```cpp
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
#include <cmath>

class Base
{
public:
    int x, y, health;

    Base(int x, int y, int health) : x(x), y(y), health(health) {}
};

class Hero
{
public:
    int id, x, y;

    Hero(int id, int x, int y) : id(id), x(x), y(y) {}
};

class Monster
{
public:
    int id, x, y, health, vx, vy, near_base, threat_for;

    Monster(int id, int x, int y, int health, int vx, int vy, int
near_base, int threat_for)
        : id(id), x(x), y(y), health(health), vx(vx), vy(vy),
near_base(near_base), threat_for(threat_for) {}
};

double distance(int x1, int y1, int x2, int y2)
{
    return sqrt(pow(x1 - x2, 2) + pow(y1 - y2, 2));
}

Monster *find_nearest_monster(const std::vector<Monster> &monsters,
const Hero &hero)
{
    Monster *nearest_monster = nullptr;
    double min_distance = std::numeric_limits<double>::max();
    for (const auto &monster : monsters)
    {
        if (monster.threat_for == 1)
        {
            const double dist = distance(hero.x, hero.y, monster.x,
monster.y);
            if (dist < min_distance)
            {
                min_distance = dist;
                nearest_monster = const_cast<Monster *>(&monster);
            }
        }
    }

    return nearest_monster;
}

int main()
{
    int base_x, base_y;
    std::cin >> base_x >> base_y;
    std::cin.ignore();
    int heroes_per_player;
    std::cin >> heroes_per_player;
    std::cin.ignore();

    while (true)
    {
        std::vector<Base> bases;
        for (int i = 0; i < 2; ++i)
        {
            int health, mana;
            std::cin >> health >> mana;
            std::cin.ignore();
            bases.emplace_back(i == 0 ? base_x : 17630 - base_x, i == 0
? base_y : 9000 - base_y, health);
        }

        int entity_count;
        std::cin >> entity_count;
        std::cin.ignore();
```

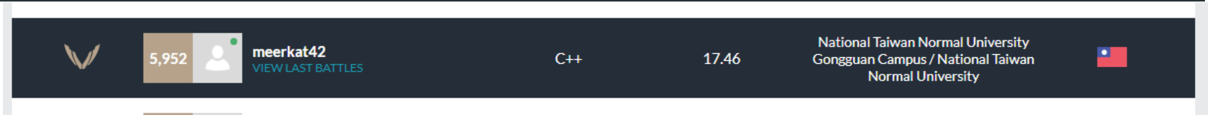
```cpp
        std::vector<Hero> heroes;
        std::vector<Monster> monsters;
        for (int i = 0; i < entity_count; ++i)
        {
            int id, type, x, y, shield_life, is_controlled, health, vx,
vy, near_base, threat_for;
            std::cin >> id >> type >> x >> y >> shield_life >>
is_controlled >> health >> vx >> vy >> near_base >> threat_for;
            std::cin.ignore();

            if (type == 1)
            {
                heroes.emplace_back(id, x, y);
            }
            else if (type == 0)
            {
                monsters.emplace_back(id, x, y, health, vx, vy,
near_base, threat_for);
            }
        }

        for (auto &hero : heroes)
        {
            Monster *nearest_monster = find_nearest_monster(monsters,
hero);
            if (nearest_monster)
            {
                std::cout << "MOVE " << nearest_monster->x << " " <<
nearest_monster->y << std::endl;
            }

            else
            {
                std::cout << "WAIT" << std::endl;
            }
        }
    }

    return 0;
}
```


5,952 meerkat42
VIEW LAST BATTLES
C++
17.46
National Taiwan Normal University Gongguan Campus / National Taiwan Normal University