


Tsuen Wan Government Secondary School Olympiad in Informatics Team

St. Mark's School Programming Team

Ying Wa Girls' School Programming Team

2025/26 HKGOI Mock

EDITORIAL

Note

Solutions of

- TMG265 Water
- TMG266 Son of Rainbow

will be provided later.

Please check https://oi.twgss.org/contests time to time for updates.

Awards

In HKGOI 2024/25,

• **Gold**	4	Cut-off 417
• Gold + **Silver**	12	Cut-off 378
• Gold + Silver + **Bronze**	24	Cut-off 212
• Gold + Silver + Bronze + **HM**	32	Cut-off 141
• Contestants with positive scores	61	

Awards

In this mock, we use the same proportion (rounding off) with respect to contestants with positive scores to distribute awards.

In 2025/26 HKGOI Mock,

• **Gold**	2	Cut-off 481
• Gold + **Silver**	6	Cut-off 417
• Gold + Silver + **Bronze**	13	Cut-off 281
• Gold + Silver + Bronze + **HM**	17	Cut-off 233
• Contestants with positive scores	32	

Rummikub

TMG261

Statement

Given three tiles from Rummikub, check if they can form a valid set.

There is at most a **Joker** which can be treated as any normal tile.

The tile is first described by its type S_i.

- If $S_i =$ `Joker`, the tile is a **Joker**.
- Otherwise, S_i is its colour and the number of it is given by N_i.

Statement

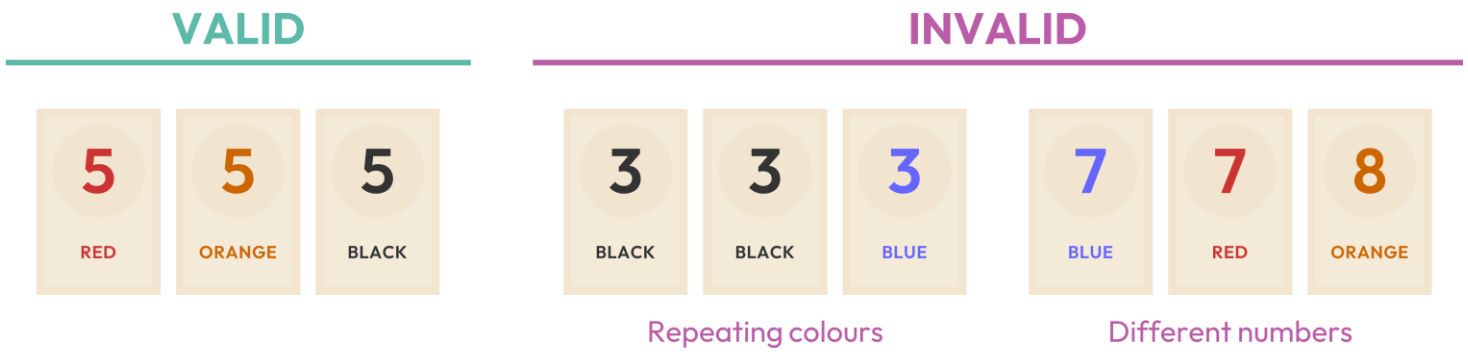
There are two types of valid sets:

Run Three tiles of the same colour with consecutive numbers.

Statement

There are two types of valid sets:

Group Three tiles with the same number but different colours.



Statistics

Attempts
32

Max	Mean	Std Dev
100	86.344	26.972

LQ	Median	UQ
82.5	100	100

19 | 31	21 | 29	25 | 29	17 | 24	18 | 24

First Solves			
	TWGSS	Wong Ho Ki	0:12:21
	YWGS	Leung Hilary	0:14:53
	SMS	Yeung Ching Yin	0:15:39

Subtasks

	Points	Constraints
1	19	$S_i \neq$ `Joker` $N_i = 5$
2	21	$S_i =$ `Blue`
3	25	$S_i \neq$ `Joker`
4	17	$S_3 =$ `Joker`
5	18	No additional constraints

Subtask 1

$S_i \neq$ `Joker`
$N_i = 5$

Clearly, **Run** is impossible to be constructed as all numbers are the same.

Refer to the definition of **Group**, colours must be pairwise different.

Therefore, you only have to check if $S_1 \neq S_2$ and $S_1 \neq S_3$ and $S_2 \neq S_3$.

Expected Score	**19**

Subtask 2

$S_i =$ `Blue`

Similarly, **Run** is impossible to be constructed as all colours are the same.

Refer to the definition of **Group**, numbers should be consecutive.

Luckily, there is a useful constraint: $1 \leq N_1 \leq N_2 \leq N_3 \leq 13$

Therefore, if numbers are consecutive,

$N_1 + 1 = N_2$ and $N_1 + 2 = N_3$ and $N_2 + 1 = N_3$.

In practice, any two of the above conditions are sufficient to determine if they are consecutive.

Expected Score	**21**

Subtask 3

$S_i \neq$ `Joker`

Now, these tiles can either form **Run** or **Group**.
Let's determine which set to be chosen.

Run: Three tiles of the same colour with consecutive numbers.
Group: Three tiles with the same number but different colours.

Subtask 3

$S_i \neq$ `Joker`

- If these tiles are in same colours, they can either form **Run** or be Invalid.
- If they are in same numbers, they can either form **Group** or be Invalid.
- Otherwise, they must be Invalid.

↓

- If $S_1 = S_2$ and $S_1 = S_3$ and $S_2 = S_3$, run the solution of Subtask 2.
- If $N_1 = N_2$ and $N_1 = N_3$ and $N_2 = N_3$, run the solution of Subtask 1.
- Otherwise, output `Invalid`.

Expected Score	**65**

Subtask 4

$S_3 =$ `Joker`

The third tile must be a **Joker**, which can be treated as any normal tile.

Same as Subtask 3, we would like to determine whether **Run** or **Group** would be formed.

As **Joker** can be treated as any normal tile, we don't need to consider it while using the same logic.

Subtask 4

$S_3 =$ `Joker`

If the first two tiles are in same colour, they can only form **Run** or Invalid.

How to check if the first two tiles can form consecutive numbers?

- If the first two tiles are consecutive, it is clear that **Joker** can form consecutive sequence with them. Hence, $N_1 + 1 = N_2$.
- Otherwise, they can be consecutive if and only if **Joker** is treated as the middle tile. Hence, $N_1 + 2 = N_2$.

So the condition is just $N_1 + 1 = N_2$ or $N_1 + 2 = N_2$.

Subtask 4

$S_3 =$ `Joker`

If the first two tiles are in same numbers, they can only form **Group** or Invalid.

For **Group**, all tiles should be in different colours.

Joker can always be in different colours with other two tiles.

Therefore, the condition is simply $S_1 \neq S_2$.

Expected Score	**17**

In fact, this subtask is easier than Subtask 3.

Full Solution

If there is no **Joker**, we can use Subtask 3 to solve it.

Otherwise, if we find a **Joker**, it is actually the same of Subtask 4.

However, input must be carefully handled as N_i does not exist if $S_i = \texttt{Joker}$.

You need to skip the input of N_i if $S_i = \texttt{Joker}$ and handle the position of the other tiles well.

Expected Score	**100**

Common Mistakes

Spelling `Valid` as `Vaild`

100 marks → 0 marks

```
if (i == 0){
    if (a[1] == a[2]){
        if (b[1] + 1 == b[2] || b[1] + 2 == b[2]){
            cout << "Vaild";
            return 0;
        }
    }else{
        if (b[1] == b[2]){
            cout << "Vaild";
            return 0;
        }
    }
}else if (i == 1){
```

Result

Subtask	Verdict	Score
1	Wrong Answer	0 /19
2	Wrong Answer	0 /21
3	Wrong Answer	0 /25
4	Wrong Answer	0 /17
5	Wrong Answer	0 /18

Always **copy keywords from problem statement** if it is used for conditional matching and output!

Little Helper

TMG262

Statement

Given tense T, which is either `present`, `past` and `continuous`.

For each word W, the parts of speech is determined by the first letter W_1:

- If W_1 is in uppercase, it is a noun.
- If W_1 is in lowercase, it is a verb.

There are N words.

The i^{th} original word and student's answer is Q_i and A_i respectively.

Statement

For the i^{th} word, the student is correct if:

- The parts of speech (noun / verb) of Q_i and A_i are the same.

  AND

- If it is a noun, A_i matches Q_i case-sensitively.
- If it is a verb, A_i matches Q_i followed by S case-insensitively.
  - If $T =$ `present`, S is empty.
  - If $T =$ `past`, $S =$ `ed`.
  - If $T =$ `continuous`, $S =$ `ing`.

Count the number of words that the student is correct.

Statistics

Attempts
22

Max	Mean	Std Dev
100	73.818	30.83

LQ	Median	UQ
60	89	100

6 | 22	14 | 19	21 | 19	13 | 15	19 | 18	16 | 14	11 | 6

First Solves Highest Scores			
	YWGS	Cheung Suet Nam	0:33:49
	TWGSS	Nip Kit Fung	1:39:51
	SMS	Yeung Ching Yin	Score 89

Subtasks

	Points	Constraints
1	6	$N = 1$ Q_1, A_1 starts with an uppercase English letter (`A`-`Z`)
2	14	Q_i, A_i starts with a uppercase English letter (`A`-`Z`)
3	21	$T =$ `present` length of Q_i = length of A_i Q_i consists of `A` and `a` only Q_i, A_i starts with `a`
4	13	$Q_i =$ `a` A_i starts with `a` A_i consists of lowercase English letters (`a`-`z`) only
5	19	$T =$ `present` length of Q_i = length of A_i Q_i, A_i starts with a lowercase English letter (`a`-`z`)
6	16	Q_i, A_i starts with a lowercase English letter (`a`-`z`)
7	11	No additional constraints

Subtask 1

$N = 1$

Q_1, A_1 starts with an uppercase English letter (`A` - `Z`)

There is only one word and it must be a noun.

As noun requires exact case-sensitive match,
we can just check if $Q_1 = A_1$.

Output `1` if so, otherwise output `0`.

Expected Score	6

Subtask 2

Q_i, A_i starts with a uppercase English letter (`A`-`Z`)

There maybe more than one word now, but all words are still noun.

Just use a loop for input and store the data into an array.
Keep track of a counter ans, initially as 0.
Then for each word, if $Q_i = A_i$, add ans by 1.
Finally output ans.

Expected Score	20

Subtask 3

$T =$ `present`
length of Q_i = length of A_i

Q_i consists of `A` and `a` only
Q_i, A_i starts with `a`

All words are verb in present tense.

For each word, the length of Q_i and A_i is the same,
while Q_i only consists of `A` and `a`.

As verb checking is case-insensitive, for each A_i,
we can loop through all of its character and see if it equals to `A` or `a`.

If all characters are `A` or `a`, the word is correct. Otherwise it is wrong.

Expected Score	**21**	Cumulative	41

Subtask 4

$Q_i =$ a
A_i starts with a

A_i consists of lowercase English letters (a - z) only

We only have to check whether each A_i is correct in the given tense T:

- If $T =$ present, we check if $A_i =$ a case-insensitively.
- If $T =$ past, we check if $A_i =$ aed case-insensitively.
- If $T =$ continuous, we check if $A_i =$ aing case-insensitively.

We can implement this by just checking all possibilities,
for example for $T =$ continuous,
we check if $A_i =$ aing, ainG, aiNg, aiNG, aIng, aInG, aINg, aING.

Expected Score	13	Cumulative	54

Subtask 5

$T =$ `present`
length of Q_i = length of A_i

Q_i, A_i starts with a lowercase English letter (`a` - `z`)

All words are verb in present tense.

For each word, the length of Q_i and A_i is the same.

We need to check if each character in Q_i and A_i matches case-insensitively.

For each character, we can change it to lowercase by using the `to_lower()` function in C++.

After changing all characters in Q_i and A_i to lowercase,
if all characters in a word matches, the word is correct.

Expected Score	40	Cumulative	73

Subtask 6

Q_i, A_i starts with a lowercase English letter (`a` - `z`)

All words are verb.

Now the tense is unknown and Q_i and A_i can be in different length.

First, we can add `ed` or `ing` to all Q_i depending on the tense.

Then, we can check if the length of Q_i and A_i is the same.

Finally, we can use the approach of Subtask 5, changing all characters into lowercase for case-insensitive checking of same length.

Expected Score	69	Cumulative	89

Full Solution

We have handled the situation of all nouns and all verbs.

Therefore, for each word:

- First check if the parts of speech of Q_i and A_i are the same.
- If it is a noun, implement the checking in Subtask 2.
- If it is a verb, implement the checking in Subtask 6.

Expected Score	**100**

Girls and Boys

TMG263

Statement

There are N students, each student is either a girl (`G`) or a boy (`B`).

Find the minimum number of swaps of adjacent members so that all girls are in one end and all boys are in another end .

Example:

- `GGBGBG` → `GGBGGB` → `GGGBGB` → `GGGGBB` 3 swaps
- `BGBBG` → `BBGBG` → `BBBGG` 2 swaps

Statistics

Attempts
29

Max	Mean	Std Dev
100	55.414	41.201

LQ	Median	UQ
14	44	100

8 | 26	12 | 20	16 | 19	20 | 14	14 | 14	19 | 13	11 | 12

First Solves			
	TWGSS	Wong Ho Ki	0:26:40
	YWGS	Leung Hilary	0:31:01
	SMS	Yeung Ching Yin	0:50:49

Subtasks

	Points	Constraints
1	8	number of `B` in $S = 0$
2	12	$1 \leq N \leq 3$
3	16	number of `B` in $S \leq 1$
4	20	number of `B` in $S \leq 2$
5	14	N is even $S_i \neq S_{i+1}$
6	19	$1 \leq N \leq 10$
7	11	No additional constraints

Subtask 1

number of `B` in $S = 0$

There is no boys, so all girls are already at the both end.

Why? Because it's GOI!

No swap is needed.

Output `0`.

Expected Score	8

Subtask 2

$1 \leq N \leq 3$

There are only 8 possible combinations: `GGG`, `GGB`, `GBG`, `GBB`, `BGG`, `BGB`, `BBG`, `BBB`.

After writing all combinations,
we can observe that only when $S =$ `GBG` or `BGB` needs swaps.

Output `1` when $S =$ `GBG` or `BGB`, or output `0` otherwise.

Expected Score	12	Cumulative	20

Subtask 3

number of `B` in $S \leq 1$

If there is 0 `B` in S, use the solution of Subtask 1.
Otherwise, the only boy should either move to the leftmost or the rightmost.

Let p be the position of the boy (in 1-base), i.e. $S_p =$ `B`.
It requires either $p - 1$ swaps or $N - p$ swaps.
Output the smaller one.

Expected Score	24	Cumulative	36

Subtask 4

number of `B` in $S \leq 2$

If there is 0 or 1 `B` in S, use the solution of Subtask 3.
Otherwise, the 2 boys should either move to the leftmost or the rightmost.

Let l and r be the positions of the two boys where $l < r$.

- Moving all boys to the left requires $(l-1)+(r-2)$ swaps.
- Moving all boys to the right requires $(N-r)+(N-l-1)$ swaps.

Output the smaller one.

Expected Score	44	Cumulative	56

Observation

We only swap a boy and a girl each time because swapping two boys or two girls are meaningless.

So if there are g girls to the left of a boy.

Moving the boy to the leftmost costs g additional swaps as each swap guarantee that a girl will be swapped to the right.

Similarly, if there are b boys to the left of a girl.

Moving the girl to the leftmost costs b additional swaps as each swap guarantee that a boy will be swapped to the right.

Subtask 5

N is even
$S_i \neq S_{i+1}$

$S =$ GBGB…GB or BGBG…BG

There are $\frac{N}{2}$ boys and $\frac{N}{2}$ girls.

We have only two cases, moving all boys to the left or all girls to the left.
It can be easily determined by whether the leftmost person is boy or girl.

- If he is boy, moving all boys to the left will be more cost-effective.
- If she is girl, moving all girls to the left will be more cost-effective.

Subtask 5

N is even
$S_i \neq S_{i+1}$

Assume that we are moving all boys to the left.

There would be $i-1$ girls to the left of the i^{th} boy.

Sum them up: $0+1+2+\cdots+\left(\frac{N}{2}-1\right)$.

By Maths, the answer is $\frac{\left(\frac{N}{2}-1\right)\left(\frac{N}{2}\right)}{2}=\frac{(N-2)N}{8}$.

Expected Score	14	Cumulative	70

Full Solution

Now, S does not have any pattern.

However, we can count the number of boys/girls before each girl/boy using for-loop.

Depend on implementation, it can be done in O(N) or $O(N^2)$.

Check both cases : moving all boys/girls to the left.

Subtask 6 is just a safety net in case the implementation is not complete.

Expected Score	**100**

Take-home Task

What if we can use crop the photo twice?

i.e. The arrangement is valid when all girls are consecutive.

e.g. `BBBGGGGGGGBB` is valid too

Cat Warn

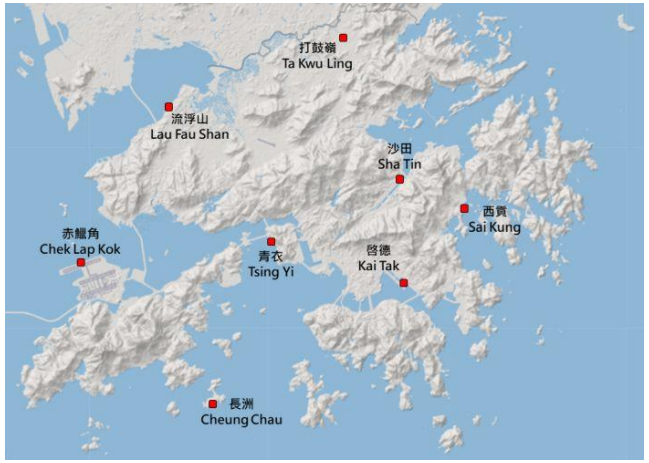
TMG264

Story

In 2006, Typhoon *Prapiroon (派比安)* stroke Hong Kong. The wind speed at most part in Hong Kong was very strong.

But according to the old standard of issuing TC Signals (which only considered wind speed at Victoria Harbour, only TC Signal No. 3 was in force for the whole time and Signal No. 8 was not issued.

This caused many criticism to HKO and a new standard of issuing TC Signals by considering 8 weather stations' wind speeds was implemented since 2007.



Statement

There are four Warning Signals:

	Standby	No requirement.
	Strong Wind	**At least** 4 weather stations observe wind speeds **higher or equal to** 41.
	Gale or Storm	**At least** 4 weather stations observe wind speeds **higher or equal to** 63.
	Hurricane	**At least** 1 weather station observe wind speeds **higher or equal to** 118.

If there are multiple signals with their requirements met, the highest one will be in force.

Statement

There are N minutes.

Given the i^{th} minute's wind speed of the eight stations be $S_{i,1}, S_{i,2}, \ldots, S_{i,8}$.

A signal should be in force until its requirement is last met, even if there exists some time that the requirement is not met in between, unless replaced by a higher signal. The sequence of the in-force Warning Signals can only go from low to high, then back to low for at most once.

VALID

INVALID

Find the number of minutes that each Warning Signal is in force.

Statistics

Attempts
22

Max	Mean	Std Dev
100	50.773	32.838

LQ	Median	UQ
35	35	90

14 | 21	21 | 16	18 | 8	17 | 9	20 | 7	10 | 5

First Solves Highest Scores			
	TWGSS	Wong Ho Ki	0:43:44
	YWGS	Leung Hilary	1:19:40
	SMS	Yeung Ching Yin	Score 69

Subtasks

	Points	Constraints
1	14	$N = 1$ $S_{1,1} = S_{1,2} = \cdots = S_{1,8}$
2	21	$N = 1$
3	18	$40 \leq S_{i,1} = S_{i,2} = \cdots = S_{i,8} \leq 41$
4	17	$S_{i,1} = S_{i,2} = \cdots = S_{i,8}$ $S_{N,1} = S_{N,2} = \cdots = S_{N,8} = 118$
5	20	$S_{i,1} = S_{i,2} = \cdots = S_{i,8}$
6	10	No additional constraints

Subtask 1

$N = 1$
$S_{1,1} = S_{1,2} = \cdots = S_{1,8}$

There is only one minute and wind speeds for all stations are the same.

Just check $S_{1,1}$!

If $S_{1,1} \geq 118$,	Signal `10` is issued!	→ Output `0 0 0 1`
Else if $S_{1,1} \geq 63$,	Signal `8` is issued!	→ Output `0 0 1 0`
Else if $S_{1,1} \geq 41$,	Signal `3` is issued.	→ Output `0 1 0 0`
Else	Singal `1` is issued.	→ Output `1 0 0 0`

Expected Score	**14**

Subtask 2

$N = 1$

Now the wind speeds for different stations maybe different!

Just implement the Signal issuing rules!

Count the number of stations that achieve the speed limit of each signal.

Then for Signal `10`, `8` and `3`, if the number of stations reaches its threshold (`1` station for `10`, `4` stations for `8` and `3`), issue that signal and break.

Otherwise, issue Signal `1`.

Expected Score	**35**

Subtask 3

$$40 \leq S_{i,1} = S_{i,2} = \cdots = S_{i,8} \leq 41$$

Now there maybe more than one minute but
wind speeds for all stations are the same for each minute.

Let say the qualified Signal for the i^{th} minute be Q_i.

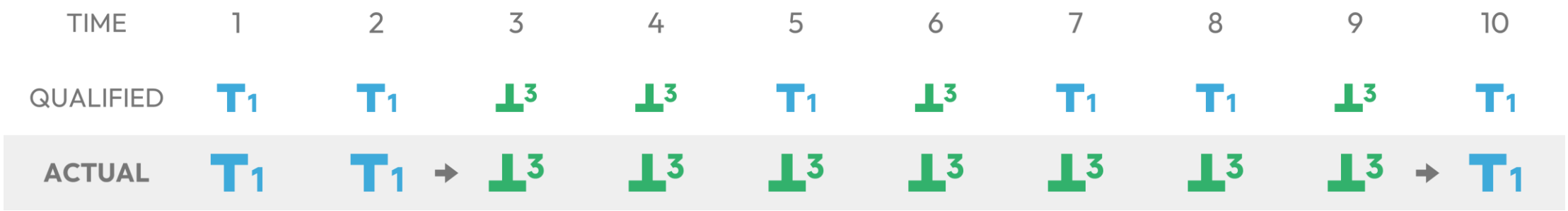
Obviously, if $S_{i,1} = 41$, $Q_i = 3$. Otherwise, $Q_i = 1$.

Now we get a sequence consists of 3 and 1 only,
how do we identify the actual Signal that is in-force for each minute.

Subtask 3

$$40 \leq S_{i,1} = S_{i,2} = \cdots = S_{i,8} \leq 41$$

Let's consider an example:



Can you observe anything?

If you can't, observe again.

Subtask 3

$$40 \leq S_{i,1} = S_{i,2} = \cdots = S_{i,8} \leq 41$$

Turns out, between the first and the last Signal `3`,
all time between them need to be Singal `3`.

Other times are Signal `1`.

Remember to also handle the case that there is no Signal `3`.

But somehow your code may slip through and get Accepted if you forgot to do so.

Expected Score	**18**	Cumulative	53

Subtask 4

$$S_{i,1} = S_{i,2} = \cdots = S_{i,8}$$
$$S_{N,1} = S_{N,2} = \cdots = S_{N,8} = 118$$

The wind speeds for all stations are the same for each minute.

The last minute's qualified Singal is `10`.

We first need to identify the qualified Signal for each minute by repeatedly implementing the Subtask 1 solution.

Subtask 4

$$S_{i,1} = S_{i,2} = \cdots = S_{i,8}$$
$$S_{N,1} = S_{N,2} = \cdots = S_{N,8} = 118$$

Let's consider another example:

TIME	1	2	3	4	5	6	7	8	9	10
QUALIFIED	T1	⊥3	T1	▲8	⊥3	▲8	T1	✚10	⊥3	✚10
ACTUAL	T1 →	⊥3	⊥3 →	▲8	▲8	▲8	▲8 →	✚10	✚10	✚10

Notice anything?

The Signals are "going up"!

Subtask 4

$$S_{i,1} = S_{i,2} = \cdots = S_{i,8}$$
$$S_{N,1} = S_{N,2} = \cdots = S_{N,8} = 118$$

As the Signals never goes down, we can conclude that for each pair of adjacent minutes, the Signal of the latter one must not be lower than the earlier one.

Let the qualified, actual Signal for the i^{th} minute be Q_i, A_i respectively.

At first, we set $A_i = Q_i$.

Then to implement this idea,
we can just set $A_i = Q_{i-1}$ if $Q_i < Q_{i-1}$ for all $2 \leq i \leq N$.

Expected Score	17	Cumulative	70

Subtask 5

$$S_{i,1} = S_{i,2} = \cdots = S_{i,8}$$

Now we only guarantee that the wind speeds for all stations are the same for each minute, which is… unhelpful (as you should know how to identify Signals if you have come to this Subtask).

From Subtask 4, we already handle the “going up” part.

That means what’s remaining is just the “going down” part.

Before implementing, we have two questions to solve:

- When should the “going down” part start?
- How do we handle “going down”?

Subtask 5

$$S_{i,1} = S_{i,2} = \cdots = S_{i,8}$$

Let's tackle the second question first: How do we handle "going down"?

Simple! Just do Subtask 4, but in reverse!

During "going down", for each pair of adjacent minutes, the Signal of the earlier one must not be lower than the latter one.

Let the qualified, actual Signal for the i^{th} minute be Q_i, A_i respectively.

To implement, we can set $A_i = Q_{i+1}$ if $Q_i < Q_{i+1}$ for all i in "going down" part.

Subtask 5

$$S_{i,1} = S_{i,2} = \cdots = S_{i,8}$$

Now, for the first question: When should the “going down” part start?

We can take inspirations from Subtask 3!

As we know between the “going up / down” part must be the highest signal throughout, instead of finding the first and last Signal 3,

we want to find the first and last highest Signal appeared.

Therefore, before the first highest Signal will be “going up”,
and after the last highest Signal will be “going down”.

Subtask 5

$$S_{i,1} = S_{i,2} = \cdots = S_{i,8}$$

In fact, we can combine both ideas and have a rather clean solution.

Let the qualified, actual Signal for the i^{th} minute be Q_i, A_i respectively.

- First, we loop through Q to find out the highest qualified Signal x.
- Then, for i from 1 increasing until the first $Q_i = x$,
  we set $A_i = Q_{i-1}$ if $Q_i < Q_{i-1}$ to implement the “going up” part.
- Next, for i from N decreasing until the first $Q_i = x$,
  we set $A_i = Q_{i+1}$ if $Q_i < Q_{i+1}$ to implement the “going down” part.

Subtask 5

$$S_{i,1} = S_{i,2} = \cdots = S_{i,8}$$

This algorithm successfully handles the “going up / down” parts.

However, when implementing it yourself, there are something to note:

- Check the bounds (by making up some samples or have a very clear mind) carefully. Do not get or check too few / too many data.
  Also, Accessing out-of-bounds data will give you Runtime Error or Wrong Answer.
- Remember to handle that highest Signal should remain in-force between the “going up / down” parts.

Expected Score	**69**	Cumulative	90

Full Solution

Subtask 5 actually already grants you all things needed for the full solution.

With wind speeds for all stations no longer the the same for each minute,
we just have to again implement the Signal issuing rules,
which you should have already done in Subtask 2.

Add this part into your Subtask 5 solution,
then you will successfully solve this problem!

Expected Score	**100**

Takeaways

Sometimes subtasks are not just for “watering” scores,
time to time they are useful for you to think of the full solution.

In this problem,

Subtask 1	Checks if you read the problem.
Subtask 2	Implements the Signal issuing rules.
Subtask 3	Observation: find the first and last high Signal.
Subtask 4	Implements the “going up” part.
Subtask 5	Extends Subtask 4 solution + observation in Subtask 3.
Full Solution	Combine Subtask 2 and 5.

Fun Fact

One participant got 90 marks…?

Subtask 2 and 5 are solved but not the full solution…?

Subtask	Verdict	Score
1	Accepted	14 /14
2	Accepted	21 /21
3	Accepted	18 /18
4	Accepted	17 /17
5	Accepted	20 /20
6	Wrong Answer	0 /10

Upon closer inspection, seems the mistake of the code is forgetting to initialise variables (arrays).

It passed Subtask 5 is just a result of weak test cases…

However, the score will not be changed this time.

No more rejudging and 搬龍門 like normal TWGSS contests!

As in HKOI / HKGOI, if you are lucky enough to pass, you pass.

HKOI / HKGOI won't have this kind of weak cases though.