

משחק כיתה

בית ספר תיכון בולט בטשקנט חוגג את יום השנה שלו באירוח משחק בין תלמידים למורים. ישנם N תלמידים שיושבים בכיתה, הממוספרים מ-0 עד $N - 1$. כל תלמיד מחזיק דף נייר המכיל מערך של מספרים שלמים. תחילה, כל הדפים ריקים, אז כל דף מכיל מערך ריק.

התלמידים משחקים במשחק עם M מורים, הממוספרים מ-0 עד $M - 1$, ועם המנהל. המשחק מתנהל ב- M שלבים, ממוספרים גם הם מ-0 עד $M - 1$. בשלב j , מורה j נכנס לכיתה והאירועים הבאים מתרחשים:

- חלק (ייתכן שאפס) מהתלמידים מרימים את היד. מובטח שבכל שלב לפחות תלמיד אחד לא מרים את היד שלו, וכל תלמיד יכול להרים את היד שלו **לכל היותר פעם אחת** במהלך כל המשחק.
- המורה מסתכל על דפי הנייר שהתלמידים מחזיקים כעת, והוא יכול **לשנות** את הדפים המוחזקים על ידי תלמידים שלא מרימים את היד שלהם בשלב זה. עבור כל תלמיד כזה, המורה יכול להחליף את התוכן של הדף שלו במערך חדש (ייתכן שריק) של מספרים שלמים. כל מספר שלם חייב להיות בין 0 לבין 63, כולל, והמערך יכול להכיל לכל היותר 63 איברים.
- לאחר שינויים אלו, מורה j עוזב והתלמידים מחליפים ביניהם את הדפים לפי תמורה סודית P . כלומר, כל תלמיד i ($0 \leq i < N$) נותן את הדף שלו לתלמיד $P[i]$, כאשר $P[0], P[1], \dots, P[N - 1]$ הם N מספרים שלמים שונים בין 0 לבין $N - 1$, כולל. הערכים של P **קבועים** במהלך כל שלבי המשחק, אבל המורים והמנהל לא יודעים אותם.

לאחר שכל M השלבים הסתיימו וההחלפה האחרונה בהתאם ל- P בוצעה, נכנס המנהל. תוך התבוננות אך ורק בדפים שבידי התלמידים, על המנהל לקבוע, עבור כל תלמיד i ($0 \leq i < N$), את מספר השלב j שבו התלמיד i הרים את ידו, או להסיק כי התלמיד i מעולם לא הרים את ידו.

המורים אינם יכולים לתקשר עם מורים אחרים או עם המנהל, למעט באמצעות מערכי המספרים השלמים הכתובים על הדפים. כל מורה יודע באיזה שלב הוא נכנס לכיתה.

המשימה שלכם היא לתכנן ולממש אסטרטגיה עבור המורים והמנהל, שתזהה נכונה האם ומתי כל תלמיד הרים את ידו. הציון שלכם ייקבע על פי האורך המרבי של מערך שנכתב על אחד הדפים: אורך מקסימלי קצר יותר יניב ציון שווה או גבוה יותר.

פרטי מימוש

עליכם לממש שתי פונקציות, אחת עבור המורים ואחת עבור המנהל.

הפונקציה שעליכם לממש עבור המורים היא:

```
std::vector<std::vector<int>> process_step(
    int N, int M, int R,
    std::vector<int> T,
    std::vector<std::vector<int>> A)
```

- N : מספר התלמידים.
 - M : מספר השלבים, וגם מספר המורים.
 - R : מספר השלב הנוכחי (מ-0 עד $M - 1$).
 - T : מערך (ייתכן שריק) הבנוי מהאינדקסים של התלמידים שהרימו את ידם בשלב זה, ממיון בסדר עולה.
 - A : מערך באורך N המתאר את הדפים, שבו $A[i]$ ($0 \leq i < N$) הוא מערך המספרים השלמים על הדף שמחזיק התלמיד i בתחילת השלב.
 - פונקציה זו תיקרא בדיוק M פעמים בכל משחק, לפי הסדר $R = 0, 1, \dots, M - 1$.
- על הפונקציה להחזיר מערך B באורך N , המתאר את הדפים לאחר השינויים של המורה, באותו הפורמט כמו A .
- עבור כל תלמיד i שהרים את ידו (כלומר, i מופיע ב- T), אסור לשנות את הדף, לכן $B[i] = A[i]$.
 - עבור כל i המקיים $0 \leq i < N$, אסור שהאורך של $B[i]$ יחרוג מ-63, וכל מספר שלם ב- $B[i]$ חייב להיות בין 0 לבין 63, כולל.
- הפונקציה שעליכם לממש עבור ההמנהל היא:

```
std::vector<int> determine_steps(
    int N, int M, std::vector<std::vector<int>> A)
```

- M, N : זהים ללמעלה.
 - A : מערך באורך N , שבו $A[i]$ זה המערך של המספרים השלמים על הדף שמוחזק על ידי תלמיד i לאחר כל M השלבים.
 - פונקציה זו נקראת בדיוק פעם אחת בכל משחק, אחרי הקריאה האחרונה ל-`process_step`.
- על פונקציה זו להחזיר המערך D באורך N . עבור כל i המקיים $0 \leq i < N$, חייב להתקיים:
- $D[i] = j$ אם התלמיד i הרים את ידו במהלך שלב j , או
 - $D[i] = -1$ אם התלמיד i אף פעם לא הרים את ידו במהלך המשחק.
- התוכנית שלכם אינה רשאית לאחסן או להעביר שום מידע מקריאה אחת ל- `process_step` לאחרת, למעט באמצעות ערך ההחזרה של `process_step`.** אם התוכנית שלכם תנסה לעשות זאת, הציון שלכם ב-CMS עלול להיות שגוי ועלול להיות מופחת לאחר סיום התחרות. שימו לב שהגריידר יכול להריץ מספר מופעים של התוכנית שלכם בו זמנית. משמעות הדבר היא שקריאות ל- `process_step` ול- `determine_steps` מאותו משחק עלולות להתבצע במופעים שונים, וקריאות ל- `process_step` ול- `determine_steps` ממשחקים שונים עלולות להתבצע באותו מופע. כל מקרה בדיקה כולל עד 5 משחקים.

אילוצים

- $2 \leq N \leq 63$

- $1 \leq M \leq 63$
- כל תלמיד מרים את ידו **לכל היותר פעם אחת** במהלך המשחק כולו. במילים אחרות, עבור כל תלמיד i , יש לכל היותר צעד אחד j שבו i מרים את ידו.
- בכל צעד, לפחות תלמיד אחד **לא** מרים את ידו.

תתי משימות וניקוד

תת משימה	ניקוד	אילוצים נוספים
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ לכל $0 \leq i \leq N - 1$
4	25	לכל היותר תלמיד אחד מרים את ידו בכל שלב.
5	56	ללא אילוצים נוספים.

בכל מקרה בדיקה, הניקוד שלך יהיה 0 (מדווח כ-CMS Output isn't correct) אם ערך החזרה של קריאה כלשהי ל-process_step לא תקין או אם ערך החזרה של קריאה כלשהי לפונקציה determine_steps לא נכון.

אחרת, יהי C האורך המירבי של מערך **כלשהו** ב- B **כלשהו** שהוחזר על ידי process_step. אז, הניקוד שלכם במקרה הבדיקה בתת משימה עם ניקוד S יחושב כ- $S \cdot X$, כאשר X תלוי ב- C בהתאם לטבלה הבאה:

תנאי	X
$C \leq 2$	1.00
$C = 3$	0.75
$C = 4$	0.55
$5 \leq C \leq 13$	$0.50 - 0.03 \cdot (C - 5)$
$14 \leq C \leq 63$	$0.19 \cdot \frac{64-C}{64-14} + 0.04$

דוגמה

התבוננו על תרחיש עם $N = 4$ תלמידים, $M = 2$ מורים, ותמורה $P = [0, 3, 1, 2]$. תחילה, כל הדפים ריקים.

הגריידר מבצע תחילה את הקריאה:

```
process_step(4, 2, 0, [1], [[], [], [], []])
```

תלמיד 1 הרים את ידו, אז לא ניתן לשנות את הדף שלו. מורה 0 יכול להחליט לרשום [10] על הדף של תלמיד 0, לכתוב [1, 63, 4] על הדף של תלמיד 3, ולהשאיר את הדף של תלמיד 2 ריק. כדי לעשות זאת, על הפונקציה להחזיר $B = [[10], [], [], [1, 63, 4]]$.

אחרי שמורה 0 עוזב, התלמידים מחליפים ביניהם את הדפים שלהם לפי P . לאחר ההחלפה, הדפים הם $A = [[10], [], [1, 63, 4], []]$.

הגריידר כעת מבצע את הקריאה:

```
process_step(4, 2, 1, [0,3], [[10], [], [1,63,4], []])
```

תלמידים 0 ו-3 הרימו את ידם, אז לא ניתן לשנות את הדפים שלהם. מורה 1 יכול להחליט לרשום $[0, 40]$ על הדף של תלמיד 1 ולכתוב $[1, 50]$ על הדף של תלמיד 2. כדי לעשות זאת, על הפונקציה להחזיר $B = [[10], [0, 40], [1, 50], []]$.

אחרי שמורה 1 עוזב, דפי הנייר שוב מוחלפים לפי P . לאחר ההחלפה, הדפים הם $A = [[10], [1, 50], [], [0, 40]]$.

לבסוף, הגריידר מבצע את הקריאה:

```
determine_steps(4, 2, [[10], [1,50], [], [0,40]])
```

על פונקציה זו להחזיר את המערך $D = [1, 0, -1, 1]$ משום שתלמידים 0 ו-3 הרימו את ידם בשלב 1, תלמיד 1 הרים את ידו בשלב 0, ותלמיד 2 אף פעם לא הרים את ידו. בדוגמה זו, $C = 3$.

גריידר לדוגמה

פורמט קלט:

```
N M
Q[0] Q[1] ... Q[N-1]
P[0] P[1] ... P[N-1]
```

כאן, Q זה מערך באורך N , כאשר $Q[i] = j$ אם תלמיד i הרים את ידו במהלך שלב j , או $Q[i] = -1$ אם תלמיד i לעולם לא הרים את ידו במהלך המשחק כולו.

פורמט פלט:

לאחר כל קריאה ל-`process_step`, הגריידר לדוגמה מדפיס כפלט את התוכן של הדפים. יהי K האורך של B ו- $L[i]$ האורך של $B[i]$ (לכל $0 \leq i < K$). הגריידר לדוגמה מדפיס את B בפורמט הבא, ושורה ריקה אחרי זה:

```
K
L[0] B[0][0] B[0][1] ... B[0][L[0]-1]
L[1] B[1][0] B[1][1] ... B[1][L[1]-1]
:
L[K-1] B[K-1][0] B[K-1][1] ... B[K-1][L[K-1]-1]
```

הגריידר לדוגמה אחר כך מחליף את הדפים לפי התמורה P . אם $K \neq N$, הגריידר לדוגמה מדפיס הודעת שגיאה ומפסיק את ריצתו במקום.

לאחר הקריאה ל-determine_steps, הגריידר לדוגמה מדפיס:

```
C H
D[0] D[1] ... D[H-1]
```

כאן, H הוא אורך המערך D המוחזר על ידי determine_steps.