**Input Data**

1. **First Line:** Describe the information of the knight.

+ HP: (1 - 999) – Is Also Maxium Health

+ level: (1 - 10)

+ remedy: (0 - 99) – Potions

+ maidenkiss: (0 - 99) – Potions

+ phoenixdown: (0-99) – Potions

+ rescue: **UPDATE** after handling events

\* -1: journey not over, continue

\* 1: journey over, princess is saved, end

\* 0: journey over, princess isn’t saved, end

**Ex:** 172 2 0 1 1

1. **Second Line:** List of events, is indexed starting at 1, each event = event code, not fixed, can duplicated

**Ex:** 1 2 9 2 18 99

|  |  |
| --- | --- |
| Event Code | Meaning |
| 0 | Bowser lose, win princess |
| 1 | Meet MadBear |
| 2 | Meet Bandit |
| 3 | Meet LordLupin |
| 4 | Meet Elf |
| 5 | Meet Troll |
| 6 | Meet Shaman |
| 7 | Meet Siren Vajsh |
| 11 | Pick Up MushMario |
| 12 | Pick Up MushFib |
| 13 | Pick Up MushGhost |
| 15 | Obtain Remedy |
| 16 | Obtain Maiden Kiss |
| 17 | Obtain Phoenix Down |
| 18 | Meet Merlin |
| 19 | Meet Asclepius |
| 99 | Meet Bowser |

1. **Third Line:** Files used in some events, format:

<file\_mush\_ghost>, <file\_asclepius\_pack>, <file\_merlin\_pack>

**Description**

1. Event Code 0:

* End the game, return current stats

rescue = 1;

// end the game

1. Event Code 1 to 5:

* Engage in combat, oponent has levelO
* Calculate levelO (with index i):

int b = i % 10;

int levelO = (i > 6) ? ((b > 5) ? b : 5) : b;

* (min = 5 when i >= 6): 1 2 3 4 5 6 7 8 9 5 5 5 5 5 5 6 7 8
* If knight’s level > levelO: win, increase level by 1, continue to the next event

level += 1;

//continue to the next event

* If knight’s level = levelO: drawn, continue to the next event

//continue to the next event

* If knight’s level < levelO: lose, take damage, check if hp <= 0 to know whether can we continue (with the help of phoenixdown)

float arr\_baseDamage[5] = {-1, 1, 1.5, 4.5, 7.5, 9.5};

HP -= arr\_baseDamage[curEvent] \* levelO \* 10;

if (HP <= 0)

{

    if (phoenixdown <= 0)

    {

        rescue = 0;

        // end the game

    }

    else

    {

        phoenixdown -= 1;

        HP = maxHP;

        // continue to the next level

    }

}

* **Ex1:**
* 172 1 0 1 0 – 5 2
* In event 1:

+ Event Code = 5, Meet Troll

+ Knight’s Level = 1 == levelO, draw, continue

* In event 2:

+ Event Code = 2, Meet Bandit

+ Knight’s Level = 1 < levelO = 2, lose

+ HP = 172 – (1.5 \* 10 \* 2) = 142 > 0, continue

* Reach the end of events, return 142 1 0 1 0 1
* **Ex2:**
* 152 1 0 1 1 – 3 5 4
* In event 1:

+ Event Code = 3, Meet Lord Lupin

+ Knight’s Level = 1 == levelO, draw, continue

* In event 2:

+ Event Code = 5, Meet Troll

+ Knight’s Level = 1 < levelO = 2, lose

+ HP = 152 – (9.5 \* 10 \* 2) = -38 <= 0

+ phoenixdown = 1 – revive

+ phoenixdown = 1- 1 = 0; HP = maxHP = 152; continue

* In event 3:

+ Event Code = 4, Meet Elf

+ Knight’s Level = 1 < levelO = 3, lose

+ HP = 152 – (7.5 \* 10 \* 3) = -73

+ phoenixdown = 0 – fucking die

+ rescue = 0; end the game

1. Event Code 6

* Similar to event from 1 to 5: engage in combat, calculate levelO, …
* If knight’s level > levelO: win, increase level by 2, continue to the next event
* If knight’s level == levelO: draw, continue to the next event
* If knight’s level < levelO:

+ Tiny effect activate, HP reduce to 1/5 (floor) in the next 3 events – if HP is <= 5 then turn to 1;

+ After 3 events, HP multiply by 5, cap at maxHP.

* If the knight has remedy potion, use it to turn back to previous HP, disable tiny effect
* If use phoenixdown when tiny, return hp to max, disable tiny effect

// When lose to the shaman

preHP = HP;

if (HP <= 5)

{

    HP = 1;

}

else

{

    HP /= 5; // cpp automatically floor the result

}

tinyEffect = 4;

// For each event

if (tinyEffect >= 1)

{

    // If has remedy

    if (remedy >= 1)

    {

        remedy -= 1;

        HP = preHP;

        tinyEffect = 0;

    }

    else

    {

        tinyEffect -= 1;

        if (tinyEffect == 0)

        {

            HP \*= 5;

            // Cap at maxHP

            if (HP > maxHP)

            {

                HP = maxHP;

            }

        }

    }

}

// If use phoenixdown

// #phoenixdown code here

if (tinyEffect >= 1)

{

    tinyEffect = 0;

}

1. Event Code 7

* Similar to event from 1 to 5: engage in combat, calculate levelO, …
* If knight’s level > levelO: win, increase level by 2, continue to the next event
* If knight’s level == levelO: draw, continue to the next event
* If knight’s level < levelO: turn into a frog in the next 3 events, level reduce to 1 in the next 3 events – even if level increase in frog state, level still return to prevLevel after the 3 events
* If the knight has maidenkiss potion, return to the prevHP
* Shaman and Vajsh will skip the fight if istiny or isfrog

// Before fighting Shaman or Vajsh

if ((frogEffect >= 1) || (tinyEffect >= 1))

{

    // continue to the next events == draw

}

// When lose to Vajsh

preLevel = level;

level = 1;

frogEffect = 4;

// For each event

if (frogEffect >= 1)

{

    // If has maidenkiss

    if (maidenkiss >= 1)

    {

        maidenkiss -= 1;

        level = preLevel;

        frogEffect = 0;

    }

    else

    {

        frogEffect -= 1;

        if (frogEffect == 0)

        {

            level = preLevel;

        }

    }

}

1. Event Code 11

* Increase HP, cap at maxHP

n1 = ((level + phoenixdown) % 5 + 1) \* 3;

for (int i = 99; i > (99 - 2 \* n1); i -= 2)

{

    s1 += i;

}

HP = HP + (s1 % 100);

if (HP > maxHP)

{

    HP = maxHP;

}

// continue to the next level

1. Event Code 12

* HP drops to the nearest Fibonanci Number

int CalculateFibo(int *n*)

{

    if ((*n* == 1) || (*n* == 2))

    {

        return 1;

    }

    return CalculateFibo(*n* - 1) + CalculateFibo(*n* - 2);

}

// Create an array of Fibonanci Number - HP cap at 999

// So arr\_fibo.lenght = 16 ~ CalculateFibo(16) = 987

// Save this array as public to avoid re-initialize it

int arr\_fibos[16];

for (int i = 0; i < 16; i++)

{

    // Can use for-loop if you want better time complexity

    arr\_fibos[i] = CalculateFibo(i + 1);

}

// Decreasse HP to the nearest Fibonanci Number

for (int i = 0; i < 16; i++)

{

    if (arr\_fibos[i] > HP)

    {

        // Work just fine if HP = Fibonanci Number

        HP = arr\_fibos[i - 1];

        break;

        // continue to the next level

    }

}