



International Olympiad in Informatics 2015

26th July - 2nd August 2015

Almaty, Kazakhstan

Day 1

scales

Language: he-IL

Scales

למינה (Amina) יש שישה מטבעות, הממוספרים מ-1 עד 6. היא יודעת שלכל אחד מהמטבעות יש משקל שונה. היא רוצה למיין אותם לפי משקליהם מהקל לכבד. לצורך כך היא פיתחה סוג חדש של מאזניים (scales).

למאזניים רגילים יש שתי כפות (pans). כדי להשתמש במאזניים כאלה, צריך לשים מטבע בכל כף, והמאזניים יכריעו איזה מהמטבעות כבד יותר.

המאזניים של מינה מורכבים יותר. יש ארבע כפות, המסומנות A, B, C, D . למאזניים יש ארבע קונפיגורציות שונות (settings), ובכל קונפיגורציה המאזניים מכריעים בנוגע לשאלה שונה לגבי המטבעות. כדי להשתמש במאזניים, מינה צריכה לשים בדיוק מטבע אחד בכל אחת מהכפות A, B ו- C . בנוסף, בקונפיגורציה הרביעית היא צריכה לשים בדיוק מטבע אחד גם בכף D .

ארבע הקונפיגורציות יכתבו למאזניים לענות על ארבעת סוגי השאלות הבאים:

1. איזה מבין המטבעות שבכפות A, B, C הוא הכבד ביותר?

2. איזה מבין המטבעות שבכפות A, B, C הוא הקל ביותר?

3. איזה מבין המטבעות שבכפות A, B, C הוא החציון? (זהו המטבע שאינו הכבד ביותר ואינו הקל ביותר מבין השלושה).

4. מבין המטבעות שבכפות A, B, C , נתבונן רק במטבעות שכבדים יותר מהמטבע שנמצא ב- D . אם יש מטבעות כאלה, איזה מהם הקל ביותר?

אחרת, אם אין מטבעות כאלה, איזה מהמטבעות A, B, C הוא הקל ביותר?

משימה

כיתבו תכנית שתמייין את ששת המטבעות של מינה לפי משקליהם. התכנית יכולה לשלוח שאילתות למאזניים של מינה כדי להשוות משקלים של מטבעות. תכניתכם תקבל כמה מקרי בדיקה (test cases) שונים לפתור, כאשר בכל מקרה בדיקה יש סט חדש של שישה מטבעות.

תכניתכם צריכה לממש את הפונקציה `init` ואת הפונקציה `orderCoins`. בכל הרצה של התכנית, ה-`grader` (תכנית הבדיקה של המארגנים) יקרא תחילה ל-`init` פעם אחת בדיוק. כך תקבלו את מספר מקרי הבדיקה, ותוכלו לאתחל משתנים. לאחר מכן ה-`grader` יקרא ל-`orderCoins` פעם אחת עבור כל מקרה בדיקה.

• הפונקציה `init(T)`:

• הפרמטר T : מספר מקרי הבדיקה שתכניתכם תידרש לפתור בהרצה הנוכחית. מובטח ש- T הוא מספר שלם בטווח $1, \dots, 18$.

• הפונקציה אינה מחזירה ערך.

• הפונקציה `orderCoins()`:

• הפונקציה הזו נקראת בדיוק פעם אחת עבור כל מקרה בדיקה.

• הפונקציה צריכה להכריע את הסדר הנכון של ששת המטבעות של מינה, על ידי קריאות כרצונכם לפונקציות `getNextLightest()` או `getLightest()` או `getMedian()` או `getNextLightest()`.

• כאשר הפונקציה יודעת את הסדר הנכון של המטבעות, עליה לדווח את הסדר על ידי קריאה לפונקציה `answer()`.

• הפונקציה צריכה לסיים לאחר הקריאה ל-`answer()`. אין לה ערך החזרה.

הפונקציות הבאות יסופקו על ידי ה-`grader`:

- הפונקציה $answer(W)$: תכניתכם צריכה לקרוא לפונקציה זו עם התשובה שמצאתם.
- הפרמטר W : מערך באורך 6 שמכיל את סדר המטבעות הנכון. הערכים $W[0]$ עד $W[5]$ צריכים להיות מספרי המטבעות (כלומר, מספרים בין 1 ל-6) ממיונים מהמטבע הקל למטבע הכבד.
- תכניתכם צריכה לקרוא לפונקציה זו אך ורק בתוך $orderCoins()$, פעם אחת עבור כל מקרה בדיקה.
- הפונקציה אינה מחזירה ערך.
- הפונקציות $getLightest(A, B, C)$, $getHeaviest(A, B, C)$, $getMedian(A, B, C)$ - פונקציות אלה מתאימות לקונפיגורציות 1, 2 ו-3 בהתאמה במאזניים של מינה.
- הפרמטרים A, B, C : המטבעות שיוצבו בכפות A, B, C בהתאמה. הפרמטרים A, B, C צריכים להיות שלושה מספרים שלמים שונים, כל אחד בין 1 ל-6 כולל.
- כל אחת מהפונקציות מחזירה את אחד המספרים מבין A, B, C : מספר המטבע שעונה על השאלה. לדוגמה, $getHeaviest(A, B, C)$ מחזירה את מספרו של המטבע הכבד ביותר מבין השלושה.
- הפונקציה $getNextLightest(A, B, C, D)$ - פונקציה זו מתאימה לקונפיגורציה מספר 4 במאזניים של מינה.
- הפרמטרים A, B, C, D : המטבעות שיושמו בכפות A, B, C, D בהתאמה. הפרמטרים A, B, C, D צריכים להיות ארבעה מספרים שלמים שונים, כל אחד בין 1 ל-6 כולל.
- הפונקציה מחזירה את אחד המספרים A, B, C : מספר המטבע שנבחר על ידי המאזניים כפי שתואר בקונפיגורציה מספר 4. כלומר, המטבע המוחזר הוא הקל ביותר מבין המטבעות שבכפות A, B, C אשר כבדים יותר מהמטבע בכף D ; או, אם אף אחד מהם אינו כבד מהמטבע שבכף D , המטבע המוחזר הוא הקל מבין המטבעות שבכפות A, B, C .

ניקוד

- בשאלה זו אין תת משימות. במקום זאת, הניקוד שלכם יוחלט על פי מספר השקילות (הכמות הכוללת של קריאות לפונקציות $getLightest()$, $getHeaviest()$, $getMedian()$ ו- $getNextLightest()$ שתכניתכם מבצעת).
- תכניתכם תורץ כמה פעמים, ובכל הרצה יהיו כמה מקרי בדיקה. נגדיר את r להיות מספר ההרצות. המספר r הוא קבוע. אם תכניתכם תטעה במיין המטבעות באיזשהו מקרה בדיקה, אפילו רק באחת ההרצות, הציון הכולל על כל ההרצות יהיה 0. אחרת, ההרצות מנוקדות כל אחת בנפרד כמתואר להלן.
- נגדיר את Q להיות המספר הקטן ביותר כך שניתן למיין כל סדרה של שישה מטבעות בעזרת Q שקילות במאזניים של מינה. כדי להוסיף לאתגר של הבעיה, לא נחשוף כאן את הערך של Q .
- נניח שמספר השקילות הגדול ביותר שהשתמשתם בו, בכל מקרי הבדיקה של כל ההרצות, הוא $Q + y$ עבור y שלם כלשהו. נתבונן בהרצה יחידה כלשהי של התכנית. נגדיר את מספר השקילות הגדול ביותר שתכניתכם ביצעה בכל T מקרי הבדיקה בהרצה הזאת להיות $Q + x$ עבור x שלם אי-שלילי כלשהו (אם השתמשתם בפחות מ- Q שקילות בכל מקרי הבדיקה בריצה, אז $x = 0$). אז הניקוד עבור ההרצה הזו יהיה $\frac{100}{r((x+y)/5+1)}$ מעוגל לפטה לשתי ספרות אחרי הנקודה העשרונית.
- בפרט, אם תכניתכם מבצעת לכל היותר Q שקילות בכל מקרה בדיקה של כל הרצה, תקבלו 100 נקודות.

דוגמה

נניח שהמטבעות ממיונים 1 5 2 1 4 3 מהקל לכבד.

| קריאה לפונקציה | ערך החזרה | הסבר |
|-------------------------------|-----------|--|
| $getMedian(4, 5, 6)$ | 6 | מטבע 6 הוא החציון מבין 4, 5, 6. |
| $getHeaviest(3, 1, 2)$ | 1 | מטבע 1 הוא הכבד ביותר מבין 1, 2, 3. |
| $getNextLightest(2, 3, 4, 5)$ | 3 | המטבעות 2, 3, 4, 5 כולם קלים ממטבע 5, לכן הקל ביותר מביניהם (3) מוחזר. |
| $getNextLightest(1, 6, 3, 4)$ | 6 | המטבעות 1 ו-6 שניהם כבדים ממטבע 4. מתוך 1 ו-6, מטבע 6 הוא הקל יותר. |
| $getHeaviest(3, 5, 6)$ | 5 | מטבע 5 הוא הכבד ביותר מבין 3, 5, 6. |
| $getMedian(1, 5, 6)$ | 1 | מטבע 1 הוא החציון מבין המטבעות 1, 5, 6. |
| $getMedian(2, 4, 6)$ | 6 | מטבע 6 הוא החציון מבין המטבעות 2, 4, 6. |
| $answer([3, 4, 6, 2, 1, 5])$ | | התכנית מצאה את התשובה הנכונה במקרה הבדיקה הזה. |

תכנית grader לדוגמה

תכנית ה-grader לדוגמה מקבלת קלט בפורמט הבא:

- שורה 1: המספר T , מספר מקרי הבדיקה.
- בכל אחת מהשורה ה-2 עד השורה ה- $T + 1$: סדרה של 6 מספרים שונים בין 1 ל-6: סדר המטבעות מהקל לכבד.

לדוגמה, להלן קלט שמכיל שני מקרי בדיקה, כאשר בראשון סדר המטבעות הוא 1 2 3 4 5 6 ובשני סדר המטבעות הוא 3 4 6 2 1 5:

```
2
1 2 3 4 5 6
3 4 6 2 1 5
```

תכנית ה-grader לדוגמה תדפיס את המערך שהועבר כפרמטר אל הפונקציה `answer()`.