```
/*** cowbull.c   -- 電腦猜人想的 Bull & Cow game
 *          @CopyLeft by tsaiwn@csie.nctu.edu.tw
**************************************/
       int possible[5040];    // global variable
       int npsible;    // global variable 記著有幾個可能答案 //

int main(){
       int n, tmp, nguess, nbull, ncow;
       int yn, myGuess;
       long now= time(0);
       srand( now &0xffff);    /* randomize, see "man srand" */
   boolean quit=false;
   while( !quit ) {   //   again:
       npsible = prepare_all_possible_answer();   // 放在 possible[5040]
       nbull = ncow = nguess = 0;
       while(nbull != 4) {
          // 由所有可能答案中隨便挑一個當 myGuess, 並印出說猜該數 //
           n = rand() % npsible;   /* 0.. npsible -1 */
           // 接著把第 n 個 與 第 0 個對調
           tmp = possible[0]; possible[0] = possible[n]; possible[n] = tmp;
           myGuess = possible[0];   /* 永遠猜第 0 個比較好寫 */
           // 印出 myGuess 爲我們猜的 ..
           ++nguess;  // 猜的次數
          // 由 user 取得幾A放 nbull, 幾 B 放 ncow
           if(nbull == 4){
               if(nguess<5)fprintf(stderr, "才 %d 次", nguess);
               fprintf(stderr, "喔 Yeah 我猜到了!\n");
               if(nguess>6)fprintf(stderr, "這麼衰..猜了 %d 次!\n", nguess);
               break;
           }
           /* 還沒猜對, 把不可能的答案刪除並修正 npsible */
           npsible = reduceAnswers(myGuess, nbull, ncow);
           if(npsible <= 0){
               fprintf(stderr, "You 欺騙我!\n");
               break;
           }
       }
       fprintf(stderr, "\n要不要繼續玩下一攤(yes, no)? ");
       yn = askPlayAgain();
       if(yn==0) quit = true; //if(yn) goto again; // 注意 Java 沒有 goto
   } // while( !quit
       fprintf(stderr, "Thank you and Bye!\n");
}
int askPlayAgain(){    /* return 1 if he/she said YES */
    //問要不要續攤? // 要則回傳 1, 否則回傳 0
}
int prepare_all_possible_answer(){
      //想辦法 (用四層 for loop ) 把所有可能答案 放在 possible[5040]
      //共有 5040 個, 最小 0123, 最大 9876
      //(因每位不能重複)
      //  把所有的答案放在  possible[0]
      // int p;  // p 記住共有幾個可能答案
    return p;
}
int reduceAnswers(int myGuess, int nbull, int ncow){
       // 刪除不可能之答案, 並將可能之答案放在 array 前端 //
       // 以下只做刪除不可能之答案 //
       /// 順便把可能答案往 array 前端移 : 留給大家想 ..
       int i, n= 0;  /* possible[0] 是第一個可以蓋掉的, 因它是錯的答案 */
             /* 所以由 possible[1] 開始看看有哪些是有可能的就留下來 */
       for(i=1; i< npsible; i++){
             // 叫小弟(function)去比較 myGuess 和 possible[i]
             //以便知道myGuess對於 possible[i]是幾A幾B //
             //若幾A幾B剛好與 nbull 和 ncow 完全相同表示爲可能答案 //
             //否則都要消去 //
       }
       return n;  // 剩下 n 個可能答案
}
```