TWGSS
OI Team

2022/23 Team Selection Test

Editorial

Statistics

	Problem	Attempts	Full	Max	Mean	Std Dev	LQ	Median	UQ	Subtasks	First Solve	
TS230	Yen Era	20	20	20	18.3	5.187	20	20	20	6 | 19 / 14 | 18	Hui Ka Wai	00:03:57
TS231	Averagers: Scoregame	14	30	30	17.357	12.81	0	21	30	13 | 9 / 8 | 9 / 9 | 6	Nip Kit Fung	00:16:21
TS232	Photo Shop	13	40	40	19.692	19.024	0	11	40	5 | 8 / 6 | 7 / 9 | 6 / 13 | 6 / 7 | 6	Chen Bai An	01:14:19
TS233	Colour Picker	9	45	45	17.333	15.797	6	12	31.5	12 | 6 / 8 | 3 / 10 | 3 / 11 | 2 / 4 | 2	Chan Sui Ki	00:39:13
TS234	Story Copier	4	65	13	3.25	5.629	0	0	6.5	13 | 1 / 9 | 0 / 17 | 0 / 11 | 0 / 14 | 0 / 1 | 0		
		20	**200**	**135**	**47**	**42.189**	**20**	**26.5**	**73**			

TWGSS
OI Team

Yen Era

TS230

Style Inspired by

米田 優峻 – 飛行機旅行 解説 (JOI 2021/2022 春季トレーニング合宿)

https://imo2023.jp/

- Daniel is really going to Chiba, Japan for the 64th IMO!
- The competition will be held on July.
- Let's look forward to get some souvenirs from him!

0 Problem Statement

Very simple!

Input	Letter Code & Era Year
Output	Gregorian calendar year

0 Problem Statement

What is era?

- Have you ever learnt something called 年號 in Chinese History? It is basically this thing.
- In Japan, there would be one era name (元号) for each emperor.
- The current one is 令和, which begins in 2019.

Fun Quiz: What was the first ever 年號 in Chinese History? (Hint: It was set up by 漢武帝)

Answer: 建元 (BC140-BC135) Mr. Lim would be sad if you didn't get this!

Subtask 1

It is guaranteed that the Year Code is “R”

Solution

Method 1

Using Maths

Input	R x
令和	x 年
Output	$2018 + x$

Solution

Method I

Using Maths

Input R x

令和 x年

Output $2018 + x$

How can we do this subtask if we don't know Maths?

(Are you kidding?)

Solution

How many cases are there in total?

Input	R 1	R 2	R 3	R 4	R 5
令和	元年	二年	三年	四年	五年
Output	2019	2020	2021	2022	2023

Solution

How many cases are there in total?

Input	R 1	R 2	R 3	R 4	R 5
令和	元年	二年	三年	四年	五年
Output	2019	2020	2021	2022	2023

There are only 5 cases!

Solution

Method 1

Using Maths

Method 2

Using the beauty of hardcode

Input R x

```
if (x == 1)
        cout << "2019" << endl;
if (x == 2)
        cout << "2020" << endl;
if (x == 3)
        cout << "2021" << endl;
if (x == 4)
        cout << "2022" << endl;
if (x == 5)
        cout << "2023" << endl;
```

I Solution

Method 1

Using Maths

Input

R x

```
if (x == 1)
        cout << "2019" << endl;
if (x == 2)
        cout << "2020" << endl;
if (x == 3)
        cout << "2021" << endl;
if (x == 4)
        cout << "2022" << endl;
if (x == 5)
        cout << "2023" << endl;
```

Method 2

Using the beauty of hardcode

Score: 6/20

Full Solution

Subtask 2

How many cases are there in total?

Code	M	T	S	H	R
Era	明治	大正	昭和	平成	令和
Cases	1868-1912 45	1912-1926 15	1926-1989 64	1989-2019 31	2019-2023 5

How many cases are there in total?

Total number of cases: 160

Code	M	T	S	H	R
Era	明治	大正	昭和	平成	令和
Cases	1868-1912 45	1912-1926 15	1926-1989 64	1989-2019 31	2019-2023 5

2 Explanation

- With 160 cases, it becomes very time-consuming for us to hardcode everything.

- Therefore, using Maths is the only way out!
- But how?

2 Solution

Input c x

1 Determine the Era

2 Output the Answer

```
if (c == 'R')
    cout << 1867 + x << endl;
if (c == 'T')
    cout << 1911 + x << endl;
if (c == 'S')
    cout << 1925 + x << endl;
if (c == 'H')
    cout << 1988 + x << endl;
if (c == 'R')
    cout << 2018 + x << endl;
```

2 Solution

Input c x

1 Determine the Era

2 Output the Answer

```
if (c == 'R')
        cout << 1867 + x << endl;
if (c == 'T')
        cout << 1911 + x << endl;
if (c == 'S')
        cout << 1925 + x << endl;
if (c == 'H')
        cout << 1988 + x << endl;
if (c == 'R')
        cout << 2018 + x << endl;
```

Input c x

1 Determine the Era

2 Output the Answer

```
if (c == 'R')
    cout << 1867 + x << endl;
if (c == 'T')
    cout << 1911 + x << endl;
if (c == 'S')
    cout << 1925 + x << endl;
if (c == 'H')
    cout << 1988 + x << endl;
if (c == 'R')
    cout << 2018 + x << endl;
```

Score: $20_{/20}$

Common Mistakes

Mistakes that cost you lots of time

Time Traveler

Say Hi from/to 明年今日

```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
  char c;
  int y;
  cin >> c >> y;
  if(c=='R'){
    cout << 2019+y << endl;
  }
  return 0;
}
```

~~2019~~ 2018

```cpp
#include <bits/stdc++.h>
using namespace std;
int main() {
  char y;
  int n, year;
  cin >> y >> n;
  if(y=='M'){
    year = 1867;
  } else if (y=='T'){
    year = 1911;
  } else if (y=='S'){
    year = 1925;
  } else if (y=='H'){
    year = 1987;
  } else if (y=='R'){
    year = 2018;
  }
  year += n;
  cout << year << endl;
}
```

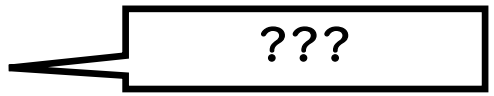
~~1987~~ 1988

All +1 Year

```cpp
#include <iostream>
using namespace std;
int main() {
 int x, a;
char num;
  cin >> x >> num ;
    if (x == 'M' ) { a= 1868 + num;}else if
      (x == 'T' ) { a= 1912 + num;}else if
        (x == 'S') { a= 1926 + num;}else if
          (x == 'H') { a=1989 + num;}else if
            (x== 'R'){ a=2019 + num;}
cout <<a;}
```

手多多 心思疏

```cpp
#include <bits/stdc++.h>
using namespace std;
string a;
int main(){
    string a;
    int b, ans=0;
    cin >> a >> b;
    if (b==0){
        ans == 1868;
    }else if (a=="M"){
       ans = 1868+b-1;
    }else if (a=="T"){
        ans = 1912+b-1;
    }else if (a=="S"){
        ans = 1926+b-1;
    }else if (a=="H"){
        ans = 1988+b-1;
    }else if (a=="R" && b<=5){
        ans = 2019+b-1;
    }
    cout << ans << endl;
}
```



簡單複雜化

自暴其短

```
#include <bits/stdc++.h>
using namespace std;
int main() {
  string S;
    int A, Z, X, W, Y, Q;
  cin >> S >> A >> Z >> X >> W >> Y >> Q;
  int ans = 1868;
  if(S == "M"){
    for(int i=0; i<A; i=i+1 ){
      ans = ans + 1;}
      if(A = 45){
    ans = ans -1;
        cout << ans;
  }
    }
  int B = 1912;
  if(S == "T"){
    for(int i=0; i<Y; i=i+1){
      B = B + 1;
    }
    if(Y = 15){
      Y = Y - 1;
    }
    cout << Y;
  }
  int C = 1926;
  if(S == "S"){
    for(int i=0; i<Z; i=i+1){
      C = C + 1;
    }
    if (Z = 64){
      Z = Z - 1;
    }
    cout << Z;
  }
  int D = 1926;
  if(S == "H"){
    for(int i=0; i<X; i=i+1){
      D = D + 1;
    }
    if(X = 31){
      X = X - 1;
    }
    cout << X;
  }
  int E = 2019;
  if(S == "R"){
    for(int i=0; i<Q; i=i+1){
      E = E + 1;
    }
    if(Q = 5){
      Q = Q - 1;
    }
    cout << Q;
  }
}
```

3 Common Mistakes

上課沒聽書　最尾會執輸

```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
  long long M, T, S, H, R, x, num, rec;
  cin >> x >> num;
  if(x==M){
   rec = (num+1867);
  }else if (x==T){
  rec = (num+1911);
   }else if(x==S){
  rec= (num+1925);
   }else if(x==H){
    rec= (num+1989);
   }else if (x==R){
    rec=(num+2018);
   }
  cout << rec << endl;

```

~~long long~~ char

'M'

```cpp
#include<bits/stdc++.h>
using namespace std;

int main(){
sring"a"
if(m){cout<<"1868-1962"}
else if (t){cout<<"1912 - 1926"}
else if (s){cout<<"1926 - 1989"}
else if (h){cout<<"1989 - 2019"}
else if (r){cout<<"2019 - 2023"}
```

Compilation Error

3 Ways to Improve

- Listen and concentrate in lessons
- Think carefully before coding
- Run locally before submitting

Averagers: Scoregame

TS231

Problem Statement

- There are N classes with M students each
- T exams in a year

- Target: Find the average score of **each exam** for all students → Output T data

Observation

- Do the number of classes and the number of students matter?
- No!

- Therefore, we can just treat the input as **N*M students taking T exams**.

Subtask 2 (T = 1)

- There is only **1 exam**

→ N*M numbers in the input

- Solution: Use 1D array to store all value, then add them up, finally devise the sum by N*M.

- Score: 8/30

Subtask 2 (T = 1)

- Alternative method: No need to use array to store the data.
- For each number in the input:
  - Just add it into the running total
- Output **running total / (N*M)**

Subtask 1 (1 <= N, M, T <= 10)

- The naïve way to solve the problem:

Use a 3D/2D array to story all numbers in the input.

Then for each exam, calculate the sum of data and output sum/(N*M).

- Score: 13/30

Full Solution

- Note that if you try to use subtask 1 approach to solve the full problem, you may get **Accepted** or **Runtime Error** depending on the implementation.

- For a 2D int array of 10000*100 or a 3D int array of 100*100*100:
- Memory usage = 4*100^3 = 4*10^6 bytes

(about **4MB**)

Full Solution

- When put the array inside main() function, it **would not work** as the array is too big.

- When put outside main() function (i.e. as global variable), it **will work**!

Full Solution

- Better way to solve the problem:
- Extend the alternative method of subtask 2.

- **DO NOT** use an array to store all data.

Full Solution

- Use an 1D array to keep track of the running total of each exam
- Then output the **running total / (N*M)** for **EACH exam**
- Score: 30/30

Exemplar 1

```cpp
#include <bits/stdc++.h>
#include <algorithm>
#include <vector>
using namespace std;

int main(){
    long long N,M,T,diu;
    cin >> N >> M >> T;
    diu = N*M*T;
    long long a[diu];
    double avg[diu];
    for (long long i = 1; i <= diu; i++)
    {
        cin >> a[i];
    }
    for (long long i = 1; i <= T; i++)
    {
        for (long long j = i; j <= diu; j=j+T)
        {
            avg[i] = avg[i] + a[j];
        }
        avg[i] = avg[i]/(N*M);
        cout << fixed << setprecision(2) << avg[i] << " ";
    }
}
```

Please name the variables politely!!!

8/30

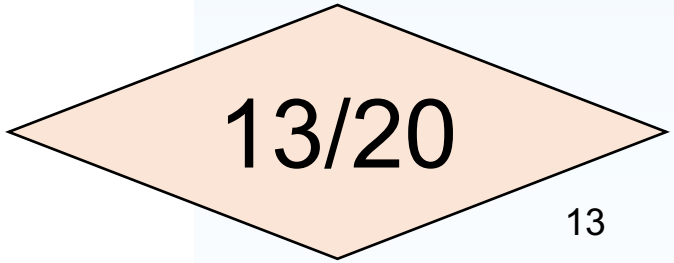
Exemplar 1

- Attempt of a full solution is seen
- But the code is not well-thought and there are many flaws
- 8/30 is given

Exemplar 2

```cpp
#include <iostream>

using namespace std;

int X[1100][1100];

int main()
{
    float N, M, T, Y;
    float A;

    cin >> N >> M >> T;

    for (int i = 0; i < N*M; i++)
    {
        for (int j = 0; j < T; j++)
        {
            cin >> X[i][j];
        }
    }

    for (int i = 0; i < T; i++)
    {
        Y = 0;
        for (int j = 0; j < N*M; j++)
        {
            Y += X[j][i];
        }
        A = Y/(N*M);
        cout << A << " ";
    }
    cout << endl;
    return 0;
}
```

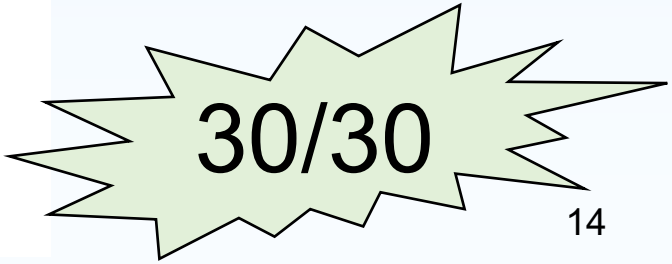
Array not big enough to store all data



Exemplar 3

```cpp
#include <bits/stdc++.h>
using namespace std;

int main(){
  int n, m, t;
  cin >> n >> m >> t;
  double average[t];
  for(int i=0;i<t;i++){
    average[i] = 0;
  }
  for(int i=0;i<n*m;i++){
    for(int j=0;j<t;j++){
      double temp;
      cin >> temp;
      average[j] += temp;
    }
  }
  for(int i=0;i<t;i++){
    average[i] /= n*m;
  }
  for(int i=0;i<t;i++){
    cout << average[i] << ' ';
  }
  cout << endl;
}
```

Clean execution of the full solution



Floating Point Numbers

The values in your answer is considered correct if their absolute or relative error do not exceed 10^{-2}. Formally, let the value in your answer be a, and the model answer be b, your answer is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-2}$. **Please note that it is not necessary to consider the precision error if you use floating point numbers normally.**

- In this problem, the error cannot exceed 10^-2, which is **VERY lenient**.

- You can just use **double** in C++ then output the value without any modification.

Floating Point Numbers

- Generally, the limit of error will be 10^-6
- Therefore, when doing the output, we sometimes have to do some modification:
- E.g. output variable **x**:
- cout << **fixed << setprecision(10)** << x;

(10 is used to further ensure the output to have a small enough error that can pass the test, you can use other numbers you want)

Thank you!

TS232 - Photo Shop

2022/23 Team Selection Test

Problem Statement

Given N positions each with “brightness index” B_i
Space needed for “washing” photo: M

Find the **total brightness index** across the optimal place interval
(which has the least total brightness index)

Subtask 1 (5pts: M = N, B_i<=100)

Just sum up all B_i

That’s it.

Subtask 2 (6pts: $B_i = i$)

As B_i is increasing with i,
the first interval from the left is always the smallest.

Just sum up the first M B_i

That’s it.

Subtask 3 (9pts: M<=N<=5000, B_i<=100)

Check every interval from [1, M] to [N-M+1, N] and get the minimum sum

Time Complexity: O(NM)

Full Solution

First calculate the sum from [1, M], then move the interval by one position every time.

[1, M] → [2, M+1] → … → [N-M+1, N]

For each move to the right, subtract the removed value from the running sum, then add the new value to it.

Time Complexity: O(N)

Subtask 4 (13pts: B_i<=100)

Pity points for missing long long

Hope to not see this kind of subtasks again!

T233 Colour Picker

2022/23 Team Selection Test

Style Inspired by PCOI

Problem Statement

Given N responses (respondent's name + colour picked), find the most popular colour.

(People may change their choice!)

Subtask 1

C_i = `red` or `yellow`

$R_i \neq R_j$ for $i \neq j$

No repeat respondent!

We can use 2 variables to count the number of occurrence of `red` and `yellow`, respectively.

If one of them is higher, output it.

Otherwise, both colours having equal occurrence, output `idk`.

Expected Score: 12 (Cumulative: 12)

Subtask 2

$1 \leq N \leq 1000$

C_i = red or yellow

Respondent may repeat!

We could use 2 for-loops to delete all repeated responses from the same person and only keep the last one.

Then we can use the Subtask 1 solution to find out the answer.

Time complexity: $O(N^2)$

Expected Score: 8 (Cumulative: 20)

Subtask 3

$1 \leq N \leq 1000$

It is guaranteed that there is only one most popular colour

More than 2 colours now!

Similar to Subtask 2, we can delete the repeated responses and keep the last one.

Then use `map<string, int>` to count each colour's occurrence.

Expected Score: 10 (Cumulative: 30)

Subtask 4

$R_i \neq R_j$ for $i \neq j$

No repeat respondent again!

We can start with the Subtask 1 solution, then use `map<string, int>` to count each colour's occurrence.

Done? No!

Remember to handle the case when multiple colours have the same highest number of occurrence!

Expected Score: 23 (Cumulative: 41)

Full Solution

The solution is separated in 2 parts:

1. Use `map<string, string>` to store the newest response of each respondent
2. Implement Subtask 4 solution

Expected Score: 45 (Cumulative: 45)

Comment

This problem is straightforward for those who are familiar with `map`.

In fact, the full solution is simpler than the Subtask 2/3 solution.

It demonstrates the importance of utilising the built-in structure in C++.

Story Copier

TS234

Problem Statement

You task is to find the number of the substring of string S which include a substring equal to the “Magical Phrase”.

Let x be a substring of S, find the number of x such that one of the substrings of x equal to the “Magical Phrase”.

For example, $S =$ abc, “Magical Phrase” $=$ b:

4 substrings (ab, abc, b, bc) include b.

Subtask 1

$1 \leq N \leq 10$

Easy Subtask.

Just check all substrings of S.

How? Loop all the substrings of the substrings of S with length K.

How? Use `substr()` function.

How? Ask yourself.

.substr()

`.substr(index, length)`

is a function of string which return a substring by **start** *index* and *length*.

Example:

$S =$ `123456`

`S.substr(2, 3)` → `345`

`S.substr(5, 1)` → `6`

Remember that *index* starts from `0`.

```cpp
#include <bits/stdc++.h>
using namespace std;

int main(){
  int N, K, ans = 0;
  bool valid;
  string S, T, x;
  cin >> N >> K >> S >> T;
  for (int i=K; i<=N; i++){
    for (int j=0; j<=N-i; j++){
      x = S.substr(j,i);
      valid = false;
      for (int l=0; l<=i-K; l++){
        if (x.substr(l, K) == T){
          valid = true;
        }
      }
      if (valid){
        ans++;
      }
    }
  }
  cout << ans;
}
```

Subtask 1 Solution

Time complexity: $O(N^3)$

Expected score: 13

Subtask 2

S and T consists of a only.

Easy Subtask.

We can **easily** observe that for every substring of S with length more than K, it includes T.

Therefore, it is a simple math question: $1 + 2 + 3 + \cdots + (N - K + 1)$.

How?

Triangular number: $\frac{N*(N+1)}{2}$

Time complexity: $O(1)$

Expected score: 9 (Cumulative: 22)

Subtask 2

How?

Here it is, may I introduce the… Triangular number!

$$1 + 2 + 3 + \cdots + x = \frac{x(x+1)}{2}$$

$$1 + 2 + 3 + \cdots + (N - K + 1) = \frac{(N - K + 1)(N - K + 2)}{2}$$

Time complexity: $O(1)$

Expected score: 9 (Cumulative: 22)

Subtask 4

$1 \leq N \leq 5000$

T has exactly one occurrence in S.

We can observe that if the left boundary is less than or equal to the start index of T in S and the right boundary is greater than or equal to the end index of T in S, this is a substring we are counting.

Subtask 4

We can seek out that we can choose a pair of left boundary and right boundary to form a substring.

Therefore, we only need to find the position of T in S by checking all the substrings with length K in S.

Time complexity: $O(N^2)$

Expected score: 11 (Cumulative: 33)

Subtask 5

$1 \leq N \leq 5000$

Using the idea of subtask 4, we can calculate the answer when there is one occurrence of T in S. But if there is more one occurrence of T in S, we will calculate some overlapping substrings.

By the definition of different substrings, if the left boundaries are different, the substrings are different. Therefore, you only need to prevent the overlap of left boundaries in order to solve subtask 5.

Time complexity: $O(N^2)$

Expected score: 14 (Cumulative: 47)

Subtask 3

$K = 1$

Easy subtask.

Using the same idea of subtask 5 only!!!

Because $K = 1$, then we can find the position of all T in S using $O(N)$.

Time complexity: $O(N)$

Expected score: 17 (Cumulative: 64)

Full Solution

Impossible subtask!!!!!!!!!!!!!!!!!!!!

Ask Daniel!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Time complexity: $O(N + K)$

Expected score: 65