

תרגיל 2

הנחיות הגשה

1. העבודה היא ביחידים. יש איסור חמור להגיש חומר שלא נכתב ויוצר ע"י הסטודנט. תשובות שלא נוצרו ונכתבו ע"י הסטודנט-- כולל חומרים מהרשת, עזרה (בתשלום או ללא תשלום) בהכנת התשובות וכו' הינן אסורות ונחשבות העתקה. התייעצות תוך ציון המקור (חברים, רשת, חומרי עזר וכו') אפשרית אולם רק לגבי נושאים קונספטואליים ולא קוד או פתרונות. במקרה של ספק, יש לקבל אישור מראש מהמרצה.
2. ההגשה היא עד ליום ראשון, בתאריך 15.11.2021 בשעה 23:30.
3. כל התוכנית תוגש בקובץ אחד. הגשת התרגיל תיעשה ע"י submit. עזרה ניתן למצוא באתר: <https://support.esc.biu.ac.il/submit/>

כדי למנוע בעיות, עדיף להגיש את הקובץ ישירות מחשבון הלינוקס שלכם.

שימו לב: העברת הקובץ דרך Windows יכולה לגרום לכך שלא יעבור קומפילציה. במקרה זה **הציון יהיה 0** ללא זכות לערעור.

4. יש לוודא שהתרגיל מתקמפל ורץ ללא שגיאות על גבי השרת planet

5. בשורה הראשונה (!) של הקובץ אותו אתם מגישים, יש לציין בהערה מס' ת.ז. ושם מלא, באופן הבא:

```
// id_number first_name last_name
```

לדוג':

```
// 123456789 Israela Israeli
```

6. בהצלחה

רקע כללי

בתרגיל זה נתרגל את האופן בו מידע מיוצג במחשב, וכן שימוש ראשוני באסמבלי.

מה צריך להגיש

מסדור קובץ **ex2.h** עם ה-headers לפונקציות בחלק הראשון ו-**makefile**. **עליכם להגיש שני קבצים בשם ex2.s, ex2.c** כש-**ex2.c** מכיל את החלק הראשון ו-**ex2.s** מכיל את החלק השני של התרגיל.

הקובץ **main1.c, main2.c** שמסדור, כולל קריאות לפונקציות שתממשו ב-**ex2.c**, וכן לקוד שתממשו ב-**ex2.s**. תוכלו לערוך קובץ זה כראות עיניכם כדי לבדוק את הקוד שלכם עם קלטים שונים (אינכם מגישים את **main1.c, main2.c**).

חלק ראשון - ייצוג

בחלק זה תממשו סוג חדש של שלמים, המייצג מספרים חיוביים ושלילים, המשתמשים בשיטת הסימן והגודל. קראנו לשיטה זו בקורס Magnitude. בשלמים מסוג זה MSB יהיה סיבית הסימן, ושאר הביטים ייצגו את הגודל של המספר – כך לדוגמא, המספר חמש שמוצג בבסיס בינארי $5 = 00000101$, ייצוג כשלילי בצורה הבאה: $-5 = 10000101$. גודל המשתנה החדש יהיה 4 בטים.

לצורך המימוש תגדירו typedef חדש, שיקרא magnitude ויגדיר את המשתנה החדש שיצרתם. לצורך השימוש ב magnitude עליכם לממש פונקציות חיבור, חיסור, כפל, שיוויון והשוואת גדלים (פונקציה אחת של > - ראו קובץ header) שיפעלו על המשתנה החדש.

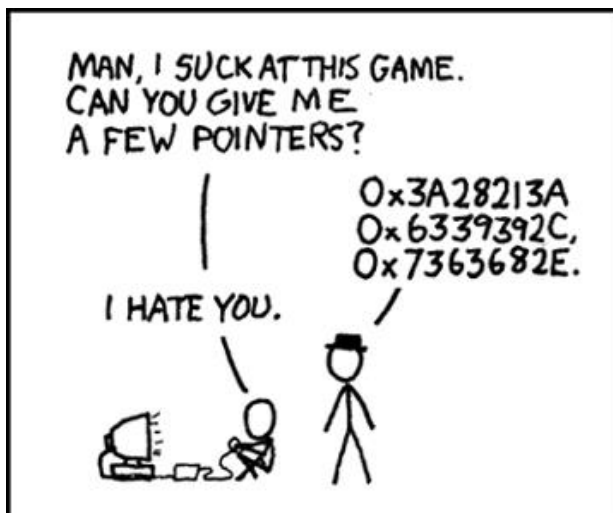
חלק שני – אסמבלי

כתבו את הקוד הבא בשפת אסמבלי.

```
int even (int num, int i){  
    return num << i;  
}  
  
int go (int A[10]) {  
    int sum = 0;  
    int i = 0;  
    while (i < 10) {  
        if (A[i] % 2 == 0) {  
            int num = even (A[i], i);  
            sum += num;  
        } else {  
            sum += A[i];  
        }  
        i++;  
    }  
    return sum;  
}
```

הערות

1. בזמן הבדיקה יוכנסו מספר קלטים שונים ויבדקו גם מקרי קצה, אך ניתן להניח כי כל הקלטים שיוכנסו יהיו בהתאם למה שהוגדר בתרגיל.
2. אין להשתמש בספריות חיצוניות.
3. תוכנית שלא תתקמפל ו/או, תיכשל בבדיקה תקבל ציון סופי 0 – לא תינתן זכות לערעורים על כך.
4. יש צורך בהערה **משמעותית** בתחילת הפתרון לכל משימה, ורצוי להוסיף הערות גם לקוד עצמו. מתכנת חיצוני (למשל, הבודק של הקורס) שמסתכל על הקוד שלכם צריך להבין בקלות את ההיגיון של הפתרון שלכם.
5. תוכלו לבדוק את קובץ האסמבלי עם הmain בעזרת הפקודה הבאה: gcc main2.c ex2.s -fno-pie



בהצלחה!