

דפי הסברים להאקתון

פרטי הקבוצה:

Team ID:

Team Name:

Team Leader:

Password:

סימונים:

+ יסמן שרשור של מחרוזות (כרגיל בjava)

H יסמן את פונקציית הגיבוב secretHash, רמת הקושי בהתאם לכתוב בשאלה.

== יסמן שיוויון, כגון בין 2 מחרוזות (בjava יש לכתוב string1.Equals(string2))

≠ יסמן אי-שיוויון, כגון בין 2 מחרוזות (בjava יש לכתוב !string1.Equals(string2))

דוגמה: אם string1="ab" ו string2="cd" אז H(string1 + string2) הינו הערך המוחזר מפונקציית הגיבוב secretHash על הקלט "abcd".

התנגשות – שתי מחרוזות string1≠string2 כך ש H(string1)=H(string2)

דף הסבר על הקבצים המצורפים

קבלתם מאיתנו 2 קבצים (נמצאים באתר הקורס, בלשונית hackathon):

1. BlockchainClient.jar

- יש להריץ את הקובץ מה-command line (בwindows ע"י הרצת cmd)
- מתוך התיקיה יש להפעיל את הקובץ ע"י הפקודה
`java -jar BlockchainClient.jar <Server-IP>`
- ולאחר מכן לעקוב אחר ההוראות.
- וידאו עם הסבר מפורט מופיע באתר הקורס בלשונית ההאקתון.

2. SecretHash.java

- אין לשנות את הקובץ
 - יש לשנות את רמת הקושי אך ורק ע"י קריאה לפונקציה `setDifficulty`
 - הקובץ מכיל 2 פונקציות:
1. `String secretHash(String string)`
פונקציה הגיבוב – פונקציה זו מקבלת מפתח (מחרוזת) ומחזירה ערך הקסדצימלי (כמחרוזת).
שימו לב: ערך הפלט מושפע גם מרמת הקושי.
 2. `void setDifficulty(int i)`
משנה את רמת הקושי של פונקציית הגיבוב.
ככל שרמת הקושי גבוהה יותר כך קשה יותר למצוא התנגשויות.

הערות חשובות לגבי רמת הקושי של פונקציית הגיבוב:

1. רמות הקושי נעות בין 0 ל-10. ככל שרמת הקושי גבוהה יותר, כך קשה יותר למצוא התנגשויות. רמות הקושי הרלוונטיות לתחרות הינן 1-9.
2. יש לשנות את רמת הקושי אך ורק ע"י הפונקציה `setDifficulty`.
3. בדף התרגילים, בכל סעיף מצויינת רמת הקושי של פונקציית הגיבוב הדרושה באותו סעיף.
4. רמת הקושי תשתנה במהלך התחרות. כלומר, תדרשו לעדכן את הרמה בקוד שלכם ב"לייב".

דף הסבר על המבנה הכללי של ה block-chain שלנו:

הבלוק צ'יין שלנו הינו רשימה משורשרת, כאשר כל בלוק הינו חוליה בשרשרת, ומתקיימים התנאים שמוסברים בהמשך. לפניכם קוד חלקי של הבלוק צ'יין ושל הבלוק:

```
class Blockchain{
    Block head;//Newest block – last block that has been mined
    Block tail;//First block ever mined – challenge is "hackathon"
    int blockchainLength;
    ...
}

class Block{
    Block prev, next;
    int difficulty;
    String teamID;
    String challenge;
    String string1, string2;
    ...
}
```

- בכל בלוק בשרשרת עבורו $next \neq null$ מקיים את התנאים הבאים:
 1. $this.challenge == H(next.string1 + next.string2);$
 2. $H(challenge+teamID+string1) == H(challenge+teamID+string2);$
 - אם $next == null$, אז במקום תנאי 1 מתקיים $challenge == "hackathon"$ (תנאי 2 עדיין צריך להתקיים).
- הערה:** במהלך התחרות, לא תצטרכו לחשב את ה challenge (הוא יחושב על-ידנו ויינתן לכם).
- שימו לב:** H הינה פונקציית הגיבוב secretHash עם דרגת הקושי (difficulty) המצויין בבלוק.