הבטים בהוראת מדעי המחשב

שבט התשע"ז - פברואר 2017

מערכת העיתון: "מחשבה"

עריכה לשונית וגרפית: "מחשבה"

עיצוב הכריכה: דפוס הטכניון

כתובת לפניות:

מערכת העיתון "הבטים בהוראת מדעי המחשב"

מחשבה - מרכז המורים הארצי למדעי המחשב

הפקולטה לחינוך למדע וטכנולוגיה

הטכניון, חיפה 3200003

טלפון: 8292880 - 04, פקס: 8293004 - 04

דוא"ל: hebetim@tx.technion.ac.il

גליון מספר 45

הפרויקט מבוצע על-פי מכרז 9.7/2013 - עבור המזכירות הפדגוגית, משרד החינוך.

© כל הזכויות שמורות למשרד החינוך

ISSN 0793-2685

תוכן הגליון

מאמר המערכת 2

הכנס הארצי וקורס להוראה יצירתית במדעי המחשב 3

אי שימוש בפונקציות בקרב תלמידי מדעי המחשב בתיכון

אחמד עומר 6

על אקראיות, מקריות ומזל

אודי מלכה 10

תחרויות תכנות 15

סמינר קיץ תשע"ו למורים מובילים 27

טופס משוב 32

קראתם את העיתון?

אל תשכחו למלא את טופס המשוב

הנמצא בעמוד האחרון

ולשלוח אותו אל מינהלת מל"מ.

כתובתנו:

מרכז המורים הארצי למדעי המחשב

הפקולטה לחינוך למדע וטכנולוגיה, טכניון, חיפה 3200003

טלפון 04-8292880 , פקס 04-8293004

מאמר מערכת

קוראים יקרים

לפניכם הגליון הראשון של "הבטים בהוראת מדעי המחשב" לשנת תשע"ז. אתם מוזמנים לקרוא אותו באתר ללא תשלום ולאחר מכן למלא את טופס המשוב ולשלוח אותו למינהלת מל"מ. שליחת דפי המשוב היא התנאי להמשך קיומו של העיתון.

מה בגיליון הפעם?

המאמר של אחמד עומר עוסק ב**אי שימוש בפונקציות בקרב תלמידי מדעי המחשב בתיכון**. אחמד שם לב במהלך עבודתו כמורה שהתלמידים לא משתמשים בפונקציות (מתודות) כאשר הם נדרשים לפתור בעיה תכנותית, למרות שהם למדו את הנושא ושולטים בו בצורה די טובה. מטרת המחקר שלו היא לבדוק תופעה זו לעומק, להבין את הסיבות והגורמים מאחוריה ולהציע דרכים להתגבר עליה.

המאמר של אודי מלכה, עוסק ב**אקראיות, מקריות ומזל**. האם בחירה של מספר אקראי על ידי אדם היא באמת אקראית? מהי חשיבותה של האקראיות בחיינו ומה אקראי סביבנו? כיצד המחשב "בוחר" מספר אקראי והאם הוא באמת אקראי? איך יודעים אם קבוצת מספרים היא קבוצה של מספרים אקראיים? דיון בכיתה בנושאים אלו עם הפעלות חווייתיות יכול להוות פתיח ללימוד אודות מספרים אקראיים ובסיס להבנה לקראת מימוש הידע בשפת התכנות הנלמדת. על אקראיות במחשב, בכיתה ומחוצה לה, במאמר שלפניכם.

קיימות ברשת תחרויות תכנות רבות. בחרנו להביא לכם מספר דוגמאות של שאלות מסוגים שונים וברמות שונות של קושי. אפשר להשתמש בשאלות האלה בכיתה או לעודד את התלמידים להשתתף בתחרויות השונות.

תוכלו למצוא בגיליון גם דיווח על כנס המורים הארצי שהתקיים במכללה ע"ש הרמלין בנתניה בתאריך 27.12.16, וסמינר קיץ תשע"ו למורים מובילים.

שימו לב לקורס כלים להוראה יצירתית במדעי המחשב, שיפתח במרץ בתל אביב. יעביר אותו אריק הוד. פרטים נוספים בגיליון.

מומלץ לעקוב אחר ההודעות השוטפות על פעילויות המרכז הארצי באתר האינטרנט שלנו. אתם מוזמנים להמשיך לגלוש באתר, לקרוא ולהוריד קבצים.

נשמח גם לקבל מכם חומרים מפרי עטכם בכל נושא שיכול לעניין את קוראי העיתון.

קריאה מהנה,
ממערכת העיתון
ומצוות המרכז הארצי

כנס המורים הארצי תשע"ז

בים שלישי 27.12.2016 התקיים כנס המורים הארצי במכללת אורט הרמלין בנתניה.
למרות מזג האוויר הסוער הגיעו מורים רבים.
להלן תכנית הכנס:

פתיחת הכנס: ד"ר אבי כהן, מפמ"ר מדעי המחשב ד"ר אריאל חורין, מנהלת מכללת אורט הרמלין ד"ר תמי לפידות, מנהלת מרכז המורים הארצי למדעי המחשב ליאת בודרקר ומשה שטיינר: הודעה על סדנה למורים חדשים
פרופ' נועם ניסן, האוניברסיטה העברית רשמים מהוראת קורס המבוא למדעי המחשב בשפת פייתון באוניברסיטה העברית
הפסקה קצרה
ד"ר עופר רימון, מנהל המינהל למדע וטכנולוגיה, משרד החינוך
שגיא בר: פרויקט מגשימים, תכנית הסייבר הלאומית
אל"מ גיל גור אריה, מפקד המרכז הטכנולוגי ב- 8200: Big Data הצורך והחשיבות בנושא
פרופ' שלומי דולב, יו"ר ועדת מקצוע הרצף של תכנית הלימודים במדעי המחשב: יסודי, חטיבת ביניים, תיכון
ארוחת צהריים קלה ותערוכת חומרי לימוד
מושבים מקבילים: • מדעי המחשב בראי גלובלי (סדנה): אריק הוד • אמת או שקר (הוראת טיפוס בוליאני): אסתי אפלבוים • טיפים להוראת מדעי המחשב וחומרי למידה שונים: דפנה מינסטר • אפליקציות לתלמידים באנדרויד: יואב מגיד

תודה לכל המרצים והמשתתפים. להתראות בכנס בשנה הבאה.

כלים להוראה יצירתית במדעי המחשב

עולם מדעי המחשב שואב כלים וידע מכלל העולמות המקיפים אותו, תורם להם ונתרם מהם בכל תחומי חיינו, בעידן הדיגיטלי ובעידן הקודם (ואף בעידן הבא).

בעולם ההוראה, תפקיד הבנת העולם הסובב את תחומנו הינו חשוב עשרות מונים מתפקידו המקורי: קיים צורך בחשיבה המופשטת הוליד צורך בהבנת תהליכים רבים, חלקם מורכבים.

עולם מדעי המחשב עומד לצד עולמות נהדרים הנעזרים בו ועלינו, כמורים, להביא עמנו אל בין כתלי הכיתה/המעבדה את העולם הסובב אותנו, המורים, ואת תלמידינו. בנוסף, תפקיד המורה הינו דו כיווני: לאפשר לכל תלמידה ותלמיד להביא לידי ביטוי את עולמם אל המקצוע.

בהשתלמות זו נחפש יחדיו דרכים יצירתיות להוראת המקצוע. נדון בדרכים בהן ניתן להנגיש את המקצוע לתלמידים בדרכים יצירתיות ולא שגרתיות, הנובעות מחשיבה משותפת שלנו, קהילת המורים למדעי המחשב.

הקורס יתקיים בגימנסיה העברית הרצליה בתל אביב, בשעות 16:00-19:00 בימי חמישי, ויעביר אותו אריק הוד.

תאריכי המפגשים: יום חמישי 2.3.17, יום חמישי 9.3.17, יום חמישי 16.3.17, יום חמישי 23.3.17, יום חמישי 30.3.17, יום חמישי 20.4.17 (סך הכל 30 שעות).

כל מפגש 4 שעות, כמו כן יינתנו שתי מטלות.

מורים שיגיעו בזמן יוכלו לחנות בחניית המורים של הגימנסיה.

תכני 10 המפגשים:

- חדשנות בהוראה.
  - מהי חדשנות?
  - האם חדשנות חייבת להיות דיגיטלית/טכנולוגית?
  - כיצד מחדשים בתוך המסגרת ומחוצה לה?
  - מבט אל יחידות לימוד נבחרות- ויישום חדשנות בהוראה.
- הביצה והתרנגולת: חדשנות, יצירתיות והוראה.
  - מבט פילוסופי על מדעי המחשב.
  - מי המציא את מי?
  - מי משרת את מי?
  - כיצד מגבירים את השטף?
- אנשים טובים באמצע הדרך
  - כיצד ניתן ללמוד מכל אדם.
  - למידה אקראית מכוונת.
- שיתופיות
  - שיתוף חומרי לימוד.
  - איתור חומרי לימוד אף במקומות לא צפויים.
- צר עולמי
  - כיצד ניתן להביא עולמות אחרים אל עולם הוראת מדעי המחשב?
  - הוראה אינטר דיסציפלינרית במדעי המחשב.

- אמנות או נמות
  - מוסיקה ומדעי המחשב.
  - איתור חומרי הוראה מסרטונים, מסרטים ומתמונות במוזיאונים.
- כיצד ניתן להביא את עולמם של התלמידים אל עולם מדעי המחשב?
  - מה עושים התלמידים לאחר שעות הלימודים?
  - הבדל בין גילאים : מרחק שנות דור.
- מדף הספרים שלי
  - חשיבות קריאת הספרים.
  - עליסה בארץ הפלאות.
  - ספרי מדע ולוגיקה.
  - שירה ופרוזה.
  - ספרי הרפתקאות.
- תפיסות תרבותיות
  - המיוחד בהוראת תלמידי ישראל.
- הוראת שפה
  - מהי שפה?
  - משמעות בלמידת יסודות, מודלים חישוביים.
  - משמעויות בלמידת מבני נתונים.
- הומור
  - מדוע אנו צוחקים?
  - כיצד מזהים הומור?
  - כיצד מזהים כוונה להצחיק?
- רעייה בשדות זרים : כיצד מלמדים במקומות אחרים?
  - מתי אנו לומדים?
  - ממי אנו לומדים?
  - מי לומד מאתנו?
- משחוק ולמידה
  - האם כל משחק הינו לימודי?
  - מתי הלמידה הינה ואינה משחק?

תמונות הסמינר המופיעות בעיתון צולמו ע"י ד"ר דן אהרוני, יריב דגן, ואיזבלה טבלין

אי-שימוש בפונקציות בקרב תלמידי מדעי-המחשב בתיכון

אחמד עומר

אוניברסיטת חיפה

בהנחיה של ד"ר עירית הדר – אוניברסיטת חיפה ופרופ' אורי לירון - הטכניון

במהלך עבודתי כמורה למדעי המחשב בתיכון, שמתי לב שהתלמידים לא משתמשים בכלל בפונקציות (מתודות) כאשר הם נדרשים לפתור איזה שהיא בעיה תכנותית, למרות שהם למדו את הנושא ושולטים בו די טובה. התלמידים פותרים בצורה פרוצדורלית, מעורפלת וארוכה ולא משתמשים בפונקציות אפילו במקומות הכי ברורים, כאשר פונקציות יכולות לתת תועלת רבה. הם משתמשים בפונקציות רק במקרה אחד: כאשר הם נדרשים לעשות את זה באופן מפורש.

פונקציות הן כלי מאוד מועיל בעבודה של כל מתכנת, וכמעט כל שפת תכנות מספקת דרכים לשימוש בפונקציות וכתיבת פונקציות חדשות. המטרה שלנו, כמורים, היא ללמד את התלמידים כיצד ליישם כישורי חשיבה מסדר גבוה, ומודעות לרמות הפשטה. לכן, אי שימוש של התלמידים בכלי כזה חזק כמו פונקציות, למרות שליטתם בנושא, הוא דבר מאוד מעניין. מטרת המחקר הנוכחי היא לבדוק תופעה זו לעומק, להבין את הסיבות והגורמים מאחוריה ולהציע דרכים להתגבר עליה.

חשיבות הפונקציות נשקפת בשלושה אספקטים עיקריים:

1. פירוק ומודולריות (decomposition and modularity): חלוקת הבעיה העיקרית לתת בעיות, דבר המאפשר התעסקות עם כל תת-בעיה באופן נפרד, בלי לדאוג לשאר הבעיות. זה מאפשר חלוקה של בעיה מורכבת לתת בעיות יותר פשוטות, מה שמוריד את מורכבות הבעיה ומאפשר טיפול ומענה יותר מוצלח. זה דומה לבניית מכונית, במקום לבנות אותה כיחידה אחת, בונים את המנוע ואת שאר חלקי המכונית באופן נפרד, ואח״כ מחברים הכול ביחד ליחידה אחת מתפקדת. בניה של תכנית באופן דומה נקרא תכנות מודולרי, והוא מועדף על תכנות פרוצדורלי בגלל היותו קל יותר ל:
   - הבנה: קל להבין מודול מסוים, שמטפל בתת בעיה, מאשר להבין תכנית אחת גדולה.
   - תכנות: לעבוד ולתכנת על מודול בנפרד.
   - בדיקה: בודקים כל מודול באופן נפרד, ורק אחר כך מתחילים לחבר ולבדוק עוד מודולים, אחד אחרי השני.
   - תחזוקה: אפשר לערוך שינויים בקוד במודול אחד, בלי לדאוג למודולים האחרים.

2. שימוש חוזר (reusability): כאשר כותבים תכנית חדשה, או באותה תכנית, אפשר להשתמש בפונקציות שכבר נכתבו כמענה לתת-בעיות קודמות בעת שנתקלים בהם עוד פעם. זה חוסך בזמן ובמשאבים ומועיל בשיפור התפוקה והאיכות בשלב פיתוח התוכנה, כאשר לא צריכים להתחיל ולכתוב הכול מאפס.

3. הפשטה (abstraction): מדד אחד של ההפשטה הוא היכולת לפשט את הדברים, להתעסק יותר עם המבנה הכללי של הבעיה ולא עם הפרטים הקטנים. הפשטה היא סוג של הכללה, לייצר רעיונות כלליים אשר לוכדים תכונות משותפות. פונקציות הן הדוגמה הכי בולטת להפשטה בתכנות.

בכדי להבין את התופעה של אי השימוש בפונקציות, ערכנו מחקר פיילוט בבית ספר בו אני מלמד. המחקר נערך על תלמידי י״ב הלומדים במגמת הנדסת תוכנה. הם ניגשו בשנה הקודמת לבגרות 5 יחידות במדעי המחשב וסיימו ללמוד ולתרגל את כל הנושאים בתוכנית הלימודים המיועדת ל- 5 יחידות מדעי המחשב, כולל פונקציות ונושאים הרבה יותר מורכבים כמו רקורסיה, מבני נתונים, הורשה וכדומה. בפיילוט השתתפו 10 תלמידים שהיו זמינים.

בפיילוט, חילקנו שלושה דפי שאלות לתלמידים שהתיישבו כל אחד על שולחן נפרד עם חומר פתוח, למעט גישה למחשב. כאשר תלמיד מסיים דף שאלות הוא מקבל את הדף השני, בכדי לא להשפיע על האופן בו הוא פותר את הדף הנוכחי. הסברנו להם שאין ציונים ואין אף גורם מלחיץ, כולל זמן, וביקשנו שיפתרו בצורה שנראית להם הכי נכונה.

בדף השאלות הראשון היו שלוש שאלות פשוטות, מאותו סוג של בעיה אלגוריתמית, בלי אינפורמציה נוספת.

בדף השאלות השני, הבאנו עוד שלוש שאלות באותה רמה ושוב מאותו סוג של בעיה אלגוריתמית, עם הנחיה לשים לב למה שדומה בשאלות, בכדי לוודא שהתלמידים שמו לב לבעיה האלגוריתמית הזהה, במקרה והם פספסו את זה בדף הראשון, אבל בלי לכוון אותם איך להתייחס לזה.

בדף השאלות השלישי, ביקשנו מהם באופן מפורש, לכתוב פונקציה, ולהשתמש בה בפתרון השאלות כמענה לבעיה האלגוריתמית הזהה. דף שאלות זה היווה סוג של שאלת ביקורת, בכדי להראות שהתלמידים כן מכירים את הנושא של פונקציות ויודעים לפתור שאלות עם פונקציות.

דף שאלות 1

> נתונים 3 מערכים שונים בשם arr1, arr2, arr3 אשר מכילים מספרים שלמים.
> אתם מתבקשים:
> 1. להוסיף 1 לכל איבר במערך arr1.
> 2. להוסיף 3 לכל איבר במערך arr2.
> 3. להחסיר 2 מכל איבר במערך arr3 (להוסיף 2-).

דף שאלות 2

> נתונים 3 מערכים שונים בשם arr4, arr5, arr6 אשר מכילים מספרים שלמים.
> אתם מתבקשים:
> 1. להכפיל כל איבר במערך arr4 ב- 2.
> 2. להכפיל כל איבר במערך arr5 ב- 10.
> 3. להכפיל כל איבר במערך arr6 ב- 3.
> שים לב לחלק הדומה בין הסעיפים.

דף שאלות 3

> א. כתבו פעולה אשר מקבלת כפרמטר מערך של מספרים אמיתיים, ופרמטר num מסוג מספר אמיתי. הפעולה תחלק כל איבר במערך במספר num.
> ב. נתונים 3 מערכים בשם arr7, arr8, arr9.
> אתם מתבקשים:
> 1. לחלק כל איבר במערך arr7 ב- 2.
> 2. לחלק כל איבר במערך arr8 ב- 2.5.
> 3. לחלק כל איבר במערך arr7 ב- 3.
> עליכם להשתמש בפונקציה שכתבתם בסעיף א׳.

התוצאות של דפי השאלות חיזקו את העמדה שלנו, והראו שהתלמידים לא השתמשו בפונקציות אלא בעת שנדרשו לעשות את זה.

בדף השאלות הראשון, כל התלמידים, ללא יוצא מן הכלל, השתמשו בשלוש לולאות for על מנת לפתור את השאלה עם שינוי של הערך בכל פעם. וזה היה המצב גם בדף השאלות השני, למרות שהייתה הנחיה ברורה לשים לב לחזרות בין שלוש השאלות.

פתרון קלאסי עבור דף שאלות 1

```
for (int i = 0; i < arr.Length; i++)
    arr[i] += 1;

for (int i = 0; i < arr.Length; i++)
    arr[i] += 3;

for (int i = 0; i < arr.Length; i++)
    arr[i] -= 2;
```

פתרון קלאסי עבור דף שאלות 2

```
for (int i = 0; i < arr.Length; i++)
    arr[i] *= 2;

for (int i = 0; i < arr.Length; i++)
    arr[i] *= 10;

for (int i = 0; i < arr.Length; i++)
    arr[i] *= 3;
```

בדף השאלות השלישי, כצפוי, כל התלמידים כתבו פונקציה כנדרש והשתמשו בה בפתרון שלוש השאלות. (הערה: הסתכלנו על המבנה הכללי של הפתרון. היו לתלמידים טעויות קטנות שלא התייחסו אליהן.)

פתרון קלאסי עבור דף שאלות 3

```
public static void Divide (double[] a, double num)
{
    for (int i = 0; i < a.Length; i++)
        a[i] /= num;
}

Divide (arr, 2);
Divide (arr, 2.5);
Divide (arr, 3);
```

בשלב שני, במטרה להבין יותר את הסיטואציה, ערכנו ראיון קבוצתי לכל התלמידים ביחד. עבור כל דף שאלות, שאלנו את התלמידים, אם הם זיהו את הבעיה האלגוריתמית הזהה, ומדוע הם לא השתמשו בפונקציות.

עבור דף השאלות הראשון, כל התלמידים ענו שהם לא זיהו את הבעיה האלגוריתמית הזהה, ומה שחשוב זה שהם פתרו את השאלות, לא חשוב איך ולא חשוב אם זה בצורה הכי יעילה או לא.

עבור דף השאלות השלישי, כולם ענו שכתבו פונקציה כי הייתה דרישה ברורה לכתוב פונקציה ולהשתמש בה.

עבור דף השאלות השני, כולם ענו שכן שמו לב לבעיה האלגוריתמית הזהה, וכשנשאלו למה לא השתמשו בפונקציה כמענה לזה, דווקא קיבלנו תשובות מעניינות, אנחנו מצטטים למטה:

1. "עניתי לפי מה שעלה לי ישר בראש".
2. "עניתי בלי לחשוב יותר מדי, כמעט באופן אוטומטי".
3. "עניתי בדרך הכי קלה".
4. "יותר נוח בשבילי לפתור בדרך הזו".
5. "מספר השאלות היה קטן, למה להתאמץ?!"
6. "עניתי בדרך הפחות מורכבת, ישר ולעניין, בלי לסבך את העניינים".
7. "עניתי בדרך שבה יש סיכוי קטן יותר לטעות".
8. "לא הייתה דרישה לכתוב פונקציה, אם הייתה דרישה כזו, הייתי כותבת".
9. "מספר השאלות היה קטן והשאלות עצמן היו פשוטות".

במטרה להבין יותר את התגובות של התלמידים, השתמשנו בארבע תיאוריות/עקרונות מתחום הפסיכולוגיה, שעשויות לתת מענה לסיטואציות דומות בהן אנשים מתעלמים ממידע שרכשו, בין אם הם מודעים לזה או לא.

ארבעת התיאוריות הן :

- dual-process theory :

התיאוריה מסבירה איך המוח האנושי עובד בשני מצבים שונים : system 1 ו- system 2. הראשון אחראי על המשימות האוטומטיות והאינטואיטיביות (לא בהכרח משימות קלות), כמו רכיבה על אופניים. השני אחראי על המשימות המורכבות שצריכות עיבוד וניתוח.

- cognitive laziness :

התיאוריה מסבירה איך שאנשים יכולים להסתפק במצב "מספיק טוב" בידיעתם שיש מצב טוב יותר, שמצריך עוד מאמץ.

- principle of least effort :

העיקרון מסביר שאנשים, ואפילו חיות ומכונות עם אינטליגנציה, יבחרו תמיד את המסלול עם התנגדות המזערית שמצריך הכי פחות מאמץ. לכולם יש את הרצון להגיע לדברים במהירות ובקלות. יש דמיון בין עיקרון זה לתיאוריה של העצלנות הקוגניטיבית (2), אבל בעת שהתיאוריה בסעיף הקודם מתמקדת יותר בתוצר הסופי, שיש מצב "מספיק טוב", העיקרון הפסיכולוגי בסעיף זה מתמקד יותר בתהליך עצמו ובבחירת התהליך.

- comfort zone :

עיקרון שמסביר את המצב הנפשי שבו אדם מרגיש נוח, נינוח ובשליטה.

בטבלה הבאה, מיפינו את הציטוטים של התלמידים עם התיאוריה / העיקרון המתאים : (יתכן שחלק מהציטוטים חופפים)

תיאוריה / עיקרון	ציטוט
Dual processing theory	1,2
Cognitive laziness	2,5,6
Principle of least effort	3,7, 9
Comfort zone	4,8

כפי שניתן לראות בטבלה, אין תיאוריה או עיקרון אחד מובהק שיכול להיות אחראי על ההתנהגות המתוארת, וגם מספר התלמידים לא היה מספיק.

במחקר יותר מורחב, שעובדים עליו כרגע, נגדיל את מספר התלמידים במטרה לנסות לאתר את העיקרון הכי משמעותי שעומד מאחורי התנהגות זו, ולהציע דרכים להתגבר על זה, בין אם זה שיטת הוראה שונה או משהו אחר לחלוטין, תלוי בתוצאות שנקבל. בנוסף לזה, נציע סט של כללים שדרכם אפשר למפות כל התנהגות דומה, של התעלמות מהידע שיש לנו בתחום מסוים, לאחת התיאוריות / העקרונות שהזכרנו לעיל, באופן חד-חד-ערכי.

מדליות לתלמידי ישראל באולימפיאדת מדעי המחשב

פורסם ב- Ynet ב- 21.8.16 http://www.ynet.co.il/articles/0,7340,L-4844234,00.html

התלמידים שהשתתפו בתחרות בינלאומית שהתקיימה ברוסיה, זכו בסוף השבוע במדליית כסף ומדליית ארד. המשלחת הישראלית במדעי המחשב כללה ארבעה תלמידים : תומר אדר מקריית חינוך רופין בעמק חפר, שזכה במדליית הכסף, לירן מרקין מתיכון ויצ"ו נהלל שזכה במדליית הארד, נועם תא-שמע מעירוני ד' בתל אביב ורון סולן מתיכון הראשונים בהרצליה. מאמן הנבחרת הוא ד"ר דוד גינת מאוניברסיטת תל אביב. התחרות התקיימה בעיר קאזאן ברוסיה בהשתתפות 81 מדינות, ובסיומה דורגה ישראל במקום ה-28.

על אקראיות, מקריות ומזל

אודי מלכה

מכללת אורט ביאליק

האם בחירה של מספר אקראי על ידי אדם היא באמת אקראית? מהי חשיבותה של האקראיות בחיינו ומה אקראי סביבנו? כיצד המחשב "בוחר" מספר אקראי והאם הוא באמת אקראי? איך יודעים אם קבוצת מספרים היא קבוצה של מספרים אקראיים? דיון בכיתה בנושאים אלו עם הפעלות חווייתיות יכול להוות פתיח ללימוד אודות מספרים אקראיים ובסיס להבנה לקראת מימוש הידע בשפת התכנות הנלמדת. על אקראיות במחשב, בכיתה ומחוצה לה, במאמר שלפניכם.

נתחיל מהתחלה

נתחיל מהתחלה. כמה מהתחלה? מרגע הבריאה. זהו אחד הרגעים שיש לגבי השתלשלות המאורעות בו דעות שנויות במחלוקת. זהו הרגע בו החל מסע ממנו התפתחו החיים כפי שאנו מכירים אותם כיום, מסע שמרגע שנוצרו החיים הוביל במשך הזמן לפיתוח תורת האבולוציה על ידי צ'ארלס דרווין. דרווין הציע את רעיון הברירה הטבעית שעל פיה לזנים וגזעים מסוימים יש סיכוי גדול יותר לשרוד ולהתרבות כאשר יש ביניהם תחרות על משאבים מוגבלים.

השאלה הגדולה היא האם השינויים שעוברים האורגניזמים בכל דור הם אקראיים. ההעברה של גנים מסוימים ולא אחרים היא אקראית אך האופן שבו מתפתחים המינים היא באמצעות הברירה הטבעית וזו אינה אקראית כלל. כלומר, אם לאחר היווצרות החיים והעברה של גנים מסוימים לאורגניזם מסוים קיים יתרון שרידות ורבייה באזור מחיה מוגדר על פני אורגניזם אחר אזי במקרה של תחרות בין הפריטים ישרוד לרוב הראשון. עובדה זו נכונה כאשר מדובר באורגניזמים מאותו הזן וגם כאשר מדובר על תחרות בין זנים שונים. תוצאה אחרת מזו שהוצעה יכולה לקרות במצב שבו קורה שינוי פתאומי על פני כדור הארץ ומערבב את הקלפים בבחינת יוצר שינויים שיכולים להוות יתרון לזן אחד שבמצב אחר ייתכן ודינו היה להיכחד מעל פני כדור הארץ. בקרב כל מין קיים גיוון של צורה, פיזיולוגיה והתנהגות שהם תוצאה של הבדלים תורשתיים (דרויאן, 1999). שינויים גנטיים אקראיים אשר עוברים מדור לדור הם למעשה מוטציות. מינים משתנים על מנת להתאים את עצמם לדרישות סביבה חדשות עקב התפשטות אוכלוסייה לסביבה חדשה או שינוי בתנאי סביבה. השינויים מתבצעים על ידי מוטציות אשר לרוב נתפסות על ידינו בעלות קונוטציות שליליות אך לעיתים הן יכולות להוות יתרון ועל פי עקרון הברירה הטבעית הן אלו שהופכת את הזן לחזק ועמיד יותר, כלומר בעל כושר שרידות גבוה יותר. הברירה הטבעית היא תופעה מעניינת שכן ניתן לראות בה כאחראית על יצירת אקראיות אך מרגע שזו באה לעולם בצורה של חיים חדשים הם פועלים על פי חוקים דטרמיניסטיים עד לרגע שבו הם מייצרים בעצמם חיים.

בחר מספר

כדי לבחון בעצמינו את המושג אקראיות נערוך ניסוי בכיתה. נבקש מהתלמידים לבחור ב-5 מספרים אקראיים בטווח שבין 1 ל-20.

מניסויים לא רשמיים נמצא כי המספר 17 חוזר על עצמו אצל לא מעט תלמידים ורוב המספרים שהתלמידים בוחרים הם מספרים ראשוניים אף על פי שהמספרים הראשוניים בטווח שניתן לבחירה מהווים קצת פחות מ-50% מכלל המספרים. משתמע מכך שבחירה של מספר אקראי על ידי אדם אינה אקראית כלל ועיקר.

אם נעצור לרגע ונחשוב מה אקראי בחיים שלנו לא בטוח שנמצא מה לומר. האם סדר ישיבת התלמידים בכיתה הוא אקראי? בוודאי שלא. הוא נקבע על ידי

ההעדפות של התלמידים, הכיסאות שכבר נתפסו על ידי אחרים ואולי גם על ידי העברה יזומה שלנו המורים. אם שום דבר אינו אקראי בחיינו, מה בדבר הסיסמאות שאנו בוחרים עבור חשבונות שונים. מחוללי סיסמאות נוצרו בדיוק מסיבה זו שכן אנו בוחרים בסיסמאות ספציפיות שקל לזכור אותן ובעל עניין שמכיר אותנו לעומק יוכל לגלותן שכן יש להן משמעות בחיינו או לכל הפחות היגיון כלשהו.

אקראיות והסתברות

לפני שנכיר את התהליך שבו המחשב "בוחר" מספר אקראי חשוב להבין מהיכן החל חקר האקראיות וגם את הצורך והשימוש של מספרים או מצבים אקראיים בחיי היומיום.

במאה ה-17 התפתחו משחקי מזל ועימם גם תחום ההסתברות. האקראיות נועדה לספק מגוון של תוצאות אפשריות בעלי סיכויים זהים וההסתברות נועדה לחשב את הסיכויים (או הסיכונים) באופן כללי ובתנאים מסוימים. במאה ה-18 הבנה של התפלגויות בטעויות מדידת גופים שמימיים היוותה את המוטיבציה להמשיך לחקור את הנושא, ופעמון ההתפלגות הנורמאלית נחשף, בדיוק כפי שנראה על ידי המתמטיקאי אברהם דה-מואבר, אשר חקר את התפלגות התוצאות המתקבלות בסדרה של הטלות מטבע. במאה ה-19 כבר נוסח חוק המספרים הגדולים ובמאה ה-20 נוסח המרחב ההסתברותי בצורה פורמאלית.

אנו נעזרים במתמטיקה לכמת את התופעות שאנו צופים או חשים בהם ובתוך כך אנו נעזרים בנוסחאות על מנת שנוכל לצפות בדיוק רב ככל שניתן את התוצאה בהינתן משתנים הידועים לנו. יחד עם זאת, אנו טועים לא מעט בכך שאנו מפרשים אירוע אקראי בצורה לא נכונה או אומדים את הסיכויים להתרחשותו באופן לא מדויק. אירועים אשר משנים את פני ההיסטוריה כמו גילויים מדעיים, התפתחויות טכנולוגיות או מלחמות עולם הם אירועים שיש לצפות להתרחשותם גם אם בהסתברות נמוכה אך זו עדיין קיימת. אירועים אחרים בעלי הסתברות נמוכה כמו הגרלת כל מספרי הלוטו שבוע אחר שבוע הם אפשריים וקרו בעבר. כדאי לזכור שהגרלת המספרים 1,2,3,4,5,6 בלוטו היא בעלת הסתברות כמו הגרלת כל 6 מספרים אחרים. כל אחת מהתוצאות האפשריות בסיטואציה כלשהי, בין אם מגוון האפשרויות ידוע לנו מראש או לא, היא אקראית. המקריות מתייחסת לתוצאה אחת מתוך טווח האפשרויות כאשר הסיכוי לכל אחת מהן הוא זהה. כאשר המקריות תואמת את הבחירה שלנו יש לנו נטייה לקרוא לה מזל.

מספרים אקראיים אמתיים ומדומים

מחשבים מייצרים מספרים אקראיים לצורכי הצפנה, סימולציה, משחקי מחשב והימורים אך האם אנו יכולים להיות בטוחים כי מדובר במספרים אקראיים באמת. מי שמייצר את המספרים הוא קטע קוד הכולל פקודות הניתנות למחשב לביצוע. המחשב הינו מכונה. מתוחכמת ביותר אך עדיין מכונה. האם מכונה יכולה לחולל דבר מה אקראי כמו מספר?

למעשה רוב המספרים שאותם מייצר המחשב אינם אקראיים באמת אלא פסאודו אקראיים. מחולל מספרים פסאודו אקראיים (בקיצור - PRNG pseudo random number generator) הוא אלגוריתם במחשב המייצר סדרת מספרים או אלמנטים המדמים בקירוב תכונות של רצף אקראי אמיתי, כזה שנוצר באמצעות מחולל מספרים אקראיים אמיתיים (TRNG - true random number generator). אלגוריתם המספרים המדומים משתמש לעיתים קרובות במצב התחלתי (initial state) או גרעין (seed) אשר משמשים כקלט לאלגוריתם לבצע מניפולציות שונות ולהפיק פלט של מספרים הנראים אקראיים.

השיטה הראשונה ליצירת מספרים אקראיים פותחה על ידי ג'ון פון ניומן שבה בוחרים מספר שאינו אקראי. מעלים את המספר בריבוע, לוקחים ממנו את ה- 10 ספרות האמצעיות, מבצעים פעולה זו שוב ושוב עד שמקבלים את סדרת המספרים הפסאודו אקראית. הבעייתיות של מחוללי המספרים הפסאודו אקראיים היא שידיעה של המצב ההתחלתי או הגרעין עלולה להוביל לידיעה של המספרים שנבחרו ולשחזר את תהליך בחירתם כך שגורם האקראיות נאבד. בנוסף, קיים הסיכון שיצירה של מספר רב של מספרים אקראיים תגרום

למחזוריות שגם היא מבטלת למעשה את אלמנט האקראיות. בחירת המצב ההתחלתי או הגרעין ליצירת מספרים אקראיים מדומים יכולה להתבצע בשיטות שונות כמו בחירה בקצב הקלדה של המשתמש או תנועות העכבר שלו.

מחוללי מספרים פסבדו אקראיים מסייעים בפיתוח סימולציות בעיקר באמצעות השימוש בקבוצת אלגוריתמים מסוג מונטה קרלו. אלגוריתמים חישוביים המשויכים לקבוצת מונטה קרלו מגרילים מספר רב של פעמים מספרים אקראיים ומבצעים עליהם חישובים. ביצוע סימולציות מספר רב של פעמים עוזר בפתרון בעיות שקיים קושי במציאת פתרון מתמטי מדויק עבורן. האפשרות לשחזר את הסימולציה הינה בזכות השימוש בגרעין זהה לזה ששימש בניסוי המקורי. אלגוריתמים הסתברותיים מהסוג של מונטה קרלו נחשבים למהירים אך המחיר של המהירות גובה את הסיכון לקבלת טעות כלשהי, כלומר אופן הפעולה של אלגוריתמים אלו הינו סטטיסטי ולא חישוב מדויק ועל כן התשובה המתקבלת עלולה להיות שגויה. רמת הדיוק נקבעת בעיקר על פי מספר הדגימות האקראיות והבדיקה של עמידתן או אי עמידתן בתנאי הנבדק (1964, Handscomb & Hammersley).

אלגוריתמים "בטוחים" יותר אשר מספקים תוצאה מדויקת אך מתבצעים לאט יותר שייכים למשפחת אלגוריתמים שנקראת לאס-וגאס. אלו אלגוריתמים מדויקים יותר שכן הפסקת הפעולה שלהם מתבצעת כאשר נמצא פתרון מוחלט לבעיה ושכן האלגוריתם אינו מסתפק במציאת פתרון בהסתברות שאינה וודאית. מרכיב האקראיות באלגוריתם הוא רק חלק באסטרטגית הריצה שלו בניסיון ליעל את זמן הריצה. שני האלגוריתמים, מונטה קרלו ולאס וגאס, נחשבים לאלגוריתמים אקראיים שכן חלק מפעולותיהם מתבצעים על יסוד בחירה או הכרעה אקראית.

מחולל המספרים הפסאודו אקראיים של חברת מיקרוסופט הוא אחד השימושיים ביותר כיום. בשנת 2007 פרסמו חוקרים מאוניברסיטת חיפה מאמר ובו טענו כי ברגע שמוצאים את המצב הפנימי הנוכחי של המחולל במערכת הפעלה "חלונות 2000" ניתן לדעת את מצבו בעת יצירת המספר הקודם בסיבוכיות שאינה מהווה קושי לבעלי עניין לנסות ולגלות את המספר האקראי. כמו כן מצאו החוקרים כי יצירת המספרים מתבצעת על ידי הקוד ברמת משתמש (user mode) ולא של מערכת (kernel mode) ועל כן אין צורך בהרשאות מנהל מערכת על מנת לגשת למחולל (2007, Dorrendorf, & Pinkas Gutterman).

לחברת אינטל קיים שבב בשם rdRand המחולל מספרים אקראיים אשר נועד לייצר מספרים אקראיים עבור כל תוכנה הנדרשת למספר כזה. נטען כנגדו כי הוא אינו בטוח ביצירת מספרים אקראיים ובשנת 2013 מערכת ההפעלה FreeBSD הפסיקה להשתמש בו כמחולל מספרים אקראיים. כמובן שחברות מיקרוסופט ואינטל אינן היחידות לייצר מחוללים או שבבים החשודים כמייצרים מספרים שאינם מספיק אקראיים. הסיפור שלהן מהווה דוגמא מייצגת לקופסא שחורה שיש לה חשיבות רבה אך היעדר מידע עליה והאופן שבו היא פועלת מהווה סיכון שיכולות להיות לו השלכות משמעותיות. לשפות תכנות רבות קיימים מחוללי מספרים פסאודו אקראיים אך הבעיה היא שלא תמיד ניתן לבחון בתיעוד הקיים של השפה מהו האלגוריתם שבו נעשה שימוש ליצירת המספרים האקראיים וכן לא תמיד ניתן לבדוק את אופן המימוש שלו.

לצרכים של הגרלה לצורך הימורים או להגרלת אירועים במשחק מחשב ניתן להסתפק במספרים פסאודו אקראיים אך כאשר מדובר בהצפנה החולשה של מחולל פסאודו אקראי יכולה להוות בעיה. הצפנה שאינה בטוחה עלולה לסכן עסקאות של קניה באינטרנט ושל העברת כספים מחשבונות בנקים ועל כן האבטחה שלה היא קריטית. מפתחות הצפנה המקובלים ליצירת קשר מאובטח ברשת האינטרנט מורכבים משני מספרים ראשוניים גדולים. מספר הספרות של מפתחות כיום הוא כ-200 ספרות ויותר והוא תלוי במידת החוזק המבוקש של ההצפנה. השימוש במספרים ראשוניים נובע מהעובדה כי קיים קושי לפרק מכפלה של שני מספרים ראשונים ולמצוא אותם (בעיית הפירוק) וכן העובדה כי המרווחים בין מספרים ראשוניים הם אקראיים (אם כי קיימים מחקרים המזהים דפוסים

דומים לפיזור המספרים הראשוניים בטווחים מסוימים הידועים לנו (Stanley, &Kumar, Ivanov 2003)). המפתחות משמשים להעברה של מידע לצד השני מבלי שגורם זר יוכל לקרוא אותו וכן משמשים למנוע התחזות לשולח המסר ולמנוע התכחשות השולח לשליחת המידע ולתוכו (נקרא גם חתימה דיגיטאלית). מסיבה זו האקראיות של המפתחות היא קריטית.

לבחירת מספריים אקראיים קיימים, כפי שצוין קודם לכן, גם מחוללי מספרים אקראיים אמתיים אך אלו יקרים ומורכבים. כדי שהמחשב ייצר מספר אקראי אמתי הוא נדרש לקבל את הקלט מאלמנט שאינו ניתן לשחזור מתוך הסביבה החיצונית כמו למשל, רעש אטמוספרי, קרינה רדיואקטיבית, רעשים תרמים או אפילו באמצעות שימוש במנורת לבה על ידי צילום התנועה האקראית של בועות שעווה צמיגות במכל זכוכית ויצירת מספרים אקראיים מהתמונות.

מבחני אקראיות

כדי לדעת האם קבוצה של מספרים מהווים קבוצה של מספרים אקראיים קיימים מבחנים שונים לבדיקת אקראיות. אף על פי שמבחינה מתמטית לא ניתן להוכיח שרצף הנוצר ממקור אקראי כלשהו הוא אכן אקראי. מבחנים שנעשים על מחוללי מספרים אקראיים אמתיים מציגים לעיתים תוצאות המראות הטיות סטטיסטיות הנגרמות כתוצאה מהשפעת הסביבה על התופעה הפיזיקלית שהיא המקור למחולל של המספר. יחד עם זאת, מבחני אקראיות מספקים וודאות ברמות סטטיסטיות שונות ויכולים לאתר נקודות חולשה במחולל המספרים האקראיים אשר גורמות לו להפיק מספרים שאינם אקראיים. או במילים אחרות, ניתן לצפות אותם מראש או לכל הפחות לשחזור את תהליך יצירתם.

החשיבות של אקראיות מספרים מביאה עמה את הצורך בפיתוח בדיקות תקניות לאקראיות מספרים הנוצרים בידי מחוללים שונים ואלו מפותחות על ידי חברות פרטיות והן על ידי המכון הבין לאומי לסטנדרטים וטכנולוגיה (Soto ,1999).

חלק מהמבחנים הסטטיסטים כוללים בדיקות אודות תדירות הסיביות 1 ו-0, בדיקת רצפים של סיביות זהות, בדיקה שבה מחלקים את הרצף לתתי סדרות ובודקים את מידת חזרתן (מבחן פוקר) וכן בדיקות רצפים של סיביות 0 ו-1 והתאמתן לרצף המצופה בעת יצירת מספר אקראי. בדיקת מידת היכולת לדחוס רצף באמצעות אלגוריתמי דחיסה שונים מהווה אף היא בדיקה למידת אקראיות של מספרים (Morgan ,1984). מעבר בהצלחה של מבחנים אלו מגדיר מספר בהסתברות גבוהה את היותו אקראי אך עדיין אינו מהווה הוכחה שהוא אכן אקראי.

קצת קוד

אחת השיטות הנפוצות להפקת מספרים פסבדו אקראיים היא אלגוריתם linear congruential generator (בקיצור LCG) שמשתמש בנוסחת הנסיגה המוצגת למטה:

$$x_{n+1} = (aX_n + c) \bmod m$$

מחולל המשתמש בנוסחה זו מייצר רצף מספרים פסבדו אקראיים שגודלם אינו עולה על גודלו של m ומחזוריותו אינה גבוהה מ-2^m. עקב היותו פשוט הוא נמצא בשימוש נרחב. בשפת C הפונקציה rand מממשת אלגוריתם מסוג LCG. להלן תוכנית בשפת C המגרילה מספר בין 1 ל-6:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void main()
{
        srand(unsigned(time(NULL)));
        intnum=1+(int)(rand()%6);
        printf("%d",num);
}
```

הפונקציה srand מקבלת את הגרעין (seed) אשר בעזרתו יופקו המספרים האקראיים. הגרעין הנבחר הוא מתוך המנגנון של השעון. כמובן שבמקרה זה מדובר במחולל פסאודו אקראי שהרי ידיעה של השעה המדויקת שבה הופעלה הפקודה תוכל לסייע לשחזור יצירת המספרים האקראיים ועל כן מדובר במספרים שאינם אקראיים באמת. שינוי שעת המחשב בעת הגרלת המספר האקראי לא תעזור לנו להפוך את המספר לאקראי אמתי שכן הפרש השעות או השעה אותה נקבע במחשב לצורך הבחירה היא בעצמה אינה אקראית.

סיכום

ייתכן והתבוננות על אקראיות בטבע ובחיי היומיום תעצב לנו תפיסה חדשה לגבי אקראיות. ייתכן ופעולות שנתפסו בעינינו כאקראיות יראו לנו מעתה כמתוכננות היטב ואירועים שמוצגים לנו כאקראיים יראו לנו תלויי הסתברות הניתנת במקרים רבים לחישוב.

הדיון בכיתה אודות נושא האקראיות עוזר להבהיר לתלמידים מה קורה מאחורי הקלעים ברגע שהם כותבים קוד בשפת C ומשתמשים בפונקציות לסנכרון עם השעון ומחוללים מספר אקראי. אך יותר מכך - הדיון מכניס אותם אל תוך עולמם של מתכנני הפונקציות ומציג להם את האתגרים, השיקולים והאפשרויות הקיימות כאשר מתכננים מחולל מספרים אקראיים.

רשימת מקורות

דרויאן, ש' (1999). **עקרונות אבולוציוניים בהתפתחות החשיבה**, רמות, אוניברסיטת ת"א.

Dorrendorf, L., Gutterman, Z., & Pinkas, B. (2007). Cryptanalysis of the Random Number Generator of the Windows Operating System.

Hammersley, John M.; and Handscomb, David C. 1964 *Monte Carlo Methods*. New York: Wiley.

Kumar, P., Ivanov, P. C& ,.Stanley, H. E. (2003).Information entropy and correlations in prime numbers.*arXiv preprint cond-mat/0303110*.

Morgan, B. J. (1984). Testing Random Numbers. *Elements of Simulation*) pp. 137-159). Springer US.

Soto, J. (1999, October). Statistical testing of random number generators. *Proceedings of the 22nd National Information Systems Security Conference*. Vol. 10, No. 99, p. 12. Gaithersburg, MD: NIST.

תמונות הסמינר המופיעות בעיתון צולמו ע"י ד"ר דן אהרוני, יריב דגן, ואיזבלה טבלין

תחרויות תכנות

קיימות ברשת תחרויות תכנות רבות. אנחנו מביאים לכם מספר דוגמאות של שאלות מסוגים שונים וברמות שונות של קושי. אפשר להשתמש בשאלות האלה בכיתה או לעודד את התלמידים להשתתף בתחרויות השונות.

הדפס יהלום

מתוך האולימפיאדה של אירלנד, תחרות 2012 (http://aipo.computing.dcu.ie/2012-programming-problems)

תחרות Irish Schools' Programming Competition היא תחרות שנתית שנערכת בדבלין. זו התחרות שמתוכה בוחרים את הנבחרת של אירלנד לאולימפיאדה. פרטים נוספים באתר /http://aipo.computing.dcu.ie

כתוב תכנית בשם Diamond.

התכנית תקלוט מספר שלם חיובי אי זוגי $3<n<19$.

התכנית תדפיס צורת יהלום , כאשר n קובע את הרוחב של היהלום (מספר ה- '*' בקו המרכזי של היהלום)

למשל, אם n=3

```
  *
 * *
* * *
 * *
  *
```

ואם n=9

```
        *
       * *
      * * *
     * * * *
    * * * * *
   * * * * * *
  * * * * * * *
 * * * * * * * *
* * * * * * * * *
 * * * * * * * *
  * * * * * * *
   * * * * * *
    * * * * *
     * * * *
      * * *
       * *
        *
```

המספר הגדול ביותר

מתוך תחרות 2016 של IPSC (/https://ipsc.ksp.sk)

תחרות IPSC (Internet Problem Solving Contest) היא תחרות שמקורה בסלובניה. זו תחרות ברשת האינטרנט בין קבוצות של 1-3 אנשים. ניתן להשתמש בכל שפת תכנות כדי לענות. פרטים נוספים באתר /https://ipsc.ksp.sk

פיטר אוהב לשחק במגרש הגרוטאות. הוא מצא במגרש מחשב מדעי והביא אותו הביתה. הוא ניסה לתקן אותו, אבל המחשב עדיין מקולקל.

הנה מה שהמחשב יכול לעשות:

- יש מספר ביטויים אריתמטיים המאוחסנים בזכרון של המחשב. פיטר יכול לטעון כל אחד מהם למסך.
- מקשי החיצים ומקש ה- delete עדיין עובדים. על ידי שימוש בהם, יכול פיטר למחוק כל תת מחרוזת של תווים מתוך הביטוי האריתמטי שמוצג על המסך.
- מקש השוויון = עדיין עובד. פיטר יכול להשתמש בו כדי לחשב את הביטוי האריתמטי שמופיע על המסך.

פיטר אוהב מספרים גדולים. לכל אחד מהביטויים בקלט, מצא את המספר הגדול ביותר שניתן לייצר מהביטוי האריתמטי שמופיע על המסך (על ידי לחיצה על מקש השוויון =).

נתונה מחרוזת S שמכילה ביטוי אריתמטי חוקי. הסר ממנה את כל התווים כך שהמחרוזת T שתישאר תהווה ביטוי אריתמטי חוקי ושהערך של T יהיה הגדול ביותר שאפשר.

ביטויים חוקיים מוגדרים כך :

- כל מספר שלם לא שלילי הוא חוקי כל עוד אין לפניו אפסים מובילים מיותרים. (למשל 4-007 אינו ביטוי חוקי)
- אם a הוא ביטוי חוקי, אז (a) הוא גם כן ביטוי חוקי.
- אם a הוא ביטוי חוקי, אז a- ו- a+ הם גם כן ביטויים חוקיים.
- אם a ו- b הם ביטויים חוקיים, אז a-b , a+b ו- a*b הם גם כן ביטויים חוקיים.
- לא קיים ביטוי חוקי מעבר למה שהגדרנו. (למשל, אין סימן חילוק)

במהלך החישוב המחשב פועל כך :

- סוגריים מחושבים קודם כל
- אחר כך, מחושבים + - האונאריים
- לאחר מכן, מחושב *
- לבסוף מחושבים + (חיבור) ו - (חיסור) משמאל לימין

שים לב שיכולים להופיע מספר פלוסים ומינוסים. למשל 1*(2+3*-+-4) הוא ביטוי חוקי שמחושב כך

- 4-+- הוא 4,
- 3*4 הוא 12,
- 2+12 הוא 14,
- ולבסוף 14*1 הוא 14.

קלט : השורה הראשונה בקובץ הקלט מכילה מספר שלם t הקובע את מספר מקרי המבחן (test case).

אחרי כל מקרה מבחן יש שורת רווח. כל מקרה מבחן כולל שורה אחת ובה ביטוי חוקי אחד S.

במקרה הפשוט של הבעיה, t=100 והאורך של כל ביטוי הוא 1-16 תווים.

במקרה המורכב יותר של הבעיה, t=1000 והאורך של כל ביטוי הוא 1-1000 תווים.

פלט : לכל מקרה מבחן, הדפס שורה אחת עם ביטוי חוקי T, כך ש T הוא תת מחרוזת של S ויש ל T את הערך המקסימלי.

אם יש מספר פתרונות, בחר אחד מהם.

דוגמה

קלט

```
3
-10
0*0*1
0+(1*2)
```

פלט

```
10
1
0+(12)
```

מניעת תאונות

מתוך תחרות 2016 של IPSC (https://ipsc.ksp.sk/)

יש לך אוסף של מילים בנות 4 תווים על השולחן. אבל חברים עם תינוק צריכים לבוא אליך בשבוע הבא ואם התינוק ינסה לאכול את המילים, הוא ייחנק. כדי למנוע את התאונה הזו, אתה רוצה להסתיר את המילים שלך בתוך מחרוזת ארוכה יחידה (ארוכה יותר מהפה של התינוק).

נתונים : בדיוק 10 מחרוזות. כל אחת מהן מכילה בדיוק 4 תווים.

עליך ליצור מחרוזת חדשה מאורך n שמכילה את כל 10 המילים הנתונות כתת מחרוזות. כל תת מחרוזת חייבת להיות רציפה. לדוגמה, ABXCD <u>לא</u> מכילה את ABCD כתת מחרוזת כיון ש- X לא שייך למחרוזת המקורית. ההופעה של 10 המילים הנתונות יכולה להיות בכל סדר ויתכן שיש ביניהן חפיפה.

לבעיה זו יש שתי תת בעיות בלתי תלויות :

הבעיה הפשוטה יותר : עליך ליצור מחרוזת באורך n=42. הבעיה המורכבת יותר : עליך ליצור מחרוזת באורך n=39

קלט : השורה הראשונה בקובץ הקלט מכילה מספר שלם t הקובע את מספר מקרי המבחן (test case).

אחרי כל מקרה מבחן יש שורת רווח.

כל מקרה מבחן כולל בדיוק 10 מחרוזות מופרדות על ידי רווח. כל מחרוזת כוללת בדיוק 4 אותיות באנגלית UPPERCASE (A-Z).

פלט: לכל מקרה מבחן, הדפס בשורה יחידה את המחרוזת באותיות UPPERCASE. כל אחת מ- 10 המילים הנתונות חייבות להופיע במחרוזת שלך כתת מחרוזת לפחות פעם אחת.

הנתונים מבטיחים שקיים פתרון. אם קיימים מספר פתרונות, ניתן לבחור רק אחד מהם.

דוגמה

קלט

```
2
TEST INTE RNET PROB ROBL OBLE
BLEM SOLV VING TEST
AAAA  AAAA  AAAA  AAAA  AAAA
AAAA AAAA  AAAA  AAAA  AAAA
```

אם n=29, אז זה אחד מהפלטים הנכונים

```
INTERNETPROBLEMSOLVINGCONTEST
AAAAAAAAAAAAAAAAAAAHELLOWORLD
```

במקרה הפשוט עלינו לייצר מחרוזת חדשה שארוכה יותר מ- 10 המילים ביחד. לכן, אפשר לשרשר את 10 המילים ולהוסיף שני תווים (כלשהם) לסוף.

המלבנים הקסומים

מתוך תחרות 2016 של IPSC (https://ipsc.ksp.sk/)

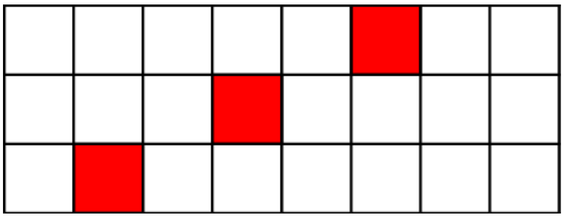
במשחק הזה, אליס מקבלת מספר שלם s וטבלה ריקה עם r=3 שורות ו- c עמודות. אליס צריכה למלא את התאים בטבלה עם מספרים לא שליליים.

שלושה תאים נקראים triplet שלשה אם הם נמצאים בשורות שונות והמרכזים שלהם יוצרים שורה ישרה. מטרת המשחק היא למלא את הטבלה כך שלכל שלשה יהיה אותו סכום.

נתונים:

מספר העמודות c והסכום הרצוי של השלשה s.

חשב את מספר הדרכים למלא את הטבלה באופן הרצוי. הואיל והמספר יכול להיות גדול מאד, חשב אותו modulo 10^9+9



לדוגמה, אחת השלשות הרצויות בלוח עם 8 עמודות

קלט: השורה הראשונה בקובץ הקלט מכילה מספר שלם t הקובע את מספר מקרי המבחן (test case).

אחרי כל מקרה מבחן יש שורת רווח.

כל מקרה מבחן כולל שורה אחת ובה השלמים c ו- s המופרדים ביניהם ברווח.

המקרה הפשוט יותר

$1 \le c \le 50\ ,\ 0 \le s \le 50$

המקרה המורכב יותר

$1 \le c \le 1000\ ,\ 0 \le s \le 10^9$

פלט

לכל מקרה מבחן, הדפס מספר שלם אחד בשורה נפרדת – מספר הפתרונות מודולו 10^9+9

דוגמה

קלט

```
2

3 1

4 3
```

פלט

```
5

34
```

במקרה הראשון של הדוגמה, יש 5 שלשות: כל עמודה ושני האלכסונים ברבוע 3x3. הסכום של כל שלשה חייב להיות 1. כלומר, כל שלשה מורכבת משני אפסים ו- 1 אחד. חמשת הפתרונות הם:

```
1 1 1   0 0 0   0 0 0   1 0 1   0 1 0
0 0 0   1 1 1   0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   1 1 1   0 1 0   1 0 1
```

במקרה המבחן השני, אחד מ- 34 הפתרונות הנכונים הוא מלבן של אחדים בגודל 3x4.

זמן הגעה

מתוך תחרות 2016 של קנדה (Junior Division)
http://cemc.math.uwaterloo.ca/contests/computing.html

דינה נוסעת כל יום לעבודה. אם אין עומס בכביש, זה לוקח 2 שעות.

שעות העומס בכביש הן מהשעה 7:00 עד 10:00 בבוקר ובין השעה 15:00 עד 19:00 בערב. אם זוהי שעת עומס אזי המהירות של דינה יורדת בחצי.

דינה יוצאת לדרך בשעה עגולה (XX:00) או 20 דקות אחרי השעה (XX:20) או 40 דקות אחרי השעה (XX:40). בהינתן זמן היציאה שלה, מתי היא תגיע לעבודה?

הקלט הוא ביטוי מהסוג HH:MM כאשר HH היא שעת ההתחלה (00, 01, …, 23) ו- MM הוא אחד משלושת זמני היציאה (00, 20, 40)

חשב את זמן ההגעה של דינה HH:MM

למשל, אם זמן היציאה הוא 05:00 אזי זמן ההגעה הוא 07:00 (אין בכלל עומס).

אם זמן היציאה הוא 07:00 אזי זמן ההגעה הוא 10:30 כיון שהיא נוהגת 3 שעות בזמן העומס אבל ב- 30 דקות האחרונות אין עומס בכביש (העומס מסתיים ב- 10:00).

אם זמן היציאה הוא 23:20 אזי היא תגיע רק ב- 01:20 כיון שדינה יוצאת בשעה שאין עומס והנסיעה לוקחת שעתיים. (היא תגיע ב- 1:20 של יום המחרת)

משחק Solitaire

מתוך תחרות 2015 של IPSC (http://ipsc.ksp.sk/)

למגי יש חפיסה מיוחדת של קלפים. יש בה 4n קלפים: לכל 1<i<n יש 4 קלפים עם הערך i (עלה, לב, יהלום, ותלתן). מגי משתמשת בחפיסה לשחק משחק סוליטר פשוט.

בתחילת המשחק החפיסה מונחת עם הפנים למטה. יש 4 מקומות ריקים, אחד לכל חפיסה של קלפים (עלה, לב, יהלום ותלתן). בנוסף, יש ליד החפיסה ערימה ריקה של קלפים שנזרקו.

מטרת המשחק היא לשים את כל הקלפים במקומם. כלומר, כל קלף מתחת לקלף העוקב לו מאותה צורה.

להלן ההגדרה של צעדי המשחק. מגי חוזרת שוב ושוב על הצעדים הבאים:

1. הרם את הקלף הראשון בחפיסה.
2. בדוק האם אפשר לשים את הקלף בחפיסה המתאימה לו.
3. אם הקלף הוא "אס" (הערך שלו הוא 1), שים אותו במקום הריק המיועד לצורה (עלה, לב, יהלום או תלתן).
4. אם הערך של הקלף v גדול מ- 1, ולקלף העליון בחפיסה המתאימה בצורה יש ערך v-1, שים את הקלף v על הקלף v-1. לדוגמה, עלה 8 מעל עלה 7.
5. אם לא יכולת לשים את הקלף בחפיסה המתאימה, אזי שים אותו בערימה של הקלפים שנזרקו. לדוגמה, אם הקלף שהוצאת כרגע הוא עלה 8 ובחפיסה של המתאימה לעלה יש קלף שערכו 3, אז הקלף עלה 8 נזרק לערימת הקלפים שנזרקו.
6. אם אין יותר קלפים בחפיסה:

   אם אין יותר קלפים בערימת הקלפים שנזרקו (כלומר כל 4i הקלפים הונחו במקומם), המשחק הסתיים.

   אחרת, קח את הערימה של הקלפים שנזרקו והפוך אותה (באותו סדר) כך שתתקבל חפיסה חדשה.

מגי תנצח את המשחק תמיד אבל לפעמים המשחק יכול להיות מאד ממושך.

נתונים n והסדר ההתחלתי של הקלפים בחפיסה. חשב כמה פעמים לקח למגי לעבור על החפיסה שלה כדי לנצח את המשחק. במילים אחרות, עליך לחשב את מספר הפעמים שהיא הפכה את ערימת הקלפים שנזרקו +1.

קלט: השורה הראשונה בקובץ הקלט מכילה מספר שלם t הקובע את מספר מקרי המבחן (test case).
אחרי כל מקרה מבחן יש שורת רווח.
כל מקרה מבחן כולל שתי שורות: בשורה אחת המספר n ובשורה השנייה רשימה של 4n הקלפים השונים (החפיסה המקורית של המשחק).
כל קלף הוא מהצורה Xy כאשר X היא אחת מארבע הצורות (עלה, לב, יהלום או תלתן) ו- y הוא ערך הקלף. למשל, H4 הוא הקלף 4 לב (hearts).

במקרה הפשוט של הבעיה, t=20 ו- n=13 בכל אחד ממקרי המבחן. (מגי משחקת עם חפיסה סטנדרטית של 52 קלפים).

במקרה המורכב יותר של הבעיה, t=12 ו- 1≤n≤150,000 בכל אחד ממקרי המבחן.

לכל מקרה מבחן, הדפס שורה יחידה ובה מספר הפעמים שלקח למגי לעבור על החפיסה שלה כדי לנצח את המשחק.

לדוגמה, אם הקלט הוא

```
1

3
S2 H3 S1 D3 D1 H1 D2 C1 S3 H2 C2 C3
```

אזי הפלט הוא 2.

אחרי שעברה על הקלפים פעם אחת יהיה המצב:

- בחפיסת העלה, הקלף שערכו 1
- בחפיסת הלב, הקלף שערכו 2 (ומתחתיו הקלף שערכו 1)
- בחפיסת היהלום, הקלף שערכו 2 (ומתחתיו הקלף שערכו 1)
- בחפיסת התלתן, הקלף שערכו 3 (ומתחתיו הקלפים שערכם 1 , 2)
- בערימת הקלפים הזרוקים יהיו S2 H3 D3 S3

מגי תהפוך את הערימה (באותו סדר) ותקבל חפיסה חדשה S2 H3 D3 S3. במעבר השני על הקלפים תצליח מגי להניח את כולם ולנצח את המשחק.

בעיית הסכום a+b

מתוך תחרות 2015 של IPSC (http://ipsc.ksp.sk/)

נתונה מחרוזת של ספרות. עליך לארגן את הספרות מחדש כך שתקבל שני מספרים שלמים וחיוביים a ו- b כך שהסכום a+b יהיה המקסימלי שאפשר.

כל מספר חייב לכלול לפחות ספרה אחת. מותר להשתמש בכל ספרה רק אותו מספר פעמים כפי שהיא מופיעה בקלט.

קלט: השורה הראשונה בקובץ הקלט מכילה מספר שלם t הקובע את מספר מקרי המבחן (test case).

אחרי כל מקרה מבחן יש שורת רווח.

כל מקרה מבחן כולל שורה אחת ובה מחרוזת אחת של ספרות. אם יש במחרוזת יותר משתי ספרות אז לא כולם אפסים.

במקרה הפשוט יותר של הבעיה t=900 ולכל מקרה מבחן יש בדיוק 3 ספרות.

במקרה המורכב יותר של הבעיה t=1000 ומספר הספרות בכל מקרה מבחן הוא בין 2 ל- 16.

לכל מקרה מבחן, הדפס שורה יחידה ובה הסכום הגדול ביותר שניתן להשיג.

לדוגמה, אם הקלט הוא

```
4

001

175

21

444444
```

אז הפלט יהיה

```
10
76
3
44448
```

הסידור האופטימלי שניתן להשיג:

במחרוזת הראשונה (001) הסכום הגדול ביותר שניתן להשיג הוא 10+0 = 10

במחרוזת השנייה (175) הסכום הגדול ביותר שניתן להשיג הוא 71+5 = 76

במחרוזת השלישית (21) הסכום הגדול ביותר שניתן להשיג הוא 1+2 = 3

במחרוזת הרביעית (44444) הסכום הגדול ביותר שניתן להשיג הוא 44444+4 = 44448

מילים מסודרות היטב

מתוך תחרות 2015 של דרום אפריקה

http://www.olympiad.org.za/

מילה מסודרת היטב היא מילה שכל האותיות שלה מסודרות בסדר אלפביתי.

למשל **act** היא מילה מסודרת היטב, אבל **cat** היא לא מילה מסודרת היטב (כי a מופיעה במילה אחרי c ולפי סדר האלפבית היא צריכה לבוא קודם). או בעברית למשל, "אגדה" היא מילה מסודרת היטב בעברית ואילו "מחר" אינה מילה מסודרת היטב.

המשימה: כתוב תכנית שתקלוט מילה כלשהי ותחזיר True אם הקלט הוא מילה מסודרת היטב ו- False אחרת. לדוגמה, אם הקלט הוא המילה will אזי התכנית תחזיר False, ואם הקלט הוא המילה all אז התכנית תחזיר True.

משחק וידיאו מחניים

מתוך תחרות 2015 של דרום אפריקה (הסיבוב הראשון)

http://www.olympiad.org.za/

עומר סיים לתכנת את משחק הוידיאו שלו, מחניים, והוא זקוק לעזרתכם כדי שתבדקו אם הרמות במשחק לא קשות מידי.

המשחק מחניים מתרחש במגרש שרוחבו N בלוקים ושחקנית בשם Aphiwe שרוחבה K בלוקים.

בכל רגע קורים שני דברים: ראשית, הכדור נזרק למקום שנקבע מראש במגרש ושנית, בניסיון להתחמק מהכדור, Aphiwe זזה בלוק אחד שמאלה או בלוק אחד ימינה או נשארת במקום.

המטרה של Aphiwe היא להתחמק מהכדורים זמן ארוך ככל שהיא יכולה. היא יודעת את כל הסדרה שהכדורים יזרקו ולכן היא יכולה לעשות מהלכים כבר בהתחלה שימנעו ממנה פגיעה בכדור במהלכים מאוחרים יותר.

המיקומים (שנקבעו מראש) הם מספרים מ- 1 בצד שמאל של המגרש עד N בצד ימין של המגרש.

בראשית המשחק Aphiwe זזה עד K בלוקים של המגרש השמאליים ביותר, ממוספרים מ- 1 עד K.

כל רמה נמשכת L שניות.

המשימה:

נתונים N, K, L והמיקומים שהכדור ייזרק מהם בכל שנייה.

כתוב תכנית שתבקש את הנתונים ואז תחשב ותדפיס את הזמן הארוך ביותר (בשניות) ש- Aphiwe מצליחה ולא נפגעת מכדור.

קלט:

השורה הראשונה בקלט תהיה N, K, L (מספרים שלמים).

השורה השנייה בקלט תכיל L מספרים שלמים מופרדים זה מזה על ידי רווח. כל איבר במיקום ה- i מצביע על המיקום ממנו נזרק הכדור בשנייה ה- i.

פלט: מספר שלם יחיד, מספר השניות המקסימלי ש- Aphiwe יכולה לשרוד לפני שהיא נפגעת מכדור. אם היא יכולה לשרוד כל הרמה (כל השניות), אז התכנית תדפיס Complete.

דוגמה:

אם הקלט הוא

4 2 2
4 2

הפלט יהיה Complete.

אם הקלט הוא

5 3 6
4 5 2 1 3 2

אז הפלט יהיה 5.

הסבר:

בהתחלה Aphiwe עומדת בצד השמאלי של המגרש, תופסת את בלוקים 1, 2, 3.

הכדור הראשון נזרק לבלוק 4 והכדור השני לבלוק 5.

כדי להתחמק משני הכדורים הראשונים Aphiwe לא צריכה לזוז בכלל. אבל אז היא תפגע מהכדור השלישי (שייזרק לבלוק 2) ותישרד במשחק רק 2 דקות.

הואיל והיא יודעת את סדרת הזריקות עוד לפני המשחק, אסטרטגיה יותר חכמה תהיה לזוז בלוק אחד ימינה כאשר נזרק הכדור הראשון וכך גם בזריקת הכדור השני (לזוז ימינה בלוק אחד). לכן, כעת היא תופסת את בלוקים 3, 4, 5 והכדור השלישי שנזרק לבלוק 2 לא פוגע בה. עליה להישאר במקום.

הכדור הרביעי נזרק לבלוק 1 ולכן היא יכולה להישאר במקומה. הכדור החמישי נזרק לבלוק 3 והיא לא יכולה להתחמק ממנו בשום דרך.

כך שעם האסטרטגיה הזו, הצליחה Aphiwe לשרוד 5 שניות לפני שהיא נפגעה מכדור.

בדוק את תכניתך עם הקלטים הבאים:

(a) 10 4 8
6 1 2 4 8 6 4 2

(b) 20 9 16
16 1 20 1 2 4 16 4 8 1 2 4 8 16 8 16

(c) 40 5 36
40 30 20 10 1 2 3 4 6 5 7 9 17 10 16 11 1 40 14 13 16 11 18 9 20 7 15 23 19 26 5 8 15 30 12 33

סדרת DNA

מתוך האולימפיאדה של אירלנד, תחרות 2012
(http://aipo.computing.dcu.ie/2012-programming-problems)

כתוב תכנית שתקרא סדרת dna. התכנית תקלוט מחרוזת, למשל GTTCAG. לאחר מכן, התכנית תחפש את הסדרה הזו בתוך מחרוזת בשם Sting dnaDatabase.

התכנית תדפיס את התו (מתוך String) שמופיע בדיוק לפני המחרוזת שנקראה ואת התו שמופיע בדיוק אחרי המחרוזות שנקראה.

למשל, עבור הקלט GTTCAG, והמחרוזת dnaDatabase כפי שהיא מופיעה בהמשך, הפלט יהיה

G A
C C
T G
A T

String dnaDatabase =
"CCTGTATTAG**GTTCAG**AATTC**GTTCAG**CAGCAGATTCGATTAGCTACAACATTCAATA"+
"AAATAGCTTCGCGCTAAATTTTTAACTTCTCTGTCGTCGCACAATCGACTTCTCTGTT"+
"TTCTT**GTTCAG**GGGTTTACCGGAATTGTTTCTGCTGCGATGAGTGCTCGTCAGCCTG"+
"AGGCTGAAAATAAAATCCGTGGTCACACCCAATAAGTTAGGTACTTTGACTTGGAG"+
"CTGGAGGAATTTGACATAGTCGA**GTTCAG**TTCTTCTCCAGCATCCACGTGAACCGTT"+
"GTAACTATGTTCTGTGCCCACACCAAAAAAACTTTCCACGTGAAAAACGAAAGTCTT"+
"TGGTTTTAATCAATAAGTGCTCTCTTCTCGGAGAGAGAAGGTTGCTTGTCTGCCGAT"+
"GTACTTTATTAAATCCAATAACCACACCAAACTTTCCACGTGTGAACTATACTCCAA"+
"AAACGAAGTATTGGTTTATCATAATCTGAACAAAGAACGATGATGATGATGATAGA"+
"GGAACCTGAGCAGCCATGTCTGAACCTATAATTGGTCGTCGTGCGACTAAATTAGGT"+
"AAAAAAGTAGTTCTAAGAGAATGATTCAATGCAAAGTTCTATTAATCGTTCAATTG";

ריבוע של מילים

מתוך תחרות 2014 של דרום אפריקה (הסיבוב הראשון)
http://www.olympiad.org.za/

נתון מערך 8x8. כתוב תכנית שתמצא בו מילים. ניתן להניח שהמילים מופיעות בשורות ונקראות משמאל לימין (קדימה) או מימין לשמאל (אחורה). התכנית תדפיס את המילה את המיקום שבו היא מתחילה (שורה ועמודה) ואת הכיוון שבו היא ממשיכה (קדימה או אחורה).

למשל, עבור המערך המופיע בהמשך והמילה COLT הפלט יהיה

COLT, row 4, column 3, forwards

בדוק את תכניתך עם הקלטים הבאים:

(a) THE
(הפלט צריך להיות THE, row 1, column 1, forwards)

(b) BEST

(הפלט צריך להיות BEST, row 6, column 5, forwards)

(c) CONTEST

(הפלט צריך להיות CONTEST, row 3, column 7, backward)

T	H	E	Q	T	T	E	P
L	R	E	K	A	W	R	E
T	S	E	T	N	O	C	I
X	Y	C	O	L	T	Z	J
R	C	H	I	L	E	S	Q
U	M	N	X	B	E	S	T
Y	X	K	O	O	C	Q	F
G	H	I	K	C	O	R	B

ריבוע של אותיות א"ב

מתוך תחרות 2013 של דרום אפריקה (הסיבוב הראשון)

http://www.olympiad.org.za/

לכל אות (גדולה או קטנה) מהא״ב יש מספר.

A = a = 1,

B = b = 2,

C = c = 3,

D = d = 4,

E = e = 5,

וכן הלאה.

כתוב תכנית שתקבל כקלט אות בודדת ותדפיס ריבוע שאורכו ורוחבו שווים לערך של אותה האות.

לדוגמה אם הקלט הוא D אז יודפס הריבוע הבא

```
D D D D
D D D D
D D D D
D D D D
```

ואם הקלט הוא e אז יודפס הריבוע הבא

```
e e e e e
e e e e e
e e e e e
e e e e e
e e e e e
```

מצב רוח שמח או עצוב

מתוך תחרות 2015 של קנדה (Junior Division)

http://cemc.math.uwaterloo.ca/contests/computing.html

נהוג להכניס בתוך טקסט סימנים של מצב רוח. למשל

:-) מסמן פנים שמחות, :-(מסמן פנים עצובות.

כתוב תכנית שתקבל טקסט כלשהו ותקבע באיזה מצב רוח נכתבה ההודעה.

קלט: שורה אחת של 1-255 תווים.

הפלט יקבע לפי החוקים האלה:

- אם בשורת הקלט אין סימנים של רגשות (שמח או עצוב), יודפס none.
- אחרת, אם בשורת הקלט יש מספר שווה של סימנים שמחים או עצובים יודפס unsure.
- אחרת, אם בשורת הקלט יש יותר סימני שמחה אז יודפס happy.
- אחרת, אם בשורת הקלט יש יותר סימני עצב אז יודפס sad.

לדוגמה, אם שורת הקלט היא

How are you :-) doing :-(today :-)?

אז הפלט יהיה

happy

אם שורת הקלט היא

:)

אז הפלט יהיה

none

אם שורת הקלט היא

This:-(is str:-(:-(ange te:-)xt.

אז הפלט יהיה

sad

ואם שורת הקלט היא

This:-(is str:-(:-(ange te:-)xt. Gre:-)at:-).

אז הפלט יהיה

unsure

משחק קובייה

מתוך תחרות 2014 של קנדה (Junior Division)
http://cemc.math.uwaterloo.ca/contests/computing.html

אנטוניה ודוד משחקים משחק קובייה. יש שתי קוביות (אחת לכל שחקן) וכל אחד מהשחקנים מתחיל עם 100 נקודות.

בכל תור, זורק כל שחקן את הקובייה שלו. השחקן שהקובייה שלו מצביעה על פחות נקודות – הפסיד. הוא מפסיד את מספר הנקודות שמופיע על הקובייה השנייה.

אם שתי הקוביות מצביעות על אותו מספר, אף אחד מהם לא מפסיד נקודות.

כתוב תכנית שתקבע את מספר הנקודות הסופי במשחק.

הקלט: שתי שורות. בשורה ראשונה מספר שלם n (מספר התורות שיהיו במשחק) $1 \le n \le 15$.

בכל אחת מ-n השורות הבאות יש שני מספרים: התוצאה שהראתה הקובייה של אנטוניה ואחריו התוצאה שהראתה הקובייה של דוד. כל זריקה של קובייה מחזירה מספר בין 1 ל-6.

הפלט: שורה ראשונה מספר הנקודות הסופי של אנטוניה. שורה שניה מספר הנקודות הסופי של דוד.

לדוגמה, אם הקלט הוא

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

אזי הפלט יהיה

```
94
91
```

הסבר: יש 4 סיבובים במשחק.

בסיבוב הראשון דוד מנצח (כי הקובייה של דוד מראה 6 והקובייה של אנטוניה מראה 5). לכן אנטוניה מפסידה 6 נקודות.

בסיבוב השני, יש תיקו בין הזריקות ולכן אף אחד מהמשחקנים לא מאבד נקודות.

בסיבוב השלישי, אנטוניה זוכה (הקובייה שלה מראה 4) ולכן דוד מפסיד 4 נקודות. בסיבוב הרביעי, אנטוניה מנצחת ולכן דוד מפסיד 5 נקודות.

בסך הכל אנטוניה הפסידה 6 נקודות ולכן יש לה 94 נקודות, דוד הפסיד 9 נקודות ולכן יש לו 91 נקודות.

הצבעה

מתוך תחרות 2014 של קנדה (Junior Division)
http://cemc.math.uwaterloo.ca/contests/computing.html

בתחרות שירה נערכת הצבעה אחרי ששתי זמרות סיימו לשיר. המשימה שלך היא לספור את ההצבעות ולקבוע את התוצאות.

הקלט יהיה שתי שורות. בשורה הראשונה יהיה מספר ההצבעות V הכולל ($1 \le V \le 15$). בשורה השנייה סדרה של תווים (A או B) המייצגים את ההצבעות לאחת משתי הזמרות.

הפלט: אחת משלוש אפשרויות:

- אם יש יותר הצבעות ל- A, יודפס A
- אם יש יותר הצבעות ל- B, יודפס B
- אם יש שוויון בהצבעות, יודפס Tie

לדוגמה, אם הקלט הוא

```
6
ABBABB
```

אזי הפלט הוא B

ואם הקלט הוא

```
6
ABBABA
```

אזי הפלט הוא Tie

רבועים מסודרים

מתוך תחרות 2014 של דרום אפריקה (הסיבוב הראשון)

http://www.olympiad.org.za/

מספר מסודר הוא מספר שכל הספרות שלו מסודרות בסדר עולה. למשל, 136 הוא מספר מסודר אבל 163 אינו מספר מסודר. 225 הוא מספר מסודר, אבל 252 אינו מספר מסודר.

מספר של ספרה אחת הוא תמיד מסודר.

כתוב תכנית שתדפיס את כל המספרים המסודרים שהם ריבועים של מספר כלשהו עד ל- v. כאשר v הוא מספר שלם חיובי קטן מ- 100,000.

לדוגמה, אם הקלט הוא v=30

אזי הפלט יהיה 1, 4, 9, 16, 25

בדוק את תכניתך עם הקלטים הבאים:

(a) V = 100

(b) V = 1000

(c) V = 2000

הספרה המועדפת עלי

מתוך תחרות 2008-2009 של DWITE (תחרות ברשת) לבתי ספר תיכוניים סיבוב שני (http://dwite.ca/)

בהינתן מספר שלם. מהי הספרה הגדולה ביותר במספר הזה?

קובץ הקלט: 5 מקרי מבחן, כל אחד מהם שורה אחת, ובה מספר הצעדים 0 <= N <= 1,000,000

קובץ הפלט: 5 שורות, ובכל אחת מהן מספר שלם יחיד 0 <= n <= 9 שהוא הספרה הגדולה ביותר במקרה המבחן המתאים.

לדוגמה, אם הקלט הוא

```
1
10
102
123450
9999
```

אזי הפלט יהיה

```
1
1
2
5
9
```

בילי הנודד

מתוך תחרות 2011-2012 של DWITE (תחרות ברשת) לבתי ספר תיכוניים סיבוב שני (http://dwite.ca/)

בילי, נודד חסר מטרה מביתו (0,0) כיוון שסגרו את חנות הממתקים שאהב.

הוריו המודאגים רוצים לדעת איפה הוא נמצא אחרי N צעדים. הם יודעים שהוא יכול לצעוד רק למטה (על המסך) או ימינה.

בילי הולך לפי התבנית הבאה:

1 צעד לימין, 2 צעדים למטה, 3 צעדים לימין, 4 צעדים למטה, וכולי.

כלומר, הצעדים שלו נראים כך

R D D R R R D D D D …

קובץ הקלט: 5 מקרי מבחן, כל אחד מהם שורה אחת, ובה מספר הצעדים 0 <= N < 100000

קובץ הפלט: 5 שורות ובכל אחת מהן, הקואורדינטות של בילי אחרי N צעדים. (עבור כל אחד ממקרי המבחן)

לדוגמה, אם הקלט הוא

```
0
1
2
3
4
```

אזי הפלט יהיה

```
0   0
1   0
1   -1
1   -2
2   -2
```

ואם הקלט יהיה

```
5
163
1729
90909
100000
```

אזי הפלט יהיה

```
3       -2
81      -82
859     -870
45369   -45540
50048   -49952
```

אתרים נוספים של תחרויות תכנות ברשת

ACM-ICPC International Collegiate Programming Contest	http://cm.baylor.edu/welcome.icpc
American Computer Science League Contests	http://www.acsl.org
American Statistical Association Statistical Computing Student Paper Competition	http://stat-computing.org/awards/student/announcement.html
Amrita University - Lord of the Code	http://amrita.edu/loc
Baltic Olympiad in Informatics – BOI	http://www.boi2012.lv/
British Informatics Olympiad (BIO)	http://www.olympiad.org.uk
Canadian Computing Competition	http://cemc.math.uwaterloo.ca/contests/computing.html
Central-European Olympiad in Informatics – CEOI	http://ceoi.inf.elte.hu
CS Games Social Computing Competition	http://csgames.org/corpo
CyberCenturion (UK)	http://cybersecuritychallenge.org.uk/competitors/cybercenturion
CyberPatriot (US)	http://www.uscyberpatriot.org/home
Dare to be Digital Video Game Competition	http://www.daretobedigital.com/
Dream it. Code it. Win it. Programming Competition	https://dreamitcodeitwinit.wordpress.com/
DWITE – Online Computer Programming Competition	http://dwite.ca/
French-Australian Regional Informatics Olympiad – FARIO	http://orac.amt.edu.au/fario
German National Computer Science Competition	http://www.bmbf.de/en/448.php
Google Call to Code	https://www.calltocode.ie/
Google Code Jam	http://code.google.com/codejam
HP CodeWars	http://www.hpcodewars.org/
HPCAC-ISC Student Cluster Competition	http://www.isc-events.com/isc12/student_cluster.html

IEEExtreme 24 Hour Programming Competition	http://www.ieee.org/membership_services/membership/students/competitions/xtreme/index.html
Imagine Cup	http://www.imaginecup.com/
Indian Computing Olympiad	http://www.iarcs.org.in/inoi
Intel International Science and Engineering Fair	http://www.societyforscience.org/isef
Intel Science Talent Search	http://www.societyforscience.org/STS
International Olympiad in Informatics	http://www.ioinformatics.org/index.shtml
Internet Problem Solving Contest (IPSC)	http://ipsc.ksp.sk/
Irish Schools' Programming Competition	http://aispc.computing.dcu.ie/
MTConnect Student Challenge	http://web.ncdmm.org/cn/ackdg/Mtconnect
National STEM Video Game Competition	http://stemchallenge.org/
ProjectCSGIRLS Competition	http://www.projectcsgirls.com/competition-details.html
SEARCC International Internet Competition	http://www.searcc.org/competitions/international-internet-competition
SEARCC International Micromouse Competition	http://www.searcc.org/competitions/international-micromouse-competition
SEARCC International Schools Software Competition	http://www.searcc.org/competitions/international-schools-software-competition
Siemens Competition in Math, Science and Technology	http://www.siemens-foundation.org/en/competition.htm
South African Computer Olympiad	http://www.olympiad.org.za/
Student Cluster Competition	http://sc12.supercomputing.org/content/student-cluster-competition
UK Bebras Computational Thinking Challenge	http://www.beaver-comp.org.uk/
USA Computing Olympiad	http://www.usaco.org/
Zero Robotics Competition	http://www.zerorobotics.org/

סמינר למורים מובילים במדעי המחשב, צפת

ביולי 2016 התקיים במלון רות רימונים בצפת סמינר קיץ תשע״ו למורים מובילים במדעי המחשב, בהשתתפות 39 מורים. ההרצאות שניתנו בסמינר:

ד״ר עפרה ברנדס, האוניברסיטה העברית: מי אתם המורים המובילים? - מסקנותיו של מחקר אקדמי ראשון בנושאי ידע של מורים מובילים והתפתחותם המקצועית בתקופות של שינויים בתכניות לימודים

ד״ר רונית בן בסט, תיכון גן נחום ראשון לציון ומכון ויצמן למדע: שימוש בטכנולוגיה חינוכית בהוראת מדעי המחשב והשפעתן על העמדות של תלמידים לבחור מקצוע מדעי בתיכון

יעקבסון זהבה, אורט גוטמן ומכללת הרמלין נתניה: ההבדל העקרוני בפרוייקט בהתמחות הגנת סייבר ובהתמחויות אחרות

הוד אריק, גימנסיה עברית הרצליה תל-אביב: לימוד תכנות באופן לא קונבנציונלי - לימוד תכנות ככלי טיפולי

בן שחר שלומית, תיכון מכבים רעות מו״ר ומקיף ב׳ ע״ש רבין מודיעין: כוחה של קהילה

שטיינר משה, תיכון ברנקו וייס, פרדס חנה: פיתוח משחק שולה מוקשים בכיתת יסודות בשיטת דפים מדורגים

פרופ׳ רן קנטי, בית הספר למדעי המחשב ע״ש בלבטניק, אוניברסיטת תל אביב: מיסוך קריפטוגרפי של תכניות מחשב: עולם חדש ואמיץ

גרינולד אוי, חוגים חיפה: דיון בשאלה מתי פעולות מחלקה ומתי פעולות מופע של עצם?

ד״ר גורן גורדון, מנהל המעבדה לסקרנות, הפקולטה להנדסה, אוניברסיטת תל אביב: סקרנות של ילדים ורובוטים

פרופ׳ אורית חזן, הפקולטה לחינוך למדע וטכנולוגיה, הטכניון: ישראל, ניו-יורק והוראת מדעי המחשב: מה ניתן ללמוד מהניסיון הישראלי?

ד״ר מיכל גורדון, המכון הטכנולוגי בחולון: טיפוח חשיבה לא-דטרמיניסטית

צדיק אסף, גימנסיה גן נחום, ראשון לציון: רובוטיקה

פקר ולרי, אורט דקל וילנאי מעלה אדומים, ואורט תעופה וחלל, מעלה אדומים: תורת המשחקים

אורן גיא, גימנסיה ריאלית ע״ש קררי, ראשון לציון: משחקים בסביבת java – libgdx

מריאנובסקי אלה, תיכון עתיד למדעים, לוד: מודל השכבות OSI

הרשקוביץ אתי, קוגל חולון ותיכון מו״ר, מודיעין: בניית תרבות של מצוינות

עוד בסמינר, התקיימה סדנה מיוחדת בהנחיית שוורץ שמואל, תיכון הדרים הוד השרון : PBL - כהיבט מאתגר למשלי כב פסוק ו (חֲנֹךְ לַנַּעַר, עַל-פִּי דַרְכּוֹ-- גַּם כִּי-יַזְקִין, לֹא-יָסוּר מִמֶּנָּה.)

כמו כן, התקיים דיון על תכנית הלימודים במדעי המחשב לכיתות ד-ו בהנחיית פרופ' אורית חזן מהטכניון.

ביום השני של הסמינר, המשתתפים יצאו לסיור מודרך בצפת ולאחריו אכלו ארוחת ערב במסעדת הארי בצפת.

.... החומרים שהוצגו בסמינר ניתנים לצפייה באתר המרכז הארצי ....

ד"ר עפרה ברנדס

ד"ר רונית בן בסט

זהבה יעקבסון

אריק הוד

שלומית בן שחר

משה שטיינר

פרופ' רן קנטי

אביטל (Evi) גרינולד

ד"ר גורן גורדון

פרופ' אורית חזן

ד"ר מיכל גורדון

אסף צדיק

שמוליק שוורץ

גיא אורן

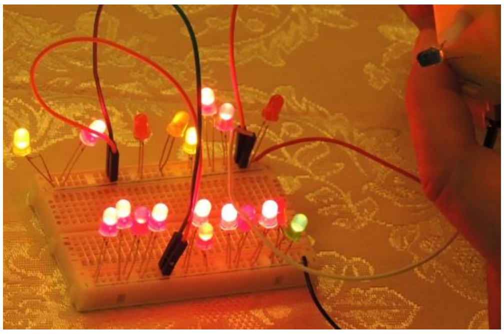
אלה מריאנובסקי

אתי הרשקוביץ

משתתפי הסמינר בסיור מודרך ברחובות צפת

משתתפי סמינר קיץ 2016

אברך שטיין אורנה, אברמוביץ ישראל, ד״ר אהרוני דן, אורן גיא, אלנקרי רוני, בן דוד דורית, בן ישי דני, ד״ר בן בסט רונית, בן שחר שלומית, ברנדס עפרה, ברסוק ז׳אנה, גורדון גורן ומיכל בתוספת 3 ילדיהם, גרינולד אביטל (Evi), דאום-רייטר ברכה, דגן יריב, פרופ׳ דולב שלומי, הוד אריק, הרשקוביץ אתי, ד״ר זוהר דורון, זמוסטיאנו יאנה, פרופ׳ חזן אורית, טבלין איזבלה, טוראל חני, יעקבסון זהבה, יפה ריקי, ירון יוסי, כהן דורית, לב אלה, לבילב אירנה, לוי-רשתי דפנה, לולב שרית, ד״ר לפידות תמי, מוכתר אבל דגנית, מינסטר דפנה, מריאנובסקי אלה, עבודי מני, פונק שרה, ד״ר פז תמר, פקר ולרי, פרג׳ון ויליאם, פרץ אילן, צדיק אסף, קורצ׳ינסקי טוני, קשת דני, שטיינר משה, שריקי גלית



תמונות הסמינר המופיעות בעיתון צולמו ע״י ד״ר דן אהרוני, יריב דגן, ואיזבלה טבלין

.... להתראות בסמינר הבא ....

הבטים בהוראת מדעי המחשב: משוב לגיליון פברואר 2017

קוראים יקרים

לאחר שסיימתם לקרוא את העיתון, אנא מלאו ושלחו משוב זה בהקדם למינהלת מל״מ.
תודה על שיתוף הפעולה,
צוות המרכז הארצי

משוב לגיליון פברואר 2017 של "הבטים בהוראת מדעי המחשב"

נא למלא ולשלוח אל:
מינהלת מל"מ
בניין קנדה, קומה 1
קריית הטכניון, חיפה 3200003
פקס: 04-8295010

1. שם בית הספר ______________________________

2. שם המשיב________________ מספר המורים שעיינו בגיליון זה בבי״ס ________

3. סמנו במשבצת המתאימה את חוות דעתכם :

הערות

חוות דעת כללית על הגליון	טובה מאד	טובה	לא טובה	
החשיבות של כתב העת	רבה מאד	רבה	לא חשוב	
מידת העניין	מעניין מאד	מעניין	לא מעניין	
תרומה לעבודתי	תורם מאד	תורם	לא תורם	

4. הערות נוספות :

____
____
____
____
____