



פרויקט - חלק 3 - טסטים

תיאור המטריה

בנוסף לפרויקט, הנכם מדרשים להכין סט של 10 טסטים שבודקים אספקטים מסוימים בקומפיילר. הטעסים יבדקו את נכונות הקומפיילר שאתה בונים. המטרה היא להתנסות עם שפת התכנות לפני פיתוח הקומפיילר, ובמסוף שהוא לכם מה לבדוק את הפרויקט שלכם.

אנחנו ממליצים לכם לבנות את הטעסים כך שהם ידקו מגוון גדול בכל האפשר של יכולות הנטמכות בשפה ואף קטעי קוד שאינם מתקיפים עבור השפה הנתונה. המטרה היא לעזור לכם לבדוק את התרגיל המסכם שלכם לפניו שליחתו.

שימוש לבני בסיסים הגשת תרגילים אלה אנחנו **נשתחוו** בין כל הסטודנטים את הטעסים שתגישו, כך שיוצעו לכם מאגר שלם של טסטים לבדיקת התרגיל.

מבנה של טסט בודד

כל טסט יהיה מורכב מ לפחות 4 קבצים: קבצי הקוד, קובץ קלט וקובץ פלט (באוצר הקורס מצורפת דוגמה).

- **קובצי הקוד:** יכולים לפחות קובץ אחד עם פונקציית `main`.
- **קובץ הקלט:** יכול ערכים לפחות פקודות ה- `read` אם קיימות. ערך ראשון בשורה ראשונה של הקובץ, ערך שני בשורה שנייה של הקובץ, וכו'.
- **קובץ הפלט:** יכול את הפלט של פקודות ה- `write` ושל הסימולטור.
- **קובץ תוצאה:** קובץ ריק בשם `pass` או `fail`. אם קיימים קובץ `pass` נצפה שהטסט עבר בהצלחה, אם קיימים קובץ `fail` נצפה שהטסט ייפול על שגיאת קומפייזה (אין בדיקה של הودעת השגיאה).

הערה: במקרה שאין בטסט צורך בקובץ קלט ופלט, יש לצרף בכל מקרה את הקבצים `input`-`output` ריקים.

דוגמה

הדוגמה המתוארת תופיע גם באתר הקורס לעיונכם.

نبנה טסט שבודק פעולות אРИתמטיות של מספרים שלמים.

קובץ הטעס:

```
int test (a,b,c:int) {
    return ( a + b ) + b * c + (b + 10) / 2;
}
```

```
void main() {
    a,b,c,res:int;
    read(a);
    read(b);
    read(c);
    res = test(a,b,c);
    write(res);
}
```

קובץ הקלט:

```
1
2
3
```

הסבר: הערך בשורה הראשונה יהיה לפי סדר פקודות ה- `read`. כלומר בסופו של דבר `a=1, b=2, c=3`.

**קובץ הפלט:**

Input integer?:Input integer?:Input integer?:15 Reached Halt.

הסבר:

כל פקודה read שהסימולטור (wvm -rx) מרים גורמת להדפסה. קיימות שתי פקודות read אפשריות:
READI – generates "Input integer?:"
READF – generates "Input real?:"

תוצאת החישוב היא 15

$$\begin{aligned}
 (a + b) + b \cdot c + (b + 10) / 2 &= \\
 (1 + 2) + 2 \cdot 3 + (2 + 10) / 2 &= \\
 3 + 2 \cdot 3 + 12 / 2 &= \\
 3 + 6 + 6 &= \\
 15
 \end{aligned}$$

בסיום כל תוכנית, הסימולטור מדפיס: " Reached Halt ." ושורה ריקה לאחר מכן.

דוגמת שגיאה

הדוגמה המתוירת תופיע גם באתר הקורס לעונכם.

בבנה טסٹ שנכשל תחבירית.

קובץ הטסט:

```
void main() {
    a:int;
    read(a+"a");
}
```

הסבר:

תוכנית זו תיכשל עם הודעה של שגיאה תחבירית על שורה 3:

"Syntax error: '+' in line number 3"

זכור כי עליינו להוסיף קובץ ריק בשם fail לתיקיות הטסט כך שהבודק האוטומטי יידע שזיהוי מטרת הטסט. כפי שכבר הסבכנו, אין בדיקה של הودעת השגיאה בתרגיל זה (חלק 3 – טסיטים).

מבנה ההגשה

יש להגיש קובץ ארכיב מסוג Bzipped2-TAR בשם מהצורה (שורשו מספרי תז – 9 ספירות):
proj-part3-tests-<student1_id>-<student2_id>.tar.bz2
 בארכיב יש לכלול 10 תיקיות (תיקיה לכל טסט) באופן הבא:

```
-- test1:
- test.cmm
- input.in
- output.out
- pass/fail
-- test2:
- test.cmm
- another.cmm
- input.in
- output.out
```



```

- pass/fail
.
.
.
-- test10:
- test.cmm
- input.in
- output.out
- pass/fail

```

במילים, כל תקיה תכיל 4 קבצים לכל היותר לפי הצורך:

- קבצי הקוד: קבצים שנגמרים בסימנת cmm
- קובץ שיכיל את הקלט לטסט: in.txt
- קובץ שיכיל את הפלט לטסט: out.txt
- קובץ ריק להצהרת תוצאה הטסט: pass/fail

שימוש לב: קובץ הארכיב צריך להיות "שטווח" (כלומר, שלא יוצר ספריות משנה בעת הפיתחה אלא הקבצים ייווצרו בספרייה המוכנית). ניתן להשתמש בפקודה מסוימת:

tar cjf proj-part3-tests-012345678-123456789.tar.bz2 *

אוף הבדיקה

מסופק לכם קובץ הריצה בשם checker וקובץ בשם checker.rsk שמיועד לינקר. checker בודק את נכונות הטסט שכונתם, חשוב ש-rsk-runtimes.rsk יהיה באותה תקיה אותו בעת הריצה. נירז אותו כך:

```

./checker test.cmm [more_cmm_files.cmm] input.in output.out
(don't forget to adjust relative path of the files!)

```

הקובץ מנקף את קוד הטסט, מרייך את קובץ ההריצה כאשר קובץ הקלט נכנס בקלט, ולבסוף מatabase השוואת diff בין הפלט לקובץ הפלט שסופק.

לכל יש שלושה פלטים אפשריים:

- .1. True – טסט עבר בהצלחה.
- .2. False – פלט הטסט לא תואם את פלט הריצה.
- .3. Failed – שגיאת קומpileציה.

בבדיקה אוטומטית משתמש בכל זה כדי לבדוק שאכן כל הטסטים שלכם מקבלים פלט תקין. תרגיל עبور הוא תרגיל אשר **כל 10 הטסטים שלו תקין** (טסטים שאמורים לעבור צריכים לקבל true וטסטים שאמורים להישלם מקבלים failed).

הערה: אין להגיש מעל ל-10 טסטים.

הוראות הגשה

- מועד אחרון להגשתה: יום ה' 01/01/2026.
- שימוש-לב למידניות בנוגע לאחוריים בהגשתה המפורסמת באתר הקורס. במקרה של נסיבות המצדיקות אי-חזר, יש לפנות מראש לצוות הקורס ל蒂יאום דחית מועד ההגשתה.
- ההגשתה בזוגות. הגשה בבודדים תתקבל רק באישור מראש מצוות הקורס.
- יש להגיש בצורה מקוונת באמצעות אתר Moodle של הקורס, מחשבנו של אחד הסטודנטים. הקפידו לוודא כי העלהם את הגישה של ההגשתה אותה התכוונתם להגיש. לא יתקבלו טענות על אי-



התאמה בין הקובץ שנמצא ב-Moodle לבין הגרסה ש"התכוונתם" להגיש ולא יתקבלו הגשות מאוחרות במרקםם באלו.

בצלחה!