

עבודה 1 - אוטומציה

מס' קבוצה	שם התרגיל	תאריך הגשה
13	GIT - 1	27/03/2023
שמות ומספרי ת"ז של המגישים		
עידו גולדמן 314971078	סער ביטס 313595829	עמית פדאלי 206225260

1. מטרות הפרויקט –

במסגרת התרגיל נתבקשנו לכתוב קוד ליצירת מחשבון המרה בין בסיס עשרוני לבסיס הקסדצימלי (בסיס 16) ומבסיס הקסדצימלי לבסיס עשרוני. את הקוד כתבנו בשפת JAVA בתוכנת Eclipse. עבור כל מחשבון המרה – ראשית כתבנו את הפונקציות המבצעות את ההמרה, לאחר מכן הוספנו התמודדויות עם מקרי קצה ולאחר מכן ביצענו בדיקות עבור מגוון רחב של מספרים כדי לוודא שהקלט נכון עבור כל פלט. לבסוף את הקוד המוגמר העלינו דרך תוכנת GIT לענף שיצרנו, ומשם הוא יעבור לענף הראשי של הקורס.

2. הנחות יסוד בפיתוח המערכת -

עבור הקוד שכתבנו הנחנו מספר הנחות מרכזיות :

- 2.1. הפלט שהמשתמש יזין לא יעלה על 50 ספרות/אותיות – הדבר נובע ממערך שהגדרנו להזנה ובו הגבלה עד 50 תווים.
- 2.2. המספר אותו המשתמש יזין במעבר בין הבסיסים יהיה מקטגוריית int בלבד ולא עבור מספרים עשרוניים.
- 2.3. במעבר מבסיס הקסדצימלי לבסיס עשרוני המשתמש יזין אותיות גדולות בלבד.
- 2.4. המספר המקסימלי עבור int בג'אווה הינו 2,147,483,647, ולכן עבור המעבר מבסיס עשרוני לבסיס הקסדצימלי ולהפך יהיה במקסימום עבור המספר 7FFFFFFF.
- 2.5. המספר 0 נכלל גם הוא באפשרויות המשתמש להזנה ויניב את התוצאה 0.

3. תיאור מצבי קיצון ושיטת הפיתרון –

המרה מעשרוני להקסדצימלי –

- 3.1. עבודה עם מספרים שליליים – עבור הכנסת מספר עשרוני שלילי, בדקנו האם במערך השאריות המספרים שליליים ובהתאם לכך הוספנו מינוס לפני המספר והכפלנו את הערכים במערך ב (-1) כדי לקבלם בערך מוחלט. כך גם התוצאה הסופית מתבטאת במינוס.

3.2. הכנסת תווים לא תואמים – עבור מספרים עשרוניים, ביצענו אכיפה כך שרק מספרים מסוג Integer יעבדו במחשבון העשרוני. במקרה שהוכנס תו מסוג אחר המשתמש יתבקש להכניס מספר מחדש.

המרה מהקסדימלי לעשרוני –

3.3. כאמור, המחשבון ההקסדימלי עובד עם אותיות A עד F בלבד, לכן עבור אותיות שאינן בטווח הזה תצר שגיאה למשתמש והוא יזין את המספר המבוקש מחדש.

3.4. כמו כן, הפלט המוכנס מתקבל ע"י string, ולכן גם נוספה הגבלה על תו של ספרה מסוימת, במקרה של תו שלא מתאים להגבלות תוצג למשתמש שגיאה.

4. תיעוד קוד התוכנית –

בחירת המשתמש במחשבון ההמרה הרצוי –

```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
    System.out.println("choose base : ");  
    System.out.println("1- Hexadecimal to Decimal");  
    System.out.println("2 - Decimal to Hexadecimal");  
    int a = sc.nextInt();  
    if(a ==1) {
```

עבור הזנת המספר 1 – יעבור המשתמש למחשבון המרה מהקסדימלי לעשרוני.
עבור הזנת המספר 2 – יעבור המשתמש למחשבון המרה מעשרוני להקסדימלי.

שיטה לחישוב המרה מהקסדימלי לעשרוני -

```
static Scanner sc = new Scanner(System.in);  
public static int hexToDecimal(String hexnumber){  
    String hstring = "0123456789ABCDEF";  
    int num = 0;  
    for (int i = 0; i < hexnumber.length(); i++)  
    {  
        char ch = hexnumber.charAt(i);  
        int n = hstring.indexOf(ch);  
        num = 16*num + n;  
    }  
    return num;  
}
```

לולאה אשר עוברת על המחרוזת שהוכנסה, כאשר עבור כל תו מתבצעת המרה למספר, במקרה ומדובר באות, הכפלה ב – 16 והוספת הספרה שכבר חושבה לפני כן, ולבסוף נחזיר את המספר הכולל שיצא.

אכיפות עבור המרה מהקסדצימלי לעשרוני -

```
boolean flag = false;
while(flag == false) {

    System.out.println("enter string : ");
    String hexnum = scanner.nextLine();
    for (int i = 0; i < hexnum.length(); i++) { |
    if (hexnum.charAt(i) < 'A' && hexnum.charAt(i) > 'F' || ( hexnum.charAt(i) < '0' && hexnum.charAt(i) > '9')) {
        System.out.println("false! enter new string");
        flag = false;
        break;
    }
    }
    flag = true;
    System.out.println("Decimal equivalent  "+hexToDecimal(hexnum));
}
}
```

מעבר על המחרוזת ובדיקה האם יש עמידה בתנאי הסף שהוגדו עבור האותיות הלוועזיות ועבור תווי המספרים.

קוד המרה מבסיס עשרוני להקסדצימלי -

```
if(number[0] < 0)
    System.out.print("-");

for (int j = index - 1; j >= 0; j--) {
    if(number[j] > 0)
    { if (number[j] > 9)
        System.out.print((char)(55 + number[j]));
      else
        System.out.print(number[j]);
    }
    if(number[j] < 0) {
        if (number[j] <= - 9)
            System.out.print((char)(55 + number[j]*(-1)));
        else
            System.out.print(number[j]*(-1));
    }
}
}
```

מעבר על המחרוזת שהוכנסה, הגדלת הספרה ב 55 במידה ועומדת בתנאי הסף, חלוקה בשארית בכדי לקבל בסיס הקסדצימלי.

אכיפת מקרי הקיצון עבור הבסיס העשרוני -

```
int x = 0;
boolean validInput = false;
while (!validInput) {
    try {
        System.out.println("Enter a number to convert");
        x = sc.nextInt();
        validInput = true;
    } catch (Exception e) {
        System.out.println("Invalid input. Please enter an decimal number");
        sc.nextLine();
    }
}

int d=0;
int [] number= new int [10];
int index =0;
while(x!=0)
{
    d=x%16;
    x=x/16;
    number [index]=d;
    index++;
}
```

פליטים לדוגמא –

```
choose base :  
1- HexaDecimal to Decimal  
2 - Decimal to HexaDecimal  
2  
Enter a number to convert  
1732  
6C4
```

```
choose base :  
1- HexaDecimal to Decimal  
2 - Decimal to HexaDecimal  
  
1  
enter string :  
1C3  
Decimal equivalent  451
```

```
choose base :  
1- HexaDecimal to Decimal  
2 - Decimal to HexaDecimal  
2  
Enter a number to convert  
1A2  
Invalid input. Please enter an decimal number  
Enter a number to convert  
12356
```

5. סיכום ומסקנות -

במהלך העבודה למדנו כיצד יש לעבוד עם תוכנת GIT להעלאת קבצים וכיצד ליצור ענף השייך לקבוצת העבודה שלנו בלבד. בנוסף הסרנו חלודה בעבודה עם קוד JAVA וביצוע המרות בין בסיסים שונים, תוך מתן תשומת לב להתמודדות עם מקרי קיצון שונים, וחשיבה כיצד להיטיב ולהקל על המשתמש הרוצה להיעזר במחשבון שבנינו. העבודה עם מעבר בין בסיסים חיזקה את ההבנה שלנו לגבי הבסיסים השונים וכיצד יש לממש את הידע שלנו על המעבר בין בסיסים גם בכתיבת קוד.