



אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

קורס אוטומציה

דו"ח מעבדה 1

תרגיל: Git

קבוצה: 20

מגישים:

עומר מיטרני 318678869

איתי בודניוק 315824698

גולן שמש 315392290

טל כלפון 205681125

מטרת הפרויקט:

מטרת המשימה הייתה להכיר את השימוש בכלי ה-Git וליישם אותו בעבודה קבוצתית, תוך כדי פיתוח מחשבון בשפת Python שמבצע המרה של מספרים בין בסיס הקסדצימלי לבסיס בינארי, ולהיפך. במסגרת התרגיל פותחה תוכנית המאפשרת להזין מספר באחד מהבסיסים ולקבל את ייצוגו בייצוג הבסיס האחר. המערכת תוכננה כך שתטפל בכל קלט תקין וכן תזהה ותטפל באופן נכון במקרה של קלט שגוי, למשל הכנסת תו לא חוקי תחת הבסיס מוצא שנבחר.

הנחות יסוד:

בפיתוח המערכת הנחנו מספר הנחות יסוד אשר להביא לידיעת המשתמש: שימוש בפונקציות מובנות: ביצענו שימוש בפונקציות מובנות בשפת Python להמרה בין הבסיסים הרצויים, לשם כך נדרשנו לבצע Slicing על הפלט של פונקציות אלו כך שיתאים לדרישות המערכת והמשתמש. סוגי קלט נתמכים: המערכת מטפלת בהמרת מספרים שלמים בלבד. כלומר, ההנחה היא שהמשתמש יזין מספר שלם בבסיס המתאים. כמו כן, המערכת מתמקדת במספרים לא שליליים בלבד והכנסת תו "-" תגרוור לשגיאת קלט לא תקין. פורמט ייצוג הקלט: עבור קלט בינארי, מצופה מהמשתמש להזין רק ספרות 0 ו-1. עבור קלט הקסדצימלי, מצופה להזין ספרות 0-9 ואותיות F-A כפי שנהוג. בחרנו להרחיב את גמישות המערכת ולאפשר למשתמש להזין גם אותיות קטנות f-a.

מצבי קיצון:

במהלך פיתוח המערכת זוהו מספר מצבי קיצון שעשוי להיעשות על ידי המשתמש ולכן ביצענו טיפול מיוחד קודם לכן. להלן המצבים העיקריים והטיפול בהם: תווים לא חוקיים בקלט: במקרה שבו המשתמש בוחר להמיר מספר מבסיס בינארי אך מזין מחרוזת שמכילה תו שאינו 0 או 1, המערכת תזהה את החריגה ותטפל בה. הפתרון שיישמונו הוא מעבר על תווי הקלט באמצעות לולאה ובדיקה שכל תו עומד בדרישות הבסיס שנבחר, בסיס בינארי. אם מתגלה תו חריג, התוכנה לא תנסה לבצע המרה ומיד תציג הודעת שגיאה מתאימה למשתמש במקום לקרוס בעת קריאת קלט לא תקין לפונקציית ההמרה של Python. באופן דומה, מקרה קיצון זה מטופל על ידי פונקציה מקבילה המתאימה לבסיס הקסדיצמלי. קלט ריק: אם המשתמש יזין ערך ריק בטרמינל המערכת, מחרוזת הקלט הריקה מזוהה במפורש כמצב קיצון. במקרה כזה התוכנית תטפל בכך כקלט לא תקין ותדווח על שגיאה ותמנע מקריאה שתקריס את המערכת. בחירת בסיס מוצא תקין: המערכת מוודא כי הבסיס מוצא שנבחר הינו בהכרח מבין שני הבסיסים הנתמכים, אחרת המערכת תחדול ותיידע את המשתמש באמצעות שגיאה.

```

# check if binary input is only 0 or 1
def is_valid_binary(s):
    for ch in s:
        if ch != '0' and ch != '1':
            return False
    return True

# check if hexa input is number or a-f or A-F
def is_valid_hexadecimal(s):
    for ch in s:
        if not (
            ('0' <= ch <= '9') or
            ('a' <= ch <= 'f') or
            ('A' <= ch <= 'F')
        ):
            return False
    return True

# ask the user for original base, support only 2 or 16
print("enter the base of the number (binary - 2, hexadecimal - 16)")
base_input_str = input()

# check if base is not 2 or 16
if base_input_str != '2' and base_input_str != '16':
    print("invalid base. only 2 and 16 are supported")
else:
    base_input = int(base_input_str)

# ask the user for number in the base
print("enter the number in this base:")
number_str = input()

if number_str.strip() == "":
    print("error. no input was given.")
elif base_input == 2:
    if not is_valid_binary(number_str):
        print("error. binary must be only 0 and 1")
    else:
        number = int(number_str, 2)
        print("hexadecimal:", hex(number)[2:].upper())
elif base_input == 16:
    if not is_valid_hexadecimal(number_str):
        print("error. hexadecimal must be 0-9 or a-f")
    else:
        number = int(number_str, 16)
        print("binary:", bin(number)[2:])

```

המערכת המוצגת מיישמת פונקציונליות להמרת מספרים בין בסיס 2 ובסיס 16. תחילה, המערכת מבקשת מהמשתמש לציין את בסיס המספר המקורי, תוך הגבלת הקלט לבסיסים 2 ו-16 בלבד. לאחר קבלת הבסיס, המערכת מבקשת מהמשתמש להזין את המספר עצמו. המערכת מבצעת בדיקות תקינות על המספר שהוזן, מוודא שהוא תואם לבסיס שצוין (כלומר, מכיל רק ספרות 0 ו-1 עבור בסיס 2, או ספרות 0-9 ו-A-F עבור בסיס 16). לבסוף, המערכת ממירה את המספר לבסיס השני ומדפיסה את התוצאה.

סיכום ומסקנות:

במסגרת עבודה זו פיתחנו מערכת למחשבון להמרת בסיסים. הפרויקט התמקד ביצירת פתרון פשוט וקל להבנה שמאפשר המרת מספרים בין בסיסים שונים.

המערכת שלנו מסוגלת לקבל מספר בבסיס נתון, להמירו לבסיס אחר ולהציג את התוצאה בצורה מדויקת. שמנו דגש על כתיבת קוד נקי ותמציתי תוך שימוש בפונקציות מובנות של Python, מה שאפשר לנו ליצור פתרון יעיל. במהלך הפיתוח, חילקנו את המשימה למרכיבים קטנים וברורים. התמקדנו בכתיבת פונקציות ממוקדות שכל אחת מהן אחראית על משימה ספציפית, והקפדנו על בדיקות קלט מקיפות שיבטיחו טיפול נכון בכל תרחיש אפשרי. עבדנו כצוות, כאשר כל אחד מאיתנו תרם לפיתוח המערכת תוך שיתוף פעולה הדוק. הגישה שלנו הייתה לכתוב קוד פשוט ויעיל שממצה את הכלים של שימוש ב-Python תוך מתן תשומת לב לפרטים כמו טיפול בשגיאות ואימות קלט.

הפרויקט היווה עבורנו הזדמנות ללמוד ולהתנסות בעקרונות פיתוח תוכנה מעשיים. למדנו כיצד לפרק בעיה מורכבת לחלקים פשוטים יותר, כיצד לכתוב קוד איכותי, וכיצד לעבוד בצוות באופן יעיל. לסיכום, הצלחנו לייצר מחשבון להמרת בסיסים שעונה על הדרישות המקוריות בצורה פשוטה וברורה. הפרויקט אפשר לנו להעמיק את ההבנה שלנו בתכנות ב-Python ובעקרונות פיתוח תוכנה בסיסיים.

נספח 1- Readme

הוראות למשתמשים:

(מערכת המרת מספרים בין בסיסים)

- פתחו את הקובץ בסביבת העבודה של פייתון.

- בחרו את בסיס המוצא של המספר כעת:

* עבור המרה מבסיס בינארי הזינו 2.

* עבור המרה מבסיס הקסדצימלי הזינו 16.

- הזינו את המספר המקורי בהתאם לבסיס שנבחר.

- המערכת תבדוק את הקלט.

* במידה והקלט שהוזן תקין, יוצג המספר בבסיס הנבחר שנתמך על ידי המערכת.

* אם הוזן מספר לא תקין או שהשדה נשאר ריק, תופיע הודעת שגיאה ברורה.

בהצלחה!

קבוצה 20.