



תרגיל 1 - קבוצה 14:

lez lchiani 208195388 209455674 יובל ציפורה רון ממן 209164839

1. מטרת הפרויקט: מחשבון להמרת בסיסים (הקסדצימלי - בינארי)

מטרת הפרויקט היא לפתח מחשבון בשפת Python או Java המסוגל לבצע המרת בין מספרים בסיס 2 (Binari) לבסיס הקסדצימלי (Base 16) ולהפך. הפרויקט nodע לאפשר שימוש נוח ומהיר במחשבון להמרת מספרים בין שני הבסיסים תוך בדיקה ואיומות של תקיןות הקלטים. לאחר בחינת האפשרויות, החלנו למש את הפרויקט בשפת **Python**. לצורך ניהול הקוד והagationו, השתמשנו ב**Git**. מטרת התרגיל הייתה לרכוש מיומנויות ניהול גרסאות, ללמוד כיצד לעבוד עם מאגרי קוד משותפים, ולהבין את תהליכי ההעלה, העדכון והמייזוג של קבצים בסביבת **Git**.

2. הנחות יסוד בפיתוח המערכת:

- א. **הנתת קלט ריק** – אנו מניחים שימושה אשר מרים את הקוד אcn רוצה להשתמש במחשבון שלנו ולבצע המרתם ולcn לא יזין קלט ריק.
- ב. **רציפות התהילה** - הקוד מניח כי ברגע שהמחשבון מופעל, הוא ירץ באופן רציף עד שהשתמש יבחר לסיס את השימוש בו. אין מנגנון להפסקת התהילה באופן בלתי צפוי (למשל, עקב תקלה במערכת או סגירה כפואה של התוכנית).
- ג. **השלמת המרת אחת לפחות** - מרגע הרצת הקוד בפעם הראשונה אנו מניחים שהשימוש יבצע לפחות המרת המרת אחת לפני שייסגור את המחשבון.
- ד. **המשתמש לא יזין נקודה בלבד** – המחשבון שלנו תומך בנקודות עשרוניות ובמינוסים, אך תחת ההנחה שהמשתמש לא יכנס נקודה בלבד ולא יכנס מספר.

3. תיאור מצבי הקיצון ושיטת הפתרון:

א. בחירה שגוייה בתפריט:

בעיה: המשתמש עלול להזין מספר שהוא לא 1 או 2 בתפריט הבחירה הראשי.
פתרונות: כדי למנוע שגיאות, המערכת בודקת אם הערך שהוזן נמצא בין האפשרויות התקינות אם הקלט אינו תקין, מוצגת הודעה שגיאה, והתפריט מוצג מחדש. כך אנו מונעים קriseה של התוכנית ומבטיחים שהמשתמש יבצע בחירה תקינה.

```
Choose a conversion:  
1. Binary - Hexadecimal  
2. Hexadecimal - Binary  
Enter 1 or 2: 5  
Invalid choice. Let's try again.
```



הפקולטה למדעי ההנדסה - המחלקה להנדסת תעשייה וניהול

אוטומציה וייצור ממוחשב 364-1-3321

אוניברסיטת בן גוריון בנגב

ב. קלט לא חוקי (תווים שאינם מתאימים לבסיס):

בעיה: משתמש עלול להזין תווים שאינם חלק מהבסיס הנכון, למשל "102" ביבנארי או "H1" בהקסדצימלי.

פתרון: ביצענו בדיקה באמצעות (notall(c in '01' for c in binary_str) עבור בינארי, ועבור הקסדצימלי notall(c in hex_digits for c in hex_str) אם נמצא תווים שאינם מוגדרים בסיס).

```
Choose a conversion:  
1. Binary - Hexadecimal  
2. Hexadecimal - Binary  
Enter 1 or 2: 1  
Enter a binary number (only 0s and 1s): @  
Not a valid binary number. Try again.
```

```
Choose a conversion:  
1. Binary - Hexadecimal  
2. Hexadecimal - Binary  
Enter 1 or 2: 2  
Enter a hexadecimal number (0-9, A-F): %  
Not a valid hexadecimal number. Try again.
```



הפקולטה למדעי ההנדסה - המחלקה להנדסת תעשייה וניהול
אוטומציה וייצור ממוחשב 364-1-3321
אוניברסיטת בן גוריון בנגב

ג. מספרים עם רווחים עודפים לפני ואחרי:

בעיה: אם המשתמש מזין "55 " (עם רווחים בתחילת או סוף הקלט), זה עלול לגרום לשגיאה.

פתרון: השתמשנו ב(`strip` כדי להסיר רווחים לפני ביצוע הבדיקה (אך המשתמש יצרך להזין מספר תקין
ללא רווחים באמצע המספר).

```
Enter a hexadecimal number (0-9, A-F): 55
Binary: 1010101

Do you want to convert another number? (yes/no): yes
Choose a conversion:
1. Binary - Hexadecimal
2. Hexadecimal - Binary
Enter 1 or 2: 2
Enter a hexadecimal number (0-9, A-F): 55
Binary: 1010101
```

ד. קלט ריק בתפריט ההתחלתי:

בעיה: המשתמש עשו ללחוץ על Enter מבלי לבחור אפשרות 1 או 2.

פתרון: כחלק מבדיקה קלט תקין בתפריט הבחירה הראשי 1 או 2 גם קלט ריק אינו תקין ובמצב זה
המשתמש נדרש לבחור פעם נוספת.

```
Choose a conversion:
1. Binary - Hexadecimal
2. Hexadecimal - Binary
Enter 1 or 2:
Invalid choice. Let's try again.
```



4. תיעוד קוד התוכנית:

```
def binary_to_hex(binary_str):  
    binary_str = binary_str.strip()  
    # Check for negative sign  
    is_negative = binary_str.startswith('-')  
    if is_negative:  
        binary_str = binary_str[1:]  
    # Check for decimal point  
    if '.' in binary_str:  
        int_part, frac_part = binary_str.split('.')  
    else:  
        int_part, frac_part = binary_str, ''  
    # Validate characters  
    if not all(c in '01' for c in int_part + frac_part):  
        return None  
    # Convert integer part  
    while len(int_part) % 4 != 0:  
        int_part = '0' + int_part  
    int_decimal = int(int_part, 2)  
    int_hex = hex(int_decimal)[2:].upper()  
    # Convert fractional part  
    frac_hex = ''  
    if frac_part:  
        frac_decimal = 0  
        for i, bit in enumerate(frac_part, start=1):  
            frac_decimal += int(bit) * (1 / (2 ** i))  
        # Convert fractional decimal to hex  
        count = 0  
        while frac_decimal > 0 and count < 10:  
            frac_decimal *= 16  
            digit = int(frac_decimal)  
            frac_hex += hex(digit)[2:].upper()  
            frac_decimal -= digit  
            count += 1  
    result = int_hex  
    if frac_hex:  
        result += '.' + frac_hex  
    if is_negative:  
        result = '-' + result  
    return result
```



א. פונקציית binary to hex

- הסרת רווחים מיותרם עם `strip()`
- בדיקה אם המספר שלילי:
 - אם מתחילה במינוס שומרים את זה בצד ומתייחסים למספר חיובי.
- פיצול לפי נקודה עשרונית:
 - אם יש נקודה עשרונית – מפראים ל-`int_part` (החלק השלים) ו-`frac_part` (החלק העשורי).
 - אם אין – רק חלק שלם.
- בדיקת תקינות:
 - מוגדים שכל התווים הם רק 0 או 1 (בבסיס 2)
- המראה של חלק השלים:
 - משלימים אפסים משמאל כדי שהאורך יהיה כפול של 4.
 - מmirrim לבסיס עשרוני, ואז לHex
- המראה של חלק העשורי:
 - מחשבים את הערך העשורי של החלק העשורי הבינארי.
 - מmirrim אותו לבסיס 16
- שילוב התוצאות:
 - לחברים את החלקים השלם + עשרוני.
 - מוסיף מינוס אם זה היה מספר שלילי.

```
def hex_to_binary(hex_str):  
    hex_str = hex_str.strip().upper()  
    # Check for negative sign  
    is_negative = hex_str.startswith('-')  
    if is_negative:  
        hex_str = hex_str[1:]  
    # Check for decimal point  
    if '.' in hex_str:  
        int_part, frac_part = hex_str.split('.')    else:  
        int_part, frac_part = hex_str, ''  
    valid_hex = "0123456789ABCDEF"  
    if not all(c in valid_hex for c in int_part + frac_part):  
        return None  
    # Convert integer part  
    int_decimal = int(int_part, 16)  
    int_binary = bin(int_decimal)[2:]  
    # Convert fractional part  
    frac_binary = ''  
    if frac_part:  
        frac_decimal = 0  
        for i, digit in enumerate(frac_part):  
            frac_decimal += int(valid_hex.index(digit)) * (1 / (16 ** (i + 1)))  
        # Convert fractional decimal to binary  
        count = 0  
        while frac_decimal > 0 and count < 20:  
            frac_decimal *= 2  
            bit = int(frac_decimal)  
            frac_binary += str(bit)  
            frac_decimal -= bit  
            count += 1  
    result = int_binary  
    if frac_binary:  
        result += '.' + frac_binary  
    if is_negative:  
        result = '-' + result  
    return result
```



הפקולטה למדעי ההנדסה - המחלקה להנדסת תעשייה וניהול
אוטומציה ויצור ממוחשב
אוניברסיטת בן גוריון בנגב

ב. פונקציית hex to binary

- הסרת רווחים מיותרם עם `strip()`
- בדיקה האם המספר שלילי:
- אם המספר מתחילה ב민וס שומרים את המינוס בצד ונתיחס למספר חיובי.
- פיצול לפי נקודה עשרונית:
 - אם יש נקודה עשרונית מפרטים ל- `frac_part` (חלק שלם) ו `int_part` (החלק עשרוני)
 - אם אין נקודה עשרונית – יש רק חלק שלם.
 - בדיקה תקינות:
 - מודדים שכל התווים הם תוויות חוקיים של בסיס 16 (0–9, A–F).
 - הمرة של חלק השלים:
 - ממירם את החלק השלים לעשרוני, אז לבינארי.
 - הمرة של חלק העשרוני:
 - מחשבים את הערך של חלק העשרוני הקסדצימלי.
 - ממירם את הערך זהה לבינארי.
- שילוב התוצאה:
 - מחברים את החלקים (שלם + עשרוני).
 - מוסיפים מינוס אם זה היה מספר שלילי.

```
27 def run_calculator():    usage
28     print("Welcome to the Lazy People Base Converter!")
29     print("We do the hard math, you just type stuff. Fair deal!\n")
30
31     while True:
32         # Ask the user to choose conversion type
33         print("Choose a conversion:")
34         print("1. Binary - Hexadecimal")
35         print("2. Hexadecimal - Binary")
36         choice = input("Enter 1 or 2: ").strip()
37
38         # If input is not valid, ask again
39         if choice not in ['1', '2']:
40             print("Invalid choice. Let's try again.\n")
41             continue
42
43         if choice == '1':
44             while True:
45                 # Get binary input from user
46                 binary_input = input("Enter a binary number (only 0s and 1s): ").strip()
47                 result = binary_to_hex(binary_input)
48                 if result is not None:
49                     print(f"Hexadecimal: {result}\n")
50                     break
51                 else:
52                     print("Not a valid binary number. Try again.\n")
53
54         elif choice == '2':
55             while True:
56                 # Get hexadecimal input from user
57                 hex_input = input("Enter a hexadecimal number (0-9, A-F): ").strip()
58                 result = hex_to_binary(hex_input)
59                 if result is not None:
60                     print(f"Binary: {result}\n")
61                     break
62                 else:
63                     print("Not a valid hexadecimal number. Try again.\n")
64
65     # Ask if the user wants to do another conversion
66     while True:
67         again = input("Do you want to convert another number? (yes/no): ").strip().lower()
68         if again == 'no':
69             print("\nThanks for using the Lazy People Base Converter! Bye")
70             break
71         elif again != 'yes':
72             print("invalid input. Try again.\n")
73         else:
74             break
75     if again == 'no':
76         break
77
78 # Run the calculator
run_calculator()
```



ג. פונקציה calculator:

- מציגה למשתמש הודעה פתיחה ותפריט בחירה.
- מחיבת את המשתמש לבחור 1 (בינהרி להקסדצימלי) או 2 (הקסדצימלי לבינהריה).
- במצב 1 המשתמש מזין מספר בינהריה והפונקציה בודקת אם הקלט תקין, אחרת המשתמש יתבקש לרשום מספר שוב.
- במצב 2 המשתמש מזין מספר הקסדצימלי והפונקציה בודקת אם הקלט תקין, אחרת המשתמש יתבקש לרשום מספר שוב.
- הפונקציה שואלת את המשתמש אם ברצונו להמיר מספר נוסף, ואם הקלט לא תקין, הפונקציה מבקשת להזין קלט חדש. (yes או no).
- אם המשתמש מקליד סח התוכנית נסגרת עם הודעה סיום.

ד. הפעלת המחשבון:

- התוכנית קוראת לפונקציה (`calculator run`) כדי להתחיל את תהליך ההמרה.

5. סיכום ומסקנות:

במהלך הפרויקט למדנו כיצד להעלות קבצים לתוך repository משותף ב Git - הסבכנו את הצעדים הנדרשים, כולל יצירת repository, ביצוע commit לקבצים והעלאתם לענן באמצעות פקודות כמו `git push`.
בנוסף, עשינו פרויקט תכוני בפייתון שעסוק בהמרת בסיסים (בינהריה והקסדצימלי). במהלך העבודה על הפרויקט הבנו את הדרך בה מספרים מיוצגיםמערכות שונות, ופיתחנו פונקציות הממירות בין בסיסים.
הסקנו כי עבודה עם Git היא מיומנות חיונית לכל מפתח, במיוחד בפרויקטים שבמוספים על שיתוף פעולה. כמו כן, המרת בין בסיסים עוזרת לנו להבין את עקרונות הייצוג של נתונים במחשב ולפתח פתרונות מדויקים ויעילים.

ובכן האיש! למדנו לא לדוחות משימות (נדרשנו לבקש אישור פעמיים ☺)



הפקולטה למדעי ההנדסה - המחלקה להנדסת תעשייה וניהול
אוטומציה ויזור מחשב 364-1-3321
אוניברסיטת בן גוריון בנגב

הוראות הפעלה – קבוצה 14:

בנינו מחשבון אשר מאפשר לבצע המרה בין בסיס בינארי לבסיס הקסדצימלי ולהפך, בצורה פשוטה ונוחה, בעת הפעלת המחשבון, תתבקשו לבחור אחת מתוך שתי האפשרויות הבאות:

אפשרות 1: המרת מספר ביןארי להקסדצימלי:

בחר 1 ולחץ **Enter**

הוזן מספר ביןארי (מכיל רק את הספרות 0 ו-1).

- דוגמה לקלט תקין - 1010

התוצאה תוצג במסך בפורמט הקסדצימלי.

אפשרות 2: המרת מספר הקסדצימלי לבינארי:

בחר 2 ולחץ **Enter**

הוזן מספר הקסדצימלי (מכיל ספרות 0-9 ואותיות F – A בלבד)

דוגמה לקלט תקין - 1F

התוצאה תוצג במסך בפורמט ביןארי.

יציאה מהתוכנית:

לאחר ביצוע המירה, תישאל האם ברצונך לבצע המרה נוספת:

- הקלד **yes** כדי המשיך.

- הקלד **סח** כדי לצאת מהתוכנית.

במקרה של יציאה, תוצג הודעה סיום.

מצבי שגיאה:

- אם הוזן ערך שאינו תואם לקלט הצפיי (למשל תוויים שאינם 0 או 1 בקלט ביןארי), תוצג הודעה:
Not a valid binary number. Try again

- אם הוזן ערך לא חוקי כהקסדצימלי (תוויים שאינם מ-0 עד 9 או מ A עד F) תוצג הודעה:
Not a valid hexadecimal number. Try again

- אם נבחרה אפשרות שאינה קיימת בתפריט (למשל 3), תוצג הודעה:

Invalid choice. Let's try again

- אם התשובה לשאלת האם להמשיך אינה "yes" או "סח" תוצג הודעה:
Invalid input. Try again