

הודעה

אביגיל ובת-שבע הן שתי חברות שמתכתבות אחת עם השנייה. לאביגיל יש הודעה M , שהיא רצף של S ביטים (כלומר אפסים או אחדות), שהיא רוצה לשלוח לבת-שבע. אביגיל מתקשרת עם בת-שבע באמצעות שליחת **פקטות**. פקטה היא רצף של 31 ביטים שממוספרים מ-0 עד 30. אביגיל רוצה לשלוח את ההודעה M לבת-שבע על ידי שליחת כמות מסוימת של פקטות אליה.

לרוע המזל, גלית שיבשה את התקשורת בין אביגיל ובת-שבע וכעת היא יכולה להכתיים את הפקטות. ליתר דיוק, בכל פקטה גלית יכולה לשנות ביטים בדיוק ב-15 אינדקסים. ספציפית, יש מערך C באורך 31, שבו כל איבר הוא 0 או 1 עם המשמעות הבאה:

- $C[i] = 1$ מסמן שגלית יכולה לשנות את הביט עם האינדקס i . נקרא לאינדקסים כאלה **נשלטים** על ידי גלית.
- $C[i] = 0$ מסמן שגלית לא יכולה לשנות את הביט עם האינדקס i .

המערך C מכיל בדיוק 15 אחדות ו-16 אפסים. בזמן שליחת ההודעה M , סט האינדקסים שנשלטים על ידי גלית נשאר זהה לכל הפקטות. אביגיל יודעת בדיוק אילו 15 אינדקסים נשלטים על ידי גלית. בת-שבע יודעת רק ש-15 אינדקסים נשלטים על ידי גלית, אבל היא לא יודעת אילו.

תהי A הפקטה שאביגיל החליטה לשלוח (נקרא לה **הפקטה המקורית**). תהי B הפקטה שבת-שבע קיבלה (נקרא לה **הפקטה המוכתמת**). לכל i כך שמתקיים $0 \leq i < 31$:

- אם גלית לא שולטת בביט עם האינדקס i ($C[i] = 0$), בת-שבע מקבלת את הביט i כפי שנשלח על ידי אביגיל ($B[i] = A[i]$).
- אחרת, אם גלית שולטת בביט עם האינדקס i ($C[i] = 1$), הערך של $B[i]$ נקבע על ידי גלית.

מייד לאחר שליחת כל פקטה, אביגיל יודעת איך הפקטה המוכתמת המתאימה לה נראית.

אחרי שאביגיל שולחת את כל הפקטות, בת-שבע מקבלת את כל הפקטות המוכתמות **בסדר שבו הן נשלחו** ועליה לשחזר את ההודעה המקורית M .

המשימה שלכם היא לתכנן ולממש אסטרטגיה שתאפשר לאביגיל לשלוח את ההודעה M לבת-שבע, כך שבת-שבע תוכל לשחזר את M מהפקטות המוכתמות. ספציפית, עליכם לממש שתי פונקציות. הפונקציה הראשונה מבצעת את הפעולות של אביגיל. היא מקבלת הודעה M ואת המערך C , וצריכה לשלוח פקטות כדי להעביר את ההודעה לבת-שבע. הפונקציה השנייה מבצעת את הפעולות של בת-שבע. היא מקבלת את הפקטות המוכתמות וצריכה לשחזר את ההודעה המקורית M .

פרטי מימוש

הפונקציה הראשונה שעליכם לממש היא:

```
void send_message(std::vector<bool> M, std::vector<bool> C)
```

- M : מערך באורך S המתאר את ההודעה שאביגיל רוצה לשלוח לבת-שבע.
- C : מערך באורך 31 המתאר אילו אינדקסים נשלטים על ידי גלית.
- פונקציה זו תיקרא **לכל היותר 2100 פעמים** בכל טסט.

הפונקציה צריכה לקרוא לפונקציה הבאה כדי לשלוח פקטה:

```
std::vector<bool> send_packet(std::vector<bool> A)
```

- A : פקטה מקורית (מערך באורך 31) המתארת את הביטים שנשלחו על ידי אביגיל.
- פונקציה זו מחזירה פקטה מוכתמת B המתארת את הביטים שהתקבלו על ידי בת-שבע.
- מותר לקרוא לפונקציה זו לכל היותר 100 פעמים בכל ריצה של הפונקציה `send_message`.

הפונקציה השנייה שעליכם לממש היא:

```
std::vector<bool> receive_message(std::vector<std::vector<bool>> R)
```

- R : מערך המתאר את הפקטות המוכתמות. הפקטות נוצרו במקור מפקטות שנשלחו על ידי אביגיל בקריאה אחת לפונקציה `send_message` וניתנות **בסדר שבו נשלחו** על ידי אביגיל. כל איבר ב- R הוא מערך באורך 31, המתאר פקטה מוכתמת.
- פונקציה זו צריכה להחזיר מערך של S ביטים שזהה להודעה המקורית M .
- פונקציה זו יכולה להיקרא **מספר פעמים** בכל טסט, **בדיוק פעם אחת** לכל קריאה מתאימה לפונקציה `send_message`. **הסדר של הקריאות לפונקציה `receive_message` אינו בהכרח זהה לסדר הקריאות המתאימות לפונקציה `send_message`.**

שימו לב שבמערכת הבדיקה הקריאות לפונקציות `send_message` ו-`receive_message` מתבצעות **בשתי תוכניות נפרדות**.

מגבלות

- $1 \leq S \leq 1024$
- במערך C בדיוק 31 איברים, מתוכם 16 שווים ל-0 ו-15 שווים ל-1.

תתי משימות וניקוד

אם בטסט כלשהו הקריאות לפונקציה `send_packet` לא מקיימות את התנאים שצוינו לעיל, או אם ערך החזרה של קריאה כלשהי לפונקציה `receive_message` אינו נכון, הניקוד של הפתרון בטסט הזה יהיה 0.

אחרת, תהי Q כמות הקריאות המקסימלית לפונקציה `send_packet` מתוך כל ההרצות של `send_message` בכל הטסטים. בנוסף, יהי X שווה ל:

- $$\begin{aligned} Q \leq 66 & \quad \alpha, 1 \bullet \\ 66 < Q \leq 100 & \quad \alpha, 0.95^{Q-66} \bullet \end{aligned}$$

אז הניקוד יחושב כך:

תת משימה	ניקוד	מגבלות נוספות
1	$10 \cdot X$	$S \leq 64$
2	$90 \cdot X$	ללא מגבלות נוספות.

שימו לב שבחלק מן המקרים ההתנהגות של הגריידר יכולה להיות **אדפטיבית**. זה אומר שהערכים שיוחזרו על ידי `send_packet` עלולים להיות תלויים לא רק בקלט של הפונקציה, אלא גם בדברים אחרים, כולל הקלטים וערכי החזרה של הקריאות הקודמות לפונקציה זו ומספרים פסאודו-רנדומליים שהגריידר מייצר. הגריידר **דטרמיניסטי** במובן שאם מריצים אותו פעמיים ובשתי ההרצות נשלחות אותן פקטות, הוא יעשה בהן את אותם שינויים.

דוגמה

הביטו בקריאה הבאה.

```
send_message([0, 1, 1, 0],
             [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
              1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

ההודעה שאביגיל מנסה לשלוח לבת-שבע היא $[0, 1, 1, 0]$. הביטים עם האינדקסים מ-0 עד 15 לא יכולים להשתנות על ידי גלית, אך הביטים עם האינדקסים מ-16 עד 30 כן יכולים להשתנות על ידי גלית.

לצורך דוגמה זו, בואו נניח שגלית ממלאת ביטים עוקבים שהיא שולטת בהם עם 0 ועם 1 לחילופין, כלומר היא קובעת את האינדקס הראשון שהיא שולטת בו (האינדקס 16 במקרה שלנו) להיות 0, את האינדקס השני שהיא שולטת בו (האינדקס 17) להיות 1, את האינדקס השלישי שהיא שולטת בו (האינדקס 18) להיות 0, וכן הלאה.

אביגיל יכולה להחליט לשלוח שני ביטים מההודעה המקורית בפקטה אחת כדלקמן: היא תשלח את הביט הראשון ב-8 האינדקסים הראשונים שהיא שולטת בהם ואת הביט השני ב-8 האינדקסים הבאים שהיא שולטת בהם.

אז אביגיל בוחרת לשלוח את הפקטה הבאה:

```
send_packet([0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
            0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

שימו לב שגלית יכולה לשנות את הביטים עם 15 האינדקסים האחרונים, אז אביגיל יכולה לקבוע אותם באופן שרירותי, שכן הם עלולים להדרס. עם האסטרטגיה המתוארת של גלית, הפונקציה מחזירה:

$K \ L$
 $D[0] \ D[1] \ \dots \ D[L-1]$

כאן, K זה מספר הקריאות ל- send_packet , D זו ההודעה שהוחזרה על ידי receive_message ו- L הוא האורך שלה.