http://www.csie.nctu.edu.tw/~tsaiwn/net/02_examples/GAMES/batnum/

BATNUM -- BATtle of NUMbers 數字大戰

```
 int nStone, maxTake;  // 總石頭數, 最多可拿幾個?
   boolean winLast = true; // 拿最後一個贏還是輸?
                           // 注意 C 不認識  true 與 boolean
   //BATtle of NUMbers
   hello( ); // welcome message
   playAgain = true;
   while(playAgain) {
      prepareGame( );
      userFirst = askUserFirst( );
      playGame( );
      playAgain = askPlayAgain( );
   }
   print("Bye bye\n");

=====================
void hello( ) {
   // welcome message and/or game rules
}
void prepareGame( ) {
    // 用亂數 生出 遊戲所需的各個 Global variables
    //nStone = 用亂數取得 15 .. 31;
    //maxTake = 用亂數取得 3..7 ;
    //用亂數決定 winLast 是 true 還是 false

}
boolean askUserFirst( ) {
   // 若 USER 要先拿, 就傳回 1
   return (38==38);  // YES / TRUE
}
boolean askPlayAgain( ) {
   // 問user 並讀入答案, 若 USER 要繼續玩, 就傳回 true
   return false;  // NO / FALSE 表示不完了 :-)
}
void userTurn( ) {
   // nTake = 問 USER 要拿幾個;
   // 檢查 nTake 是否合乎規定? 若不符合則要求 user 重新輸入
   // 然後當然要從 nStone 減去 nTake
}
void computerTurn( ) {
  // 先寫隨便拿但要合乎規定, 就是取 1..min(maxTake, nStone) 的亂數
  // 再想出可贏就會贏的策略 (很簡單 :-)
  // Hint: 拿 max(1, (nStone - xxx) % (1+maxTake) ) 就會贏
  //    其中 xxx 是  0  if lastWin is true
  ///        xxx 是  1  if lastWin == false
}
void playGame( ) {
  // user and the computer take Turn until no more stones left
  // if(userFirst) userTurn( );
  // Loop until no more stone
  //   computerTurn( );
  //   if no more stone then leave the Loop;
  //   userTurn( );
  // end Loop
  // Judge the game result and print some message..
  // 如何判斷誰輸誰贏? 須知道誰拿走最後一個!
  // 注意拿最後一個是輸還是贏?
  //   (lastWin == true 表示拿到最後一個的贏 )
} // playGame
```