

【滿分最高 120 分】【四、五、六三題最多計 22 分】

(隨身碟、手機、書包放教室前後，編譯成功可執行，並瞭解自己程式意義，助教詢問程式意義，依據題目定義與通過測資類別給分，每 1 類測試資料(明/暗測)都通過，得該題分數。每完成一題，務必舉手評分一次，程式不得使用全域變數

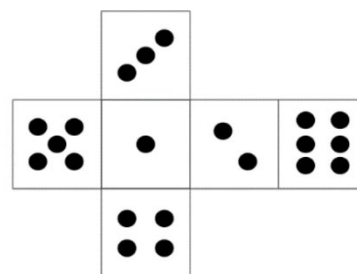
一、列印圖形(22%)(6%, 6%, 5%, 5%)

依照輸入圖案編號與參數，輸出對應圖形。每種圖形皆需使用 function 完成，每個 function 至多使用一層迴圈，可使用 function 呼叫 function。

輸入說明	輸出說明
1. 第 1 行代表圖形編號。 2. 第 2 行代表圖形的參數。	1. 輸出相對應的圖形

Sample Input 1: 圖形 1	Sample Output:
1 5	1 12 123 1234 12345 54321 4321 321 21 1
Sample Input 2: 圖形 2	Sample Output:
2 5	__1__ __212__ __32123__ __4321234__ 543212345 __4321234__ __32123__ __212__ __1__
Sample Input 3: 圖形 3	Sample Output:
3 4	1____ 121____ 12321__ 1234321
Sample Input 4: 圖形 4	Sample Output:
4 5	12345__ _54321__ __12345__ __54321__ __12345

二、骰子遊戲(24%) (4%, 4%, 4%, 4%, 4%, 4%)



A, B 兩人各用四顆骰子計點比賽，一開始都是點數 1 朝上，點數 4 朝前，點數 2 朝右(骰子展開如上圖所示)。接下來 M 次修改操作，每次操作包含四個整數 x, w, y, z 代表四個骰子操作。

x, w, y, z 意義為：

- 1 表示向前滾一次。(滾完之後，點數 3 朝上)
- 2 表示向後滾一次。(滾完之後，點數 4 朝上)
- 3 表示向右滾一次。(滾完之後，點數 5 朝上)
- 4 表示向左滾一次。(滾完之後，點數 6 朝上)

請依計分方式計算 A, B 兩人得分，比較 A, B 兩人分數，得分數高為勝者。

計分方式：

1. 若二顆點數均相同，只算另外 2 顆點數的總和，例如 6, 6, 3, 2 只得 5 分，6, 6, 3, 3 或 6, 6, 6, 2 或 6, 6, 6, 6 計為 0 分。若有兩組二顆點數相同，取點數大的計分。
2. 其餘狀況皆為 0 分

輸入說明	輸出說明
1. 第 1 行為 A 四顆骰子修正次數 M_A ， $1 \leq M_A \leq 5$ 2. 接下來 M_A 行，為 x, w, y, z 修正 3. 接著為 B 四顆骰子修正次數 M_B ， $1 \leq M_B \leq 5$ 4. 接下來 M_B 行，為 x, w, y, z 修正	遊戲勝負結果 (A win, B win, Tie)

Sample input 1:	Sample output:
4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 4 4 1 3 2 4 1 3 2 4 1 3 2 4 1 3 2	Tie
Sample input 2:	Sample output:
2 1 2 3 4 2 1 4 3 2 4 3 2 1 3 4 1 2	Tie
Sample input 3:	Sample output:
2 1 2 3 4	Tie

4 3 2 1 3 1 2 3 4 2 3 1 1 4 4 4 4	
Sample input 4:	Sample output:
4 1 1 2 2 4 4 3 3 3 3 1 1 2 2 1 1 3 1 2 3 4 1 2 3 4 3 1 4 2	Tie
Sample input 5:	Sample output:
2 1 2 3 4 2 3 1 1 5 1 2 3 4 4 2 3 1 3 3 3 3 1 1 2 4 3 2 1 4	B win
Sample input 6:	Sample output:
4 2 2 2 1 2 3 3 3 4 4 2 2 4 1 2 3 3 1 2 3 4 4 2 3 1 2 2 3 3	A win

三、棒球題目 (22%) (5%, 5%, 6%, 6%)

假設球員打擊情況：

- (1) 安打：以 1, 2, 3 和 H 代表一、二、三和全(四)壘打。
- (2) 出局：以 0 表示 (OUT)。

規則如下：

- (1) 球場上有四個壘包，稱為本壘、一、二和三壘。本壘握球棒稱「擊球員」，在另外三個壘包稱「跑壘員」。
- (2) 當擊球員打擊「安打」時，擊球員與跑壘員可移動；「出局」時，跑壘員不動，擊球員離場換下一位。
- (3) 比賽開始由第 1 位打擊，接著 2, 3, ..., 8, 9 位球員。
- (4) 打出 K 壘打時，場上球員(擊球員和跑壘員)會前進 K 個壘包。本壘到一壘，接著二、三壘，最後回到本壘。回到本壘可得 1 分。
- (5) 每達到三個出局數時，壘包清空(跑壘員都得離開)，重新開始。

輸入說明	輸出說明
1.第 1~9 行為棒次 1~9 球員的打擊 次數整數 a(1<=a< =5)和每次的打擊 結果。 2.第 10 行為累計 出局數整 b(1<=b< =27)	1.第 1 行輸出到第 b 個出局數 時的總得分 2.第 2~4 行依所有球員的跑壘 數由小到大，若相同則以棒次 先後排序並輸出前三位打者的 資訊(第幾棒, 安打數, 跑壘數 量)。

	※ 跑壘數量: 球員在比賽中 總共跑的壘包數量，包含 自己安打跑的壘包數與隊 友安打而跑的壘包數。
--	--

Sample input 1: 求非最後一個出局數	Sample output:
4 0 1 2 1 3 1 2 2 3 0 1 2 3 1 1 1 3 0 1 H 3 1 0 0 3 0 1 0 3 1 2 1 3 1 1 1 5	5 7 0 0 5 1 1 3 1 3
Sample input 2: 求最後一個出局數	Sample output:
4 0 1 2 0 3 0 0 1 3 2 1 1 3 1 0 2 3 1 1 0 3 2 1 2 3 0 1 2 3 1 2 1 3 1 1 1 7	15 2 1 4 4 2 6 5 2 7
Sample input 3: 有跑壘數相同	Sample output:
4 0 0 2 0 4 0 1 0 0 4 3 1 3 2 4 2 1 2 0 3 1 1 0 3 H 2 1 3 1 3 0 3 2 2 2 3 2 0 H 8	18 1 1 4 2 1 4 9 2 6
Sample input 4: 連續安打	Sample output:
5 1 1 2 0 2 5 2 1 0 2 2 5 2 1 1 1 1 5 H 1 2 3 0 4 1 0 2 0 4 2 2 1 1 4 2 1 0 0 4 1 0 2 0 4 2 0 0 1 8	19 9 1 4 7 2 5 8 2 6

四、DF-expression(22%)(3%, 3%, 4%, 4%, 4%, 4%)

DF-expression 是儲存圖像資訊的表達方式，
圖像為一個 NxN 方陣中，
若方陣中的方格全部皆為白色則記為 0，
若方陣中的方格全部皆為黑色則記為 1，
若方陣中的方格不全為白或不全為黑，則記為 2，並將此方陣分割為更小的 4 塊全等方陣(左上、右上、左下、右下)，再依序(左上→右上→左下→右下)記錄四塊方陣資訊。

方格座標定位法：
圖像方陣中最左上方的方格定為 0,0，
由此方格往右的方格座標依序為 0,1、0,2、0,3 以此類推，
由此方格往下的方格座標依序為 1,0、2,0、3,0 以此類推，

請根據題目給予的 DF-expression 表達式判斷出圖像方陣中的白色方格的座標位置，並且將白色方格的座標位置依照由左而右，由上到下進行排序後依序輸出。

輸入說明	輸出說明
1.第 1 行輸入一個 DF-expression 的表達式 S (1 <= strlen(S) <= 100)， 2.第 2 行輸入該張圖像的寬度 N(必為 2 的非負整數次方，2<=N<= 8)	1.白色方格的座標位置依照由左而右，由上到下進行排序後依序輸出，每行 x 值和 y 值間以逗號間隔。 2.若全都為黑色方塊，輸出"all black"

Sample input 1:	Sample output:
1 4	all black
Sample input 2:	Sample output:
21101 4	2,0 2,1 3,0 3,1
Sample input 3:	Sample output:
2210111021001 4	0,1 2,0 2,1 2,3 3,0 3,1 3,2
Sample input 4:	Sample output:
2120101211010 4	0,2 1,2 2,2 2,3 3,0 3,2 3,3
Sample input 5:	Sample output:
221011211012100121 100 4	0,1 1,2 2,1 3,0 3,2

	3,3
Sample input 6:	Sample output:
2221001010011 8	0,1 0,2 0,3 0,4 0,5 0,6 0,7 1,0 1,2 1,3 1,4 1,5 1,6 1,7 2,2 2,3 2,4 2,5 2,6 2,7 3,2 3,3 3,4 3,5 3,6 3,7

五、分數四則運算(22%)(3%, 3%, 3%, 3%, 3%, 3%, 4%)

計算兩個分數的四則運算 +, -, *, /，輸出兩個分數的相加、相減、相乘及相除結果。
請實作以下函式進行計算：

```
void add(int *numer1, int *denom1, int *numer2, int *denom2, int *resNumer, int *resDenom)
void sub(int *numer1, int *denom1, int *numer2, int *denom2, int *resNumer, int *resDenom)
void mul(int *numer1, int *denom1, int *numer2, int *denom2, int *resNumer, int *resDenom)
void div(int *numer1, int *denom1, int *numer2, int *denom2, int *resNumer, int *resDenom)
```

輸入說明	輸出說明
1.第 1, 2 行各代表一個分數。	1.輸出 4 行，分別為兩分數相加、相減、相乘及相除結果。 2.若輸入分數的分母為 0，則輸出 error。 3.結果分數須化為最簡分數。 4.若結果大於 1，化為整數與真分數，真分數部分要加括號，如 (7/6)輸出 1(1/6)。 5.負數結果的輸出，負號在最前面。 6.若結果為 0，輸出 0。

Sample Input 1 : 分母為零
3/5 17/0
Sample output:

error
Sample Input 2 : 同號，帶分數
8(3/4)
17(14/45)
Sample output:
26(11/180)
-8(101/180)
151(17/36)
1575/3116
Sample Input 3 : 同號，真分數
57/391
8721/10000
Sample output:
1(69911/3910000)
-2839911/3910000
29241/230000
10000/59823
Sample Input 4 : 不同號，帶分數
6(21/32)
-11(33/140)
Sample output:
-4(649/1120)
17(999/1120)
-74(3529/4480)
-7455/12584
Sample Input 5 : 不同號，真分數
-319/2310
182/239
Sample output:
31289/50190
-45151/50190
-377/3585
-6931/38220
Sample Input 6 : 同號，真分數與帶分數
-77/89
-7(121/545)
Sample output:
-8(4229/48505)
6(17309/48505)
6(12042/48505)
41965/350304
Sample Input 7 : 不同號，真分數與帶分數
-1212/1225
3(13/48)
Sample output:
2(16549/58800)
-4(15301/58800)
-3(1157/4900)
-58176/192325

六、有限狀態機(22%)(3%, 3%, 3%, 3%, 3%, 3%, 4%)

請使用有限狀態機判斷輸入值為以下何種型態：

- 變數規則：
 - 必須由 "大(小)寫英文字母" 或 "數字" 或 "_" 組成。
 - 必須以 "大(小)寫英文字母" 或是 "_" 開頭。
- 整數規則：
 - 整數除了 "負號" 必須所有字元皆為數字，且 "負號" 只能位於開頭
 - 開頭不可為 0，但整數為 0 的狀況例外。
 - 範圍在 -2147483648 至 2147483647 之間
- 浮點數規則：
 - 浮點數除了小數點與負號外，所有字元皆為數字。

b. 開頭不可為小數點，結尾也不可為小數點，小數點最多只有一個。

c. 開頭若為 0，最多一個。

4. 最簡真分數規則：

a. 分子分母以斜線符號 "/" 隔開，斜線前為分子，斜線後為分母分子必須小於分母。

b. 分子分母皆為數字，數字不以 0 開頭，分子可為負數，分母不可為負數，分母不得為 0。

c. 分子和分母必須互質

5. 長整數規則：

a. 整數必須所有字元皆為數字，且開頭不可為 0

b. 範圍為 >2147483647 或 <-2147483648

6. 十六進位整數規則：

a. 必須以 "0x" 或是 "0X" 開頭

b. 除了開頭其餘部分必須由 "數字" 或 "a~f" 或 "A~F" 組成

輸入說明	輸出說明
1. 第 1 行代表字串輸入次數 N。	1. 輸出字串的對應型態
2. 其後 N 行，輸入字串 S。	2. 若字串為變數，輸出 variable
	3. 若字串為整數，輸出 integer
	4. 若字串為浮點數，輸出 float
	5. 若字串為最簡真分數，輸出 proper fraction
	6. 若字串為長整數，輸出 long int
	7. 若字串為十六進位整數，輸出 hexadecimal int
	8. 若以上情況皆不符合，輸出 error

Sample Input 1: 變數	Sample Output:
4 NTUT1 _csie _1_1_2 _cat1_1_a_	variable variable variable variable
Sample Input 2: 整數	Sample Output:
7 2147483647 2147483640 111112233 0 -654897 -214748364 -2147483648	integer integer integer integer integer integer integer
Sample Input 3: 浮點數	Sample Output:
4 89545.54215 -6456.1435 0.72054 -0.747845 646484.0	float float float float float
Sample Input 4: 最簡真分數	Sample Output:
4 -1/2 3169/3539	proper fraction proper fraction proper fraction

-8/17 2/3	proper fraction
Sample Input 5: 長整數	Sample Output:
5 -2147483649 -7898854254212 10000000002 2147483650 2147483648	long int long int long int long int long int
Sample Input 6: 十六進位整數	Sample Output:
5 0x123456 0Xabcdf 0xAFDE 0X1a2c4f 0xC2aE41D	hexadecimal int hexadecimal int hexadecimal int hexadecimal int hexadecimal int
Sample Input 7: 錯誤	Sample Output:
8 .____. -0. 0xxx1A23 11/33 4566.00 *_acxg 540/457	error error error error error error error

-36444599854869275678931156617041753522072011927.29 638644544436605604050232
Sample Input 3:
-53156313351.56356131651518489999 6546556465.5631561560000001
Sample Output:
-46609756886.00040516051518479999 -59702869817.12671747251518499999 -347990806857179555289.2544120740102389356137444899 58489999
Sample Input 4:
-51551315156151651656151561.56416466156156416111110 0001 -456465465465487941051641.5415641566556487489789797 1
Sample Output:
-52007780621617139597203203.10572881821721291009007 9711 -51094849690686163715099920.02260050490591541213212 0291 23531395068110826834284557857357124426712808936571. 66675538568731842068294886919511539014938670475997 971
Sample Input 5:
51698465156316131561.9996513153131311 -51698465156316131561.9996513153131311
Sample Output:
0 103396930312632263123.9993026306262622 -2672731299518833137421601593582223294188.750492629 08856725686284408578721
Sample Input 6:
-56465153165116464165465.5616168498411111011 56465153165116464165465.5616168498411111011
Sample Output:
0 -112930306330232928330931.1232336996822222022 -3188313521960061851107507244358180236003616467.569 82096194294015722487275219365442121

✓七、長浮點數大數運算(30%)(5%, 5%, 5%, 5%, 5%, 5%)
將長浮點數分成整數與小數兩個部分儲存於兩個一維陣列，計算兩個長浮點數的相加、相減、相乘，輸出的兩個部分長度最大精確各為 60 位。

輸入說明	輸出說明
1.第 1 行代表 1 個長浮點數。 2.第 2 行代表 2 個長浮點數。	1.輸出兩長浮點數計算加法結果。 2.輸出兩長浮點數計算減法結果。 3.輸出兩長浮點數計算乘法結果。 4.負數結果的輸出，負號在最前面。 5.若結果為 0，輸出 0。

Sample Input 1:
561561151564564987.56131231845646564645146846153644 5648941 6541655656464565.554644646999999945611
Sample Output:
568102807221029553.11595696545646559206246846153644 5648941 555019495908100422.00666767145646570084046846153644 5648941 3673539683583091767903507186701247.4348566622358720 99446851000885001205407648714321257599747951
Sample Input 2:
6455345646415646531879564161.5313 -5645646546456465651.4565646165489798465464
Sample Output:
6455345640769999985423098510.0747353834510201534536 6455345652061293078336029812.9878646165489798465464