)/3;
        Ihr=z/Fhr;
        res=Ihr*(x-Fhr);
        cout<<res<<endl;
    }
    return 0;
}
```

Explanation:

Per Family (Hour) = (x+y)/3;

Income (Doller/Hour) = z/Fhr;

Family_A Income (Total) = Ihr*(x-Fhr);