**בעיית אסירים ( Prisoners problem)**

מנהל בית הכלא לקח מספר אסירים שנידונו למוות, אסף אותם בחדר אחד ואמר להם שהוא יעשה איתם עסקת שחרור.  
הוא אמר להם שיש חדר מיוחד אין בו שום דבר חוץ מנורה אחת.  
מנהל הכלא יכניס את כל אחד מהאסירים האלו בצורה אקראית, כלומר בסדר כלשהו ובמספר פעמים כלשהו. אף אחד מהם לא יידע כמה פעמים הוא יזמין את כל אחד מהם. בחדר זה, כל אחד יכול להדליק את הנורה, לכבות אותה, או לא לעשות דבר.  
**ואם מתישהו מישהו מהם יוכל לומר למנהל הכלא שכבר כל אחד מהם היה בחדר זה לפחות פעם אחת וזה יהיה נכון באותו רגע, מנהל הכלא ישחרר את כולם.**

יש לאסירים **חמש דקות** לקבוע איך הם יכולים להתמודד עם המצב. בעוד חמש דקות מנהל הכלא יפזר אותם בתאים נפרדים שבהם הם יישארו עד סוף החיים שלהם או עד השחרור.   
**שאלה**: מהי האסטרטגיה של האסירים שתאפשר להם להישאר בחיים?

n - מספר אסירים

למצוא איסטרטגיה לשני מקרים:

1. בתחילת התהליך ידוע שמנורה דלוקה.
2. בתחילת התהליך לא ידוע האם מנורה דלוקה או לא

**להלן האיסטרטגיה:**

אסיר מספר 0 – הוא "המונה"

(n – 1) אסירים אחרים – הם "רגילים"

count - מספר האסירים שהיו בחדר

בתחילת התהליך 0 = count

אם המנורה לא דלוקה, "המונה" מדליק את המנורה בכל ביקור שלו ומוסיף 1 ל count

אחרת, לא עושה כלום.

אסירים אחרים ("רגילים") מכבים את המנורה רק בביקור הראשון.

==========================================

1. **מקרה ראשון: בתחילת התהליך ידוע שמנורה דלוקה -(lamp ON)**

1.1

**Lamp ON**

נכנס אסיר מספר אקראי:

אם זה אסיר "רגיל" והוא בפעם ראשונה, הוא מכבה את המנורה, אחרת, לא עושה כלום.

אם זה אסיר "המונה" והוא בפעם ראשונה ומוסיף 1 ל-count, אחרת, לא עושה כלום.

1.2

**Lamp OFF**

נכנס אסיר מספר אקראי:

אם זה אסיר "רגיל" הוא לא עושה כלום.

אם זה אסיר "המונה" והוא מדליק את המנורה ומוסיף 1 ל-count.

סוף התהליך כאשר count = n .

Psedo-code:

// first the lamp is on

public static int **prisonersLampOn(int n)**

**boolean** enter[] = **new boolean**[n]

**for** (**int** i = 0; i < enter.length; i++)

enter[i] = **false**

end-for

**boolean** lamp = **true**

**// the fist is the counter**

**int** count = 0, steps = 0

**while** (count < n)

steps++

**int** p = (**int**)(Math.*random*()\*n)

**if** (p==0) // the counter enters

**if** (!enter[p])//the counter enters for the first time

enter[p] = **true**

count++

end-if

**if** (!lamp)

lamp = **true**

count++

end-if

**else**

// p>=1, prisoner enters for the first time && the lamp is on

**if** (!enter[p] && lamp)

lamp = **false**

enter[p] = **true**

end-if

end-if

end-while

**boolean** ans = **true**

**for** (**int** i = 0; ans && i < enter.length; i++)

**if** (!enter[i]) ans = **false**

end-for

**if** (!ans)

print (“Fail!”)

**else**

print (“We are free!”)

end-if

**return** steps

**end-prisonersLampOn**

1. **מקרה שני: בתחילת התהליך לא ידוע האם מנורה דלוקה או לא -(lamp Unknown)**

Psedo-code:

// first the lamp is unknown

public static int **prisonersLampUnknow(int n)**

**int** enter[] = **new int**[n]

**for** (**int** i = 0; i < enter.length; i++)

enter[i] = **0**

end-for

**boolean** lamp = ((**int**)(Math.*random*()\*2)==0) ? false : true

**// the fist is the counter**

**int** count = 0, steps = 0

**while** (count < 2\*n)

steps++

**int** p = (**int**)(Math.*random*()\*n)

**if** (p==0 && !lamp ) // the counter enters

lamp = **true**

count++

**if** (enter[0]==0)

enter[0]++

count++

end-if

**else**

// p>=1, prisoner enters for the first and second time && the lamp is on

**if** (enter[p]<2 && lamp)

lamp = **false**

enter[p] ++

end-if

end-if

end-while

**boolean** ans = **true**

**for** (**int** i = 0; ans && i < enter.length; i++)

**if** (enter[i]==0) ans = **false**

end-for

**if** (!ans)

print (“Fail!”)

**else**

print (“We are free!”)

end-if

**return** steps

**end-prisonersLampUnknow**