Tsuen Wan Government Secondary School

Olympiad in Informatics Team

2025/26 INTER-HOUSE CODING COMPETITION

SCHEDULE

Date 10 February 2026

Duration 4 hours

PROBLEMS

	Problem	Limit		Score	
		Time	*Memory*	*Subtasks*	*Total*
TH261	The Chosen One	1s	256MB	10 | 13 | 22 | 15	60
TH262	Extra Damage	1s	256MB	9 | 16 | 27 | 18	70
TH263	Legendary Card	1s	256MB	11 | 21 | 33 | 25	90
TH264	Defence is Best Attack	1s	256MB	9 | 17 | 24 | 33 | 27	110
TH265	Sacrifice for the Win	1s	256MB	10 | 24 | 18 | 21 | 31 | 26	130
TH266	Luck is Skill	1s	256MB	150	150
TH267	Boss Fight	1s	256MB	17 | 35 | 23 | 32 | 28 | 45	180
TH268	Dream or Reality?	1s	256MB	210	210
				Full Score	1000

REMINDERS

- All problems are divided into subtasks. All test cases in a subtask must be passed to get points.
- You should answer in C++20. You may also attempt in other secondary languages provided on TWGSS Online Judge, including Python 3. It is not guaranteed that the problems are solvable using those secondary languages.
- Unless otherwise specified, your output will be automatically fixed as follows:
  - Trailing spaces in each line will be removed.
  - An end-of-line character will be added to the end of the output if not present.
- 64-bit integers may be required in some problems, use `long long` in C++ if needed.
- All stories are hypothetically written. In case of coincidences, they are coincidences.

TH261 The Chosen One

Score	Limit	
60	1s	256MB

Fuzzy Dreams.

When you awake, you suddenly saw a guy sitting in front of you, separated by a desk. There are many cards placed on the desk, displayed on the wall, …, basically everywhere. The cards actually look pretty good.

"Why am I here?" you asked, oblivious of the reason why you are seeing what you are seeing.

"Well, because… you are the chosen one to save this kingdom from despair! Once upon a time, we are a strong Card Kingdom," said the guy. "However, three years ago, …"

"Okay, I do not need the background story. Can you just tell me what to do?" You asked.

The guy smiled and replied, "don't be so impatient, young person. Patience is a good personality to have. Back in the days, …"

"Okay I really do not need the background story. Can you really just tell me what to do?" You asked again.

The guy laughed, "haha, you got me. Long story short, since you are the chosen one, you should be able to solve this very simple task. In the Card Kingdom, there are three pure elements, Fire, Wood and Water. The interesting part is that, you can further mix different elements together. For example, if you merge Fire and Wood, you get Firewood; if you merge Fire and Water, you get Firewater; and if you merge Wood and Water, you get Woodwater. Finally, if you merge all three together, you get Rainbow."

"Alright, I kind of get what you mean. What is so different between pure and mixed elements?" You are (forced to be) interested about what the guy is saying.

The smile on the guy glow even larger, and said: "Well, each pure element card (Fire, Wood, Water) worths 1 gold coin, each mixed element card of two (Firewood, Firewater, Woodwater) worths 3 gold coins, while the Rainbow element card worths 7 gold coins! Your target is to maximise the gold coin that the elements to be sold. As the chosen one, this should be trivial for you, right?"

INPUT

The first and only line contains three integers R, G and B, the number of Fire, Wood and Water elements, respectively.

OUTPUT

Output a single integer, the maximum worth of the elements (in gold coins) that can be obtained by merging given elements together.

SAMPLE TESTS

	Input	Output
1	3 1 4	14

You can form 1 Rainbow, 2 Firewater and 1 Water elements, resulting the final worth of $1 \times 7 + 2 \times 3 + 1 = 14$ gold coins.

	Input	Output
2	2 3 0	7

CONSTRAINTS

$0 \leq R, G, B \leq 10^8$

SUBTASKS

	Points	Constraints
1	10	$G = B = 0$
2	13	$B = 0$
3	22	$R \geq G \geq B$
4	15	No additional constraints

TH262 Extra Damage

Score	Limit	
70	1s	256MB

"Good job", said the guy, "apparently you are really the chosen one! Hmmm… Let me see… Let me introduce you some of the mechanics before you go out and fight some stronger opponents!" The guy turn around and take out a deck of cards with numbers written on it. Why does he have so many decks of cards?

"This is the training deck", said by the guy with a straight face. "You can treat the numbers as the attack value. Maybe I say it this way - imagine you are fighting a monster card with a certain HP. What if your attack is too strong, and dealt way too much damage to the monster card?"

"Huh the monster dies?" you answered, confused of why the guy will ask you this obvious question.

"Yes you are right! However, there is a catch - if you over-hit the card - that means if you hit more than your monster's HP, then the drop item from the monster would improve with the extra damage! For example, if the monster HP is 3 and you dealt 2 damage then 3 damage in two rounds, then 2 extra damage is dealt. Obviously, you cannot continue to hit the monster after it dies - so it takes some skill to do this exactly", explained the guy.

"Okay I kind of get what you mean, interesting!" you said.

The guy smiled and said, "good! Now I have cards of attack value from 1 to N - and imagine there is a monster card of HP K. Just like in real life, you cannot reused the same card twice. How would you attack the monster to get the best drop item? Take your time, since this is a very helpful skill to have later on! Back in the days, …"

"Alright, please give me some time to investigate how this mechanics work!" you said, ignoring what the guy is going to say.

INPUT

The first and only line contains two integers N and K, the number of cards you owned and the enemy HP, respectively.

OUTPUT

On the first line, output a single integer, the maximum extra damage dealt to the enemy.

On the second line, output the length of your playing sequence.

On the third line, output the cards that you are playing in sequential order.

SAMPLE TESTS

	Input	Output
1	3 3	2 2 2 3

The enemy has 3 HP initially.

After the first attack of power 2, the enemy HP becomes 1.

Then, with the second attack of power 3, the enemy is beaten, and 2 extra damage is dealt.

2	4 8	2 4 1 2 4 3
3	2 3	0 2 2 1

CONSTRAINTS

$2 \leq N \leq 1000$

$1 \leq K \leq \frac{N(N+1)}{2}$

SUBTASKS

	Points	Constraints
1	9	$N = 2$
2	16	$K \leq N$
3	27	$K \leq \frac{N(N-1)}{2}$
4	18	No additional constraints

TH263 Legendary Card

Score	Limit	
90	1s	256MB

The guy smiled at you. “Very good! You now have some basic concepts about how this works.” said the guy. “Let me gift you one of the strong card…”, he turn around, open a drawer in the many drawers he has, and pick out a card. “This is one of the legendary cards I have - let me gift it to you, the chosen one!”

“Seems there are stars on the card - wow there are 15 stars on the card! Is there a limit of how many stars are there?” you asked the guy.

The guy smiled and said, “you ask the right question - the number of stars is the rarity of the card. The higher the rarity of a card, the higher the attack and defence value of the card becomes. Furthermore, the card might gain some special effects, for example double attack with some probability! You can go to the stone card merger - you see, the machine next to the door - to merge two cards of the same rarity. After pressed the two cards together, you will get any card of one higher rarity!”

“Is there a limit of how many stars can be on the same card?” You ask, confused.

The guy smirked and said, “not really, you can merge cards to obtain however legendary cards.” The guy continued, “however, each one more star consumed exponential amount of resources - for example, if you want a legendary card of 30 stars, you will need 2^{30} normal cards of 1 star. That would cost you an eternity to collect!”

“Alright - let me buy some extra cards with my coins to see how legendary my cards can get,” you thought to yourself. You bring out the gold coins, then the guy just pulled out a deck of cards of different rarities. “You can buy a consecutive stack of cards within this deck”, said the guy, “then you can merge however you wish! These common cards are of rarity 1 to 10 stars, since cards of even higher rarity are considered rare. My price is 1 gold coin per card. How would you like to choose?”

You quickly pick up the deck and skim through the rarity on the cards. You quickly thought through many different scenarios of which stack to pick, “what should I choose, to obtain the most legendary card while not cost that much”, you thought.

“Can you give me some time to do some calculations?” You asked the guy. “Sure, you can and should take your time to think clearly. I would wait you before dinner!”

INPUT

The first line of input contains an integer N, the number of cards.

The second line contains N integers $A_1, A_2, \ldots, A_N$, representing the rarity of the cards in the deck in this order.

OUTPUT

Output the maximum number of stars and the corresponding minimum cost in gold coins by buying a consecutive stack of cards, after arbitrary merging, on the same line.

SAMPLE TESTS

	Input	Output
1	7 1 3 2 1 1 2 4	5 6

Buying the 2^{nd} to the 7^{th} card costs 6 gold coins. We can obtain a 5 stars card by merging them.

	Input	Output
2	4 1 1 1 1	3 4

CONSTRAINTS

$1 \leq N \leq 2 \times 10^5$

$1 \leq A_i \leq 10$

SUBTASKS

	Points	Constraints
1	11	$N = 2$
2	21	$A_i = 1$ for all $1 \leq i \leq N$
3	33	$1 \leq N \leq 5000$
4	25	No additional constraints

TH264 Defence is Best Attack

Score	Limit	
110	1s	256MB

Selling you your first deck of card, the guy smiled and said, “very good. I have nothing else to teach you - you would need to explore on your own! Experience is the best teacher. You know, back in the days, …”

“Alright thank you very much for your help! I would be forever thankful for you!” you thank him again before he talked too much, as it would be time-consuming to see too many useless stories.

“Wait wait wait before you leave, at least take one of the manuals that I write before,” the guy opens some random drawer again and take out a book that is slightly dusty, “you can take this, and this will be very helpful when you face any issues!” said the guy.

“Thank you very much!” Take the book, you then push the door and walk out. The outside looks like a village. “Where can I find something to fight?” You wonder. “Let me walk into some more hidden places - maybe this would trigger some magical events!”

After half an hour, you finally find a mythical creature - a unicorn. “Wait why is there a unicorn? What am I seeing?” You wonder, not realising life is full of surprises. Why is not important - what important is that you are getting attacked!

The unicorn has put down some cards to attack you, and it is the unicorn’s turn! What you can do is to maximise your defence. Since you have just built the most legendary card ever possible, you can instantly defeat the unicorn once it is your turn. However, you would have to survive this turn.

“Wait I have some idea,” you are checking the manual, “I can spawn stones for defences using 1 mana! Let's see - for a card with attack ATT and defence DEF assigned to it, the damage dealt would be zero if DEF is greater or equal to ATT, and the square of difference between ATT and DEF otherwise.”

“Hmm how should I minimise the damage dealt to me with my current mana? Let me do some computations…” you speak with a bit of enthusiasm.

INPUT

The first line of input contains two integers N and M, the number of cards the unicorn has put down and your current mana, respectively.

The next line contains N integers, $A_1, A_2, \ldots, A_N$, the attack values of the N cards owned by the unicorn, from the weakest to the strongest.

OUTPUT

Output a single integer, the minimum damage dealt to you, by optimising the way you are casting the stones.

SAMPLE TESTS

	Input	Output
1	5 7 1 2 3 4 5	14

You can place stone in the way $\{0, 0, 1, 2, 4\}$. Then, the damage dealt is $1 + 4 + 4 + 4 + 1 = 14$, which can be shown to be minimum.

	Input	Output
2	5 20 1 2 3 4 5	0

You can block all attacks with 15 mana.

CONSTRAINTS

$1 \leq N \leq 10^5$

$1 \leq M \leq 10^{10}$

$1 \leq A_1 \leq A_2 \leq \cdots \leq A_N \leq 10^5$

SUBTASKS

	Points	Constraints
1	9	$N = 1$
2	17	$A_1 = A_2 = \cdots = A_N$
3	24	$N, M \leq 2000$
4	33	$N \leq 2000$
5	27	No additional constraints

TH265 Sacrifice for the Win

Score	Limit	
130	1s	256MB

You successfully survived the attack - and now it is your turn to attack the unicorn!

"Right let me play my cards - wait why some cards has blood label on it? Let me look it up in the manual… Hmm the number of blood means that number of cards I need to sacrifice on the field!"

You scroll through your cards, and thought to yourself, "seems like every card has either 0, 1 or 2 blood on them, so every time I play a card from your hand to the field, I need to sacrifice some card on the field. How can I maximise the total attack? Seems like I will win anyways but the larger the total attack value, the better the drop item is… Hmm…"

"Let me do some serious planning before playing the cards!" you thought. You left the unicorn there and just use your mana to fly on a tree to get some time to think. "Unicorn surely cannot fly", you thought. "Wait why can I fly?" You have no idea why this happens but it seems not matter anyways…

INPUT

The first line contains an integer N, the number of cards you own.

The next N lines each contains two integers A_i and B_i, the attack value and the number of cards required to be sacrificed, respectively.

OUTPUT

Output a single integer, the maximum attack value you can create.

SAMPLE TESTS

	Input	Output
1	5 1 0 2 0 3 0 2 1 5 2	8

- Play Card 1
- Play Card 2
- Play Card 3
- Play Card 4, Sacrifice Card 1
- Play Card 5, Sacrifice Card 2 and 4

Final attack value: $3 + 5 = 8$

CONSTRAINTS

$1 \leq N \leq 2 \times 10^5$

$1 \leq A_i \leq 10^9$

$0 \leq B_i \leq 2$

SUBTASKS

	Points	Constraints
1	10	$B_i = 0$ for all $1 \leq i \leq N$
2	24	$N \leq 5000$ $B_i = 0$ or 1 for all $1 \leq i \leq N$
3	18	$B_i = 0$ or 1 for all $1 \leq i \leq N$
4	21	$N \leq 200$
5	31	$N \leq 5000$
6	26	No additional constraints

TH266 Luck is Skill

Score	Limit	
150	1s	256MB

Finding the best possible strategy to hit the unicorn with your cards obtained (and of course with the legendary cards gift by the guy), you jump down from the tree and one-hit the unicorn! That is very cool and you obtain a mysterious drop from the unicorn - the horn of the unicorn - and it is shining! This is amazing but this doesn't seem to have any use, so you go back to find the guy and try to see what use it has.

“Welcome! Oh it is you the chosen one”, said the guy. “Anything good happened? I saw your great smile,” asked the guy.

“I got a shiny horn from defeating the unicorn! Let me take it out”, you take the unicorn horn out of your pocket. “Wow, this is the legendary lucky horn! With it on you, you can be much more lucky than usual!” said the guy. “Wow, really, that is great!” said you. The guy smiled and said: “Actually - luck is a skill as well! You know, there is a luck game I like to play - let me take back the practice deck…” You see back the cards with sequential numbers written on it - this time, it is from 1 to 15.

“Let me play a little game with you - the deck is randomly shuffled, then you can draw at most K times the top card. After each draw, you can either choose to stop or draw another card - and the card that you previously own is put away for now. Now your target is to maximise the number on the card you own,” explained the guy.

“Hmm interesting,” said you, “so it is a luck game but also a skill game, since you have to decide when to stop!” “Yes, exactly!” The guy replied you with a big smile, and continued: “Do you want to play this game with me - maybe just to make it more exciting, lets bet some gold coins!” Exclaimed the guy.

“Cool - let's do this!” You agreed - and the game of luck begins…

IMPLEMENTATION

You should implement the function `void be_lucky(int N, int K)`:

- N: the total number of cards in the deck.
- K: the upper limit of how many cards you can drawn from the deck.
- This procedure is called at most 1000 times per run.

You can call the following functions:

`int draw_card()`

- Draw a card from the deck.
- This can be called at most K times within the same call to `be_lucky`.

Please note that the cards that will be drawn is pseudo-randomly generated. Also, the cases are deterministic in the sense that the test cases are pre-generated and fixed. That means if you submit the same code twice which perform the exact same behaviour, you should get the same score.

TEMPLATE

```
#include "luck.h"

void be_lucky(int N, int K) {
    draw_card();
}
```

SCORING

On each test case, you will receive zero score if your program:

- does not exit correctly, or
- violates any rules mentioned in the section "Implementation".

Otherwise, your score will depend on the sum of all the numbers over the rounds, X. It also depends on the sum of all numbers over the rounds of the guy (which obviously refers to the problem author).

$$score = \begin{cases} 10 & \text{if } X \geq Y \\ 10 \times \left(\dfrac{X - 7500}{Y - 7500} \right)^{10} & \text{if } 7500 < X < Y \\ 0 & \text{otherwise.} \end{cases}$$

There are exactly 15 test cases, each worths 10 points, corresponding to $K = 1, 2, \ldots, 15$ respectively. Your score for this problem is simply the sum of scores over all the test cases.

SAMPLE GRADER

The sample grader reads the input in the following format:
Line 1: N K T, where T is the number of times you wish to run the game.

If your solution is correct, the output will be in the following format:
Line 1: `OK`
Line 2: X, where X is the total score you get from the T runs of the game.

If your solution is wrong, the output will be in the following format depending on the reason:
Line 1: `WA`
Line 2: Simple description of the reason for the wrong answer

Compilation Command: `g++ sample_grader.cpp luck.cpp -o luck`
Run Command: `./luck` (Linux or Mac) OR `luck` (Windows)

SAMPLE TESTS

	Input	Output
1	15 8 10	OK 76

CONSTRAINTS

$N = 15$

$1 \leq K \leq N$

$T = 1000$

TH267 Boss Fight

Score	Limit	
180	1s	256MB

"Wow I am actually impress by you," said the guy, "I don't think anyone can defeat me in my own game!"

"Haha, maybe luck really matters," you said.

The guy smiled. "Ding Ding Ding", the phone on the table rang. "Excuse me", the guy picked up the phone, and the happiness slowly fades from his face. "I got an urgent call that the big boss is targeting this village - and will be here in 4 hours - account the time of our discussion, only 3 hours 59 minutes is left! We need you - the chosen one - to help us to defend this village!"

You shocked, but after letting the emotion sank in, you made the good decision (since otherwise the story ends): "Alright, let me help you guys. Anything do I have to prepare for this - and does the boss has anything special?" You asked.

"Good question - the special part of the boss is that the boss is elemental - do you know about the 'conquer order' among elements?" The guy said.

"Not really - how does that work? I just remember the three pure elements - Fire, Wood and Water and apparently you can mix them together," you said.

The guy gives you a thumb, and said, "that is great! You still remember what we have talked about at the beginning. However, the boss can only be fight by pure element cards. Meanwhile, there is an extra buff / debuff due to elements conquer each other. Hmm... the conquer order is Fire, Wood, Water. That means if you use Fire card to attack Wood card, the attack strength is doubled, and the same holds for using Wood card to attack Water card or using Water card to attack Fire card. If you attack in the opposite order - then the attack strength is halved! Of course, if you attack with the same element, the attack strength is not affected." The guy explained.

"Alright - so how these elements work on the boss?" You are confused, because you not exactly sure how this works.

The guy thought for a second, then say: "There will be many round of the boss fight, since, well, the boss is very strong. Each round, the boss will play some amount of elemental cards, which you can then attack each of the card with one of your card, which the total attack is the sum over all the cards."

"Alright, let me think about a strategy to defeat the boss!" You said confidently, and the final struggle begins...

INPUT

The first line contains two integers N and Q, the number of cards you currently own and the number of rounds of the boss fight, respectively.

The next N lines each contains a character and an integer C_i and A_i, describing the element and the attack value of the card, respectively. If $C_i =$ `R`, the card is Fire. If $C_i =$ `G`, the card is Wood. Otherwise, if $C_i =$ `B`, the card is Water.

Then, Q lines follow, describing the rounds of the boss fight. Each line contains three integers R_j, G_j and B_j, representing the number of cards of the Fire, Wood and Water elements the boss has placed in the j^{th} round.

OUTPUT

Output Q lines, the maximum attack value you can create from the cards after considering the elemental effects for each round.

SAMPLE TESTS

1

Input	Output
5 3 R 2 R 4 G 8 B 4 B 6 3 0 2 2 2 1 0 0 5	41 48 29

In the first round, the boss placed three Fire cards and two Water cards.

- Card 1: Attack Water card. Attack: $2 \times 0.5 = 1$.
- Card 2: Attack Fire card. Attack: $4 \times 1 = 4$.
- Card 3: Attack Water card. Attack: $8 \times 2 = 16$.
- Card 4: Attack Fire card. Attack: $4 \times 2 = 8$.
- Card 5: Attack Fire card. Attack: $6 \times 2 = 12$.

Total attack: $1 + 4 + 16 + 8 + 12 = 41$

2

1 3 R 2 1 0 0 0 1 0 0 0 1	2 4 1

CONSTRAINTS

$1 \leq N, Q \leq 2 \times 10^5$

$C_i =$ `R`, `G` or `B`

$2 \leq A_i \leq 10^9$

A_i is even

$0 \leq R_j, G_j, B_j \leq N$

$R_j + G_j + B_j = N$

SUBTASKS

	Points	Constraints
1	17	$N, Q \leq 5000$ $C_i =$ `R` for all $1 \leq i \leq N$
2	35	$N \leq 100$ $Q \leq 10$
3	23	$N \leq 5000$ $Q \leq 10$
4	32	$N, Q \leq 5000$
5	28	$A_i = 2$ for all $1 \leq i \leq N$
6	45	No additional constraints

TH268 Dream or Reality?

Score	Limit	
210	1s	256MB

After a hour or two of serious fight, you finally defeat the boss! “Woah woah woah it drops time rewind item! I can rewind back to the midnight of the previous day”, you exclaimed. “Why would I ever use it though?” You then put the time rewind item back to your pocket.

“Wow what a day! Let me have more read on the manual before I sleep...” you thought. You randomly find a table to start reading. You slowly feel tired, and your head fell onto the table...

“Urgh! Wait why my surrounding has changed and I have backed to the reality? I thought I am the main character after reincarination”, you thought, “Oh no I am originally studying for the book test! Oh no!” You touch your pocket, surprised to find the time rewind item. “Wow I still have the time rewind item! Let me pass the book test using it!”

IMPLEMENTATION

You should implement the function `void book_test(int` N`, int` P`)`:

- N: the number of multiple-choice (A, B, C, D) questions in the book test.
- P: the number of MC questions you need to get correct to pass the book test.
- This procedure is called at most 100 times per run.

You can call the following functions:

`int test(std::string` S`)`

- S: Your MC answers, represented by a string of length N that consists of `A`, `B`, `C` and `D`.
- This can be called at most 80 times within the same call to `book_test`.
- This returns the number of MC questions you get correct.

Please note that the MC answers that will be drawn is pseudo-randomly generated. Also, the cases are deterministic in the sense that the test cases are pre-generated and fixed. That means if you submit the same code twice which perform the exact same behaviour, you should get the same score.

TEMPLATE

```
#include "test.h"

void book_test(int N, int P) {
    std::string S(N, ' ');
    for (int i = 0; i < N; i++) {
        S[i] = 'A';
    }
    int c = test(S);
}
```

SCORING

On each test case, you will receive zero score if your program:

- does not exit correctly, or
- violates any rules mentioned in the section "Implementation", or
- fails the book test everytime.

Otherwise, your score will depend on K, the maximum number of calls to `test` over all cases in that test case.

$$score = \begin{cases} 210 & \text{if } K \leq 12 \\ 90 + 15 \times (20 - K) & \text{if } 12 < K \leq 20 \\ 50 + 4 \times (30 - K) & \text{if } 20 < K \leq 30 \\ 80 - K & \text{if } 30 < K \leq 80 \\ 0 & \text{otherwise.} \end{cases}$$

Your score for this problem is the minimum score over all the test cases.

SAMPLE GRADER

The sample grader reads the input in the following format:
Line 1: N P T, where T is the number of cases you want to try.

If your solution is correct, the output will be in the following format:
Line 1: `OK`
Line 2: K

If your solution is wrong, the output will be in the following format depending on the reason:
Line 1: `WA`
Line 2: Simple description of the reason for the wrong answer

Compilation Command: `g++ sample_grader.cpp test.cpp -o test`
Run Command: `./test` (Linux or Mac) OR `test` (Windows)

SAMPLE TESTS

	Input	Output
1	20 10 100	OK 16

CONSTRAINTS

$N = 20$

$P = 10$

$T = 100$