

競賽概論

課前廢話

- Author: 鄭宇翔/藍白/中央資工碩一
- 富皓的狗
- ICPC regional 1 honorable mention, 4 bronze medalist.
- 打了五年（實際上因為疫情可以打第六年），目前是退役狀態
- 感謝 LeeShoWhaodian 的講義參考，但我會刪掉一些非新手向的部分
- 重要的紅色，正常的敘述黑色，額外補充、對當前不太重要的綠色

語法介紹

LeeShoWhaodian 的語法目錄

- 快讀/快出/換行 "\n"[i == n]
- GCC pragma
- memset(arr,value,sizeof(arr))-1,0,0x3f3f3f3f
- for(auto i : arr)
- lambda sort(begin, end, [](const int lhs, const int rhs){return lhs < rhs});
- define int ll
- __int128
- begin, rbegin, end, rend
- prev, next

快讀/快出

- `ios::sync_with_stdio(0); cin.tie(NULL); cout.tie(NULL);` + 使用 “\n” 代替 `endl`;
- 注意 `ios` 後 `cin`, `scanf`, `cout`, `printf` 不能混用，詳細資訊請看[這邊](#)

```
2  getchar() -> getchar_unlocked()
3  fread() -> fread_unlocked()
4  -----
5  inline char readchar() {
6      const int S = 1<<20; // buffer size
7      static char buf[S], *p = buf, *q = buf;
8      if(p == q && (q = (p=buf)+fread(buf,1,S,stdin)) == buf) return EOF;
9      return *p++;
10 }
11
12 ✓ inline int nxtint() {
13     // if readchar can't use, change readchar() to getchar()
14     int x = 0, neg = 0, c = readchar();
15     if (c == EOF) return -1;
16     while (('0' > c || c > '9') && c != '-' && c != EOF) c = readchar();
17     if (c == '-') neg = true, c = readchar();
18     while ('0' <= c && c <= '9') x = x * 10 + (c ^ '0'), c = readchar();
19     return (neg? x: -x);
20 }
```

```
void write(int x){
    if(x<0)
        putchar('-'),x=-x;
    if(x>9)
        write(x/10);
    putchar(x%10+'0');
    return;
}
```

換行小技巧

- 不用寫 if/else 來判斷要不要換行

0	1
空白	換行

```
for(int i=1;i<=5;i++){  
    cout<<i<<" \n"[i==5];  
}
```

GCC pragma

- 一些奇怪的優化，拉長編譯時間換取執行時間
- 不大好用，但在極端情況下確實有隊伍撈到 AC
- 但被 NCPC 主辦方無事前說明的情況下手動 WA 了，直接吃 penalty
- NCPC / ICPC 寫不到六題，非常不建議用這東西

```
1 #pragma GCC optimize("Ofast")
2 #pragma GCC target("sse3","sse2","sse")
3 #pragma GCC target("avx","sse4","sse4.1","sse4.2","ssse3")
4 #pragma GCC target("f16c")
5 #pragma GCC optimize("inline","fast-math","unroll-loops","no-stack-protector")
6 #pragma GCC diagnostic error "-fwhole-program"
7 #pragma GCC diagnostic error "-fcse-skip-blocks"
8 #pragma GCC diagnostic error "-funsafe-loop-optimizations"
9 #pragma GCC diagnostic error "-std=c++14"
```

memset()

- 可以將整個 arr 的所有位元組都填上同一個 value ($0 \leq value < 256$)
- memset(arr, 0, sizeof(arr));
- 0 代表歸零，1 代表 -1，3f 代表一個大數 (略小於 2^{30})

memset

memset

```
void * memset ( void * ptr, int value, size_t num );
```

Fill block of memory

Sets the first *num* bytes of the block of memory pointed by *ptr* to the specified *value* (interpreted as an unsigned char).

Parameters

ptr

Pointer to the block of memory to fill.

value

Value to be set. The value is passed as an int, but the function fills the block of memory using the *unsigned char* conversion of this *value*.

num

Number of bytes to be set to the *value*.

[size_t](#) is an unsigned integral type.

Range-based for loop

- `for(auto i: arr)`
- `auto variable` 是 c++11 引進的新變數型態，可以自動判斷變數型態

```
4 int main(){  
5     vector<int> v(100, 0);  
6     for(int i = 0; i < v.size(); i++){  
7         v[i] += 2;  
8     }  
9     // NOTE: num won't be changed!  
10    for(auto num: v) num += 2;  
11    // use referencing operator to modify with range-based for loop  
12    for(auto &num: v) num += 2;  
13 }
```

Range-based for loop

- 配上 container iterator 效果更加
- 現在比賽都有 C++17 可以用，所以語法上不用太擔心

```
4 int main(){
5     map<int, int> m;
6     for(int i = 0; i < 100; i++) m[i] = 2 * i;
7     for(map<int, int>:: iterator it = m.begin(); it != m.end(); it++){
8         cout << it->first << ' ' << it->second << endl;
9     }
10    for(auto it = m.begin(); it != m.end(); it++);
11    // note it's "it.first" not "it->first" !
12    for(auto it: m) cout << it.first << ' ' << it.second << endl;
13    // c++ 17 or higher
14    for(auto [k, v]: m) cout << k << ' ' << v << endl;
15 }
```

#define int long long

- 當你發現運算過程會爆 int 想要一勞永逸的做法
- Define 完要記得將 int main 改成 signed main / int32_t main

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 #define int long long
4
5 signed main(){
6     map<int, int> m;
7     for(int i = 0; i < 100;
```

__int128, __float128

- 顧名思義，擁有 128 bits 的 signed int 跟 float
- unsigned __int128 就是 unsigned 128-bits int
- C/C++ standard I/O 不認識 __int128，所以要重新寫 read/write function
- 搭配著數學題一起用
- 有興趣的請 Google 或[參考這篇](#)

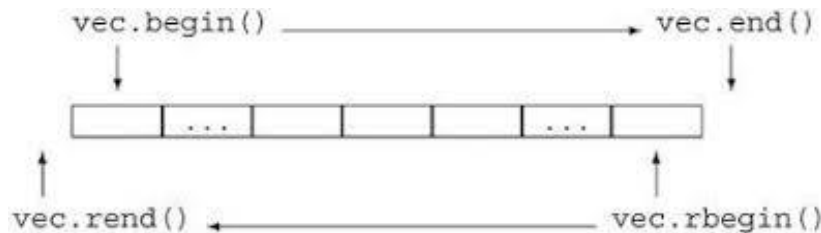
__int128, __float128

- 顧名思義，擁有 128 bits 的 signed int 跟 float
- unsigned __int128 就是 unsigned 128-bits int
- C/C++ standard I/O 不認識 __int128，所以要重新寫 read/write function
- 搭配著數學題一起用
- 有興趣的請 Google 或[參考這篇](#)，去看看 read/write 的寫法

begin, end, rbegin, rend

- vector, map, ... 之類的 STL container 才會有
- end() 會是 container 的最後一個元素的下一個位置
- rend() 會是第一個元素的前面一個位置
- [詳細介紹](#)

```
5 signed main(){  
6     vector<int> v;  
7     // sort the vector increasingly.  
8     sort(v.begin(), v.end());  
9     // sort the vector decreasingly.  
10    sort(v.begin(), v.end(), greater<int>());  
11    sort(v.rbegin(), v.rend());  
12 }
```



prev, next

- prev是指用來獲取一個距離指定迭代器左側 n 個元素的迭代器。
- next是指用來獲取一個距離指定迭代器右側 n 個元素的迭代器。
- 例如：vec[10] = {0, 1, 2, 3, 4, 5}, it 指向第 2 個位置 (0-based)
- prev(it, 2) = 0, prev(it, -2) = 4
- next(it, 2) = 4, next(it, -2) = 0

賽制介紹

比賽簡介

- NCPC 初賽
 - 3 hours，經驗來看六題左右
 - 會找各校的人幫忙，辦在各校裡面，除台清交比賽自己辦之外，剩下學校同一時間用這些題目取前 6 隊進 NCPC 決賽
- NCPC 決賽
 - 5 hours，10 ~ 15 題
 - 北部學校在師大圖書館校區的教育學院大樓，南部學校在中山大學的圖書資訊大樓 B1

比賽簡介

- TOPC

- 3 hours，至少 7 題（往年都是 10 題左右）
- 有 $\text{ceil}(\text{題數}/3)$ 題可以用 python3 寫
- 自我道德約束，地點不限

- ICPC

- 第一天報到放 codebook、文具、水壺，第二天寄放背包之後空手進會場
- 5 hours，8~13 題
- 臺灣最大的程式設計比賽
- 17 到 23 年的比賽地點分別是：東華大學、臺灣大學、臺北商業大學 * 3、桃園武漢國中、桃園綜合體育館

比賽簡介







- 三人一臺電腦
- 都使用 Dom Judge (2018 年的比賽是 PC²)
- NCPC 初賽能不能印 code，環境...看學校
- TOPC 可以自己用印表機印 code
- NCPC 決賽 / ICPC regional 官方會幫你印 code via DOMjudge
- NCPC 決賽 / ICPC regional 答對一題會發一顆氣球，scoreboard frozen 之後 AC 不會發氣球
- NCPC 決賽 / ICPC regional OS 均為 ubuntu
- 語言都是 C/C++, Java, Python, kotlin

ICPC 晉級規定

- 參加 TOPC 且解出一題的隊伍中，錄取每校最優一隊，此類限 30 隊。
- 私立比賽、科大比賽各前 10 名
- CPE 錄取 10 隊，但每校至多一隊，且需參加 TOPC 並解出至少一題。
- 以上五類隊伍中，出現重複隊伍或因故放棄參賽者，由同校參與 TOPC 之隊伍依成績擇優遞補，直到該校隊伍用罄為止。
- 如前六類隊伍總數未達100隊時，由主辦單位邀請 TOPC 績優隊伍參賽，受邀隊伍之隊伍排名須為前 50 %。

Ranking & Time penalty?

- 比 AC 題數 -> 比 penalty -> 比 submission ID 早晚
- penalty 算法是 AC 所花的時間分鐘數總和 + penalty 分鐘數
- CE 外的 incorrect (WA, TLE, MLE, RE, no-output) 一次, 該題的 Time penalty + 20 min
- 沒有 AC 的題目不算在 penalty 裡面

48		♡	CurryBaofan Fu Jen Catholic University	5	443	01 1 try	15 tries	40 1 try	207 1 try		01 2 tries		10 1 try
49		♡	oyasumi_suya National Taiwan Ocean University	5	457	60 1 try	1 try	35 1 try	243 2 tries		74 1 try		25 1 try
50		♡	LLL National Taipei University	5	457	55 1 try	5 tries	23 1 try	289 3 tries		37 1 try	7 tries	13 1 try
51		♡	NTNU_fighting National Taiwan Normal University	5	474	55 2 tries	237 3 tries	38 1 try	2 tries		72 1 try		12 1 try
52		♡	NTPU_Kite National Taipei University	5	501	57 2 tries	3 tries	19 1 try	190 1 try		161 2 tries	3 tries	34 1 try
			NTNU TSAIBIA			42		50	243		140		26

賽前準備

- 看好各比賽的官方通知
- 事先準備合規的 codebook (不要訂裝，可能會被拆) 去比賽會場，ICPC day1 codebook 就要帶齊
- ICPC regional rule 2023:

隊伍可以攜帶下列物品進賽場：

1. 一本紙本自然語言辭典 (不允許電子辭典)
2. 一本合乎下列規定的隊伍筆記，至多複印三份
 - 至多 25 頁 A4 紙或 letter 紙
 - 單面列印或書寫 (背面必須空白)
 - 頁碼清楚標示於右上角
 - 矯正後視力在距離五十公分時可以在不使用放大鏡與其他濾鏡時閱讀。
 - 請於練習賽前準備好隊伍筆記帶入賽場，賽場不提供列印。
3. 文具如筆、鉛筆、橡皮擦、直尺。
4. 無電子功能、不可食用且不干擾其他隊伍與工作人員之吉祥物或是幸運物

隊伍不得攜帶以下物品：

1. 任何機器可讀的資料。
2. 任何通訊裝置，如行動電話、智慧手錶等等。
3. 任何電子裝置，含電子錶、鍵盤。
4. 額外的空白紙、方格紙。
5. 食物與飲料。

一些細節

- TOPC 使用 lazy evaluation，遇到什麼問題就輸出什麼 (RE -> MLE/TLE -> WA -> AC)
- 聽清楚裁判的指示，比賽都可以問 judge 主機的編譯參數 (via DOMjudge)
- 開題 (default code, setting) / 讀題工作分配
- 看 scoreboard 或氣球跟題 / 倒數一小時時間評估
- 記得去吃點心 (NCPC 唯一值得稱讚的只有點心)
- 剩下的備賽事項