

מבוא למדעי המחשב – אביב 2023 – מטלה 2

נושאים: כתיבת תוכנית בשפת C.

משקל מציון התרגיל: 10%

תאריך הגשה: 23:50 24.04.2023

הגשה באיחור: ניתן להגיש באיחור של יום (בהורדת 10 נקודות – ציון מקסימלי 90), יומיים (בהורדת 20 נקודות – ציון מקסימלי 80) או שלושה ימים (בהורדת 30 נקודות – ציון מקסימלי 70). לאחר מכן לא תתאפשר ההגשה (מלבד לאיחורים מוצדקים לפי תקנון האוניברסיטה).

הנחיות כלליות

שאלות בנוגע לתרגיל יש לפרסם **באופן ציבורי** בפורום הייעודי למטלה הנמצא במודל.

בקשות להארכה מסיבות מוצדקות (מילואים, לידה, אשפוז וכו') יש לשלוח למייל tom.ben-dor@biu.ac.il בצירוף: שם מלא, שם משתמש במערכת ההגשה, מספר תעודת זהות ומסמכים רלוונטיים לפי הצורך.

יש להקפיד מאוד על הוראות עיצוב הקלט והפלט, בדיוק על פי הדוגמאות המצורפות. אין להוסיף או להשמיט רווחים או תווים אחרים, ואין להחליף אותיות גדולות בקטנות או להיפך. חוסר הקפדה על פרטים אלו עלול לגרור הורדה משמעותית ביותר בציון התרגיל עד כדי 0. ראו עצמכם הוזרתם!

שימו לב שאתם עוקבים במדויק אחרי ההנחיות במסמך ה-Style Guide המפורסם באתר הקורס.

עליכם לכתוב קוד על פי הוראות התרגיל ולוודא שקיבלתם 100 בבדיקה האוטומטית הראשונית, וכן שהתרגיל מתקמפל ורץ על שרתי המחלקה (planet) **ללא שגיאות וללא אזהרות**. תרגיל שלא עומד בסטנדרטים הבסיסיים הללו יגרור, בשל הטרחה שהוא מייצר בתהליך הבדיקה שלו הורדת נקודות משמעותית בציון שלו.

להזכיר העבודה היא אישית. "עבודה משותפת" דינה כהעתקה. העתקות נבדקות על ידי מערכת ההגשה האוטומטית, ותרגיל שהועתק יגרור בין השאר ציון 0 ופגיעה בציוני התרגול הסופיים **לכל הגורמים** השותפים בהעתקה. אתם יכולים לדון בגישות לפתרון התרגיל באופן תיאורטי, אך אין לשתף קוד בשום צורה.

בפיתוח הקוד ניתן להשתמש בכל סביבת עבודה, העיקר הוא שתדעו איך לקחת את קבצי הקוד מתוך הסביבה הזו, לבדוק אותם על שרתי האוניברסיטה, ולהגיש אותם באמצעות מערכת ההגשה.

בפיתוח הקוד אין להיעזר בכלים מונחי למידת מכונה מכל סוג, על כל קטע קוד להיכתב על ידי המגיש/ה בלבד ובאופן עצמאי.

דוגמת הרצה של התרגיל נמצאת בסוף המסמך.

תפריט

בתחילת התוכנית יודפס למסך התפריט הבא:

```
Choose an option:
0. Exit
1. Hourglass
2. Reverse bits
3. Reverse digits
4. Convert to Decimal
5. Base64 to Decimal
6. Swap bits
```

לאחר מכן, על פי בחירת המשתמש, תתבצע המשימה המבוקשת.
בסיום ביצוע משימה, יש להדפיס מחדש את התפריט ולפעול על פי בחירת המשתמש.
במידה והמשתמש מקיש 0, יש לסיים את התוכנית.

במידה והמשתמש מקיש אופציה שאינה מופיעה בתפריט, יש להדפיס **Invalid option ולהדפיס**
מחדש את התפריט.

ניתן להניח שהמשתמש יכניס תו בודד (יכול להיות כל תו) ולאחר מכן ילחץ Enter.

שימו לב: לאחר הדפסת התפריט יש לרדת שורה.

דוגמאות הרצה לתרגיל עם קלט ופלט מצופים מופיעים בסוף המסמך וכן מעודכנים במערכת הגשת התרגילים.

משימה 1 – ציור שעון חול

לאחר שהמשתמש בחר במשימה 1 – הוא יתבקש להכניס שני פרמטרים באופן הבא:

Enter size:

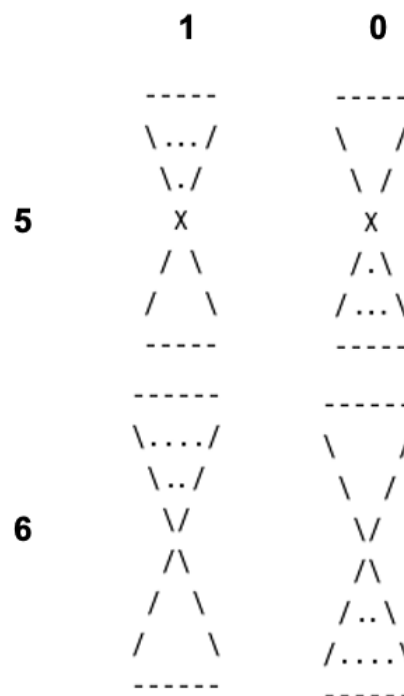
5

Enter flag:

1

באדום – קלט לדוגמה מהמשתמש (יכול להיות כל מספר *int*).

לאחר שהמשתמש הכניס את שני הפרמטרים שהתבקש, מודפס שעון חול עם גרגירי חול, בהתאם לפרמטרים. להלן טבלה של 4 הרצות שונות של ציור שעון החול (כל שעון חול הוא הדפסה שונה שהתבצעה בעקבות בקשה של המשתמש):



המשותף לשתי ההרצות בשורה העליונה הוא שהפרמטר *size* (גודל השעון) הוא 5.
המשותף לשתי ההרצות בשורה התחתונה הוא שהפרמטר *size* (גודל השעון) הוא 6.
המשותף לשתי ההרצות בטור השמאלי הוא שהפרמטר *flag* הוא 1 – משמע גרגירי החול בשעון עוד לא נפלו ולכן הם בחלק העליון של השעון.
המשותף לשתי ההרצות בטור הימני הוא שהפרמטר *flag* הוא 0 – משמע גרגירי החול בשעון נפלו ולכן הם בחלק התחתון של השעון.

כלומר, אם המשתמש הכניס:

```
Enter size:
5
Enter flag:
1
```

יודפס שעון החול משמאל למעלה בטבלה.

נקודות חשובות:

- 5 ו-6 הן רק דוגמאות – המשתמש יכול להכניס כל מספר שירצה ויודפס שעון מתאים.
- בסה"כ מודפסות $size + 2$ שורות: בהתחלה ובסוף שורה של $size$ מקפים, ביניהן שעון החול.
- מצד ימין לשעון **לא מודפסים** רווחים – התו האחרון בכל שורה הוא / או \ או -
- אם $size$ אי זוגי, לדוגמה 5 – בשורה האמצעית (מספר 3 במקרה של 5) של שעון החול מודפס X באמצע השעון.
- אם $size$ זוגי, לדוגמה 6 – בשורות האמצעיות (מספר 3 ו-4 במקרה של 6) מודפס:



- מובטח שהמשתמש יכניס מספרים שלמים כפרמטרים – לא מובטח שהם יהיו מתאימים ויש לבדוק זאת (לאחר קבלת שני המספרים).
אם המספרים לא עומדים בדרישות הבאות, יש להדפיס Invalid Parameters ולחזור לתפריט הראשי בלי להדפיס את שעון החול:
 - יש לבדוק ש- $size$ הוא לפחות 3.
 - יש לבדוק ש- $flag$ הוא 0 או 1.

משימה 2 – היפוך ביטים

לאחר שהשתמש בחר במשימה 2 – הוא יתבקש להכניס מספר *int* כלשהו:

Enter a number:
123456789

באדום – קלט לדוגמה מהשתמש (יכול להיות כל מספר *int*).

עליכם להדפיס את המספר העשירוני המתקבל על ידי היפוך של סדר הביטים המייצגים את המספר שקיבלתם.

נזכיר כי בתרגול ראינו שמשתנה מסוג *int* מיוצג באמצעות 4 בתים, או 32 ביטים. לדוגמה, המספר הנ"ל 123456789 הוא למעשה:

00000111010110111100110100010101

ואם נהפוך את סדר הביטים נקבל:

10101000101100111101101011100000

השקול למספר -1464608032 – בשיטת המשלים ל-2 עם 4 בתים.

הפלט יהיה בצורה הבאה:

Enter a number:
123456789
The reversed number is -1464608032

ניתן להניח שהמספר שהוזן על ידי המשתמש הינו מספר *int* תקין.

במשימה זו מותר להשתמש אך ורק בפעולות bitwise – אין להשתמש בכפל, חילוק, חיסור, חיבור, מודולו. לצורכי מונה ללולאה מותר להשתמש ב ++ פעם אחת בלבד.

משימה 3 - היפוך ספרות

לאחר שהמשתמש בחר במשימה 3 - הוא יתבקש להכניס מספר *int* כלשהו:

```
Enter a number:  
123456789
```

באדום - קלט לדוגמה מהמשתמש (יכול להיות כל מספר *int*).

עליכם להדפיס את המספר המתקבל על ידי היפוך של סדר הספרות במספר.

הפלט יהיה בצורה הבאה:

```
Enter a number:  
123456789  
The reversed number is 987654321
```

ניתן להניח שהמספר שהוזן על ידי המשתמש הינו מספר *int* תקין וחיובי, **אלו כל ההנחות המותרות**.
חשבו על מקרי קצה.

משימה 4 – המרה מבסיס כלשהו 9 – 2 לבסיס דצימלי

לאחר שהשתמש בחר במשימה 4 – הוא יתבקש להכניס שני פרמטרים באופן הבא:

```
Enter base:
2
Enter a number:
1010110101
```

באדום – קלט לדוגמה מהשתמש.

יש להדפיס כתשובה את ייצוג $number_{base}$ בבסיס 10.
לדוגמה:

```
Enter base:
2
Enter a number:
1010110101
The converted number is 693
```

נקודות חשובות:

- כאשר המשתמש מתבקש להכניס מספר, הוא יקליד טקסט באורך לא ידוע (אינסופי בפוטנציאל שלו) – המשתמש יקליד רצף תווים כלשהו ובסיומו יקליד Enter. זה לא חייב להיות מספר ובטח לא `int`.
- המספרים שמוכנסים הם `unsigned` (במקרה ומדובר בבסיס בינארי).
- מובטח שהתשובה תהיה בטווח של משתנה מסוג `int`.
- מובטח שהמשתמש יכניס בסיס שהוא מספר שלם – לא מובטח שהבסיס יהיה בטווח הנתמך 2 – 9 כולל, יש לבדוק זאת. במידה ומוכנס בסיס לא מתאים יש להדפיס (**מיד** לאחר קבלת הבסיס ולא לבקש את המספר) `Invalid Base` עם ירידת שורה ולחזור לתפריט הראשי.
- לאחר קבלת המספר, יש לבדוק האם הוא מספר חוקי לפי הבסיס הנתון. לדוגמה עבור בסיס 2 מספרים חוקיים הם מספרים המכילים **רק** אפסים ואחדות. אם יש תו לא חוקי יש להדפיס:

```
Invalid character %c in base %d
```

כאשר התו הלא חוקי והבסיס שהוכנס במקום האחוזים, ולחזור לתפריט הראשי.

משימה 5 – המרה מבסיס 64 לבסיס דצימלי

בדומה למשימה 4, במשימה זו המשתמש יתבקש להכניס מספר בבסיס 64 ויקבל את המספר בבסיס עשרוני:

```
Enter a number:  
1+2/3  
The converted number is 905670647
```

גם כאן יש להדפיס הודעת שגיאה במקרה בו המספר שהוכנס אינו מספר חוקי בבסיס 64. להזכירכם, בבסיס 64 מספרים חוקיים מכילים את התווים $0-9, a-z, A-Z, +, /$ בלבד.

```
Enter a number:  
Number!  
Invalid character ! in base 64
```

מכיוון שמספרים בבסיס 64 הם משמעותית ארוכים יותר בבסיס 10, לא ניתן להניח שהתוצאה תיכנס בתוך משתנה `int`, במקום נניח שהתוצאה תיכנס במשתנה `long` (כפול מגודל `int`).

משימה 6 – החלפת ביטים

לאחר שהשתמש בחר במשימה 6 – הוא יתבקש להכניס מספר *int* כלשהו ושני אינדקסים

```
Enter a number:  
34  
Enter indexes:  
5 4
```

עליכם להדפיס את המספר המתקבל על ידי החלפת הביט ה**חמישי** מימין בביט ה**שישי** מימין (כתלות בקלט). שימו לב שהספירה מתחילה כך שהביט הימני ביותר הוא אינדקס 0.

במקרה זה:

$$34_{10} = 100010_2$$

לאחר ההחלפה נקבל:

$$010010_2 = 18_{10}$$

לכן הפלט המצופה הוא:

```
Enter a number:  
34  
Enter indexes:  
5 4  
The result is 18
```

נקודות חשובות:

- סדר הכנסת האינדקסים לא משנה.
- המספר שהוכנס מובטח להיות מספר שלם וחיובי בגודל *int*. לכן, יצגו אותו בתור *unsigned int*. מה היתרונות של ייצוג זה?
- יש לבדוק האם האינדקסים חוקיים. אינדקסים חוקיים הם אינדקסים לא שליליים וגם בטווח התקין של מספר *int*, כפי שנלמד. במידה ואחד האינדקסים לא חוקיים יש להדפיס:

```
Invalid indexes
```

במשימה זו מותר להשתמש אך ורק בפעולות bitwise – אין להשתמש בכפל, חילוק, חיסור, חיבור, מודולו בכלל.

הוראות הגשה

יש להגיש את התוכנית שכתבתם בקובץ בודד בשם ex_2.c לקבוצה מספר 14.

דוגמת הרצה

כאמור, דוגמה זו היא **חלקית**, עליכם לחשוב על כלל מקרי הקצה האפשריים בתוכנית ולבדוק אותם.

```
Choose an option:
0. Exit
1. Hourglass
2. Reverse bits
3. Reverse digits
4. Convert to Decimal
5. Base64 to Decimal
6. Swap bits
7
Invalid option
Choose an option:
0. Exit
1. Hourglass
2. Reverse bits
3. Reverse digits
4. Convert to Decimal
5. Base64 to Decimal
6. Swap bits
1
Enter size:
5
Enter flag:
0
-----
\  /
 \ /
  X
 /.\
/... \
-----
Choose an option:
0. Exit
1. Hourglass
2. Reverse bits
3. Reverse digits
4. Convert to Decimal
5. Base64 to Decimal
6. Swap bits
1
Enter size:
5
Enter flag:
1
-----
\.../
 \./
  X
 /.\
/  \
-----
Choose an option:
```

```

0. Exit
1. Hourglass
2. Reverse bits
3. Reverse digits
4. Convert to Decimal
5. Base64 to Decimal
6. Swap bits
1
Enter size:
6
Enter flag:
0
-----
\      /
 \    /
  \  /
   \/
  /\
 /.. \
/....\
-----
Choose an option:
0. Exit
1. Hourglass
2. Reverse bits
3. Reverse digits
4. Convert to Decimal
5. Base64 to Decimal
6. Swap bits
1
Enter size:
6
Enter flag:
1
-----
\.... /
 \.. /
  \ /
   V
  /\
 /.. \
/....\
-----
Choose an option:
0. Exit
1. Hourglass
2. Reverse bits
3. Reverse digits
4. Convert to Decimal
5. Base64 to Decimal
6. Swap bits
2
Enter a number:
123456789
The reversed number is -1464608032
Choose an option:
0. Exit
1. Hourglass
2. Reverse bits
3. Reverse digits
4. Convert to Decimal
5. Base64 to Decimal
6. Swap bits
3
Enter a number:
1234
The reversed number is 4321

```

```
Choose an option:
0. Exit
1. Hourglass
2. Reverse bits
3. Reverse digits
4. Convert to Decimal
5. Base64 to Decimal
6. Swap bits
4
Enter base:
2
Enter a number:
111
The converted number is 7
Choose an option:
0. Exit
1. Hourglass
2. Reverse bits
3. Reverse digits
4. Convert to Decimal
5. Base64 to Decimal
6. Swap bits
4
Enter base:
3
Enter a number:
1234
Invalid character 3 in base 3
Choose an option:
0. Exit
1. Hourglass
2. Reverse bits
3. Reverse digits
4. Convert to Decimal
5. Base64 to Decimal
6. Swap bits
5
Enter a number:
1+2/3
The converted number is 905670647
Choose an option:
0. Exit
1. Hourglass
2. Reverse bits
3. Reverse digits
4. Convert to Decimal
5. Base64 to Decimal
6. Swap bits
6
Enter a number:
34
Enter indexes:
4 5
The result is 18
Choose an option:
0. Exit
1. Hourglass
2. Reverse bits
3. Reverse digits
4. Convert to Decimal
5. Base64 to Decimal
6. Swap bits
0
```

