

תורת הקומפילציה 236360 סמסטר חורף תשע"ז

תרגיל בית מס 5 **תרגום לשפת ביניים, BP**

מתרגלת אחראית על התרגיל: מיכל בדיאן
sbadian@tx

הערות:

שאלות במייל נא להוסיף HW5 בנושא ההודעה (כולל מילואים).
לתרגיל ייפתח FAQ באתר הקורס.
הנחיות שיופיעו בסעיף זה עד יומיים לפי הגשת התרגיל מחייבות.
התרגיל ייבדק אוטומטית, **הקפידו למלא אחר ההוראות במדויק**.
כמו כן ייבדקו גם העתקות.
שימו לב שזמן ההגשה קרוב למועד מבחן הקורס, מומלץ להתחיל מוקדם.

כללי

בתרגיל זה עליכם לממש תרגום לשפת ביניים בשיטת Backpatching עבור שפת WEATHER שהכרתם בתרגיל בית מספר 4.

שפת הביניים

זיכרון

נשתמש במערך s לייצוג הזיכרון עבור משתני התוכנית ומשתנים זמניים. המערך הינו מטיפוס int וניתן לגשת אליו לקריאה או לכתיבה ע"י $s[i]$, כאשר i מספר שלם. שימו לב: המערך s מוגדר בסקריפט ההרצה של קוד בשפת ביניים bvm.pl (פירוט על הסקריפט יובא בהמשך), לכן אין צורך להגדיר אותו אלא רק לבצע בו שימוש ביצירת הקוד בשפת הביניים כמתואר להלן.

פקודות בשפת הביניים:

1. $s[i]=\text{value1 aop value2}$
2. $s[i]=\text{value}$
3. goto address
4. $\text{if value1 rop value2 goto address}$
5. print string
6. $\text{print } s[i]$
7. $\text{input } s[i]$

כאשר:

- value קבוע מספרי או כתובת בזיכרון ($s[i]$).
- address היא כתובת של פקודה.
- aop פעולה אריתמטית: חיבור, חיסור, כפל וחילוק (בשלמים).
- rop פעולת השוואה ($>$, $<$, $=$, $!=$).
- string מחרוזת מוקפת בגרשיים.

משמעות הפקודות:

1. ביצוע הפעולה האריתמטית ושמירת התוצאה בכתובת i במערך.
2. שמירת הערך value בכתובת i במערך.
3. קפיצה לכתובת address לבצוע הפקודה הבאה.
4. שיערוך התנאי הבוליאני וקפיצה ל- address אם מתקיים התנאי, אחרת ביצוע הפקודה הבאה.
5. הדפסת המחרוזת string .
6. הדפסת הערך המספרי בכתובת i .
7. קריאת ערך מספרי מהקלט ושמירתו בכתובת i .

כתובות

הכתובות של הפקודות מתחילות מ-0 וגדלות ב-1 בכל פקודה.

תחביר

תחביר השפה הוא כפי שהוצג בתרגיל בית 4.

חלק א – טיפול במשתנים זמניים

ממשו את הפונקציה `newTemp()` אשר מחזירה מחרוזת ייחודית בכל פעם שהיא נקראת. למשל באופן הבא:
בפעם הראשונה תחזיר "@t1", בפעם השנייה תחזיר "@t2", וכן הלאה.

חלק ב – הרחבת הסמנטיקה

יש לבצע את ההמרות הנדרשות ע"פ החוקים שהוגדרו בתרגיל בית 4, בחלק של בדיקת הטיפוסים.

ב1- ביטויים אריתמטיים והשמות

למשל, ההשמה:

`C myTempVar = 0 K;`

היא חוקית ונדרשת המרה בקוד הביניים, כך שבפועל `myTempVar` יקבל את הערך -273 (במעלות צלזיוס). ההמרה מתבצעת ע"פ הטיפוס של ה `LVALUE` בהשמה.

הערה: כל החישובים הם באריתמטיקה של שלמים (יתכן שבעת ביצוע המרות נאבד דיוק – אין צורך לטפל בסוגיה זו במסגרת תרגיל זה).

להלן טבלת ההמרות בין המידות השונות:

	From Celsius	To Celsius
Fahrenheit	$[^{\circ}\text{F}] = \frac{(^{\circ}\text{C}) \cdot 9}{5} + 32$	$[^{\circ}\text{C}] = \frac{(([^{\circ}\text{F}] - 32) \cdot 5)}{9}$
Kelvin	$[^{\circ}\text{K}] = [^{\circ}\text{C}] + 273$	$[^{\circ}\text{C}] = [^{\circ}\text{K}] - 273$

- יש להקפיד על ביצוע חישוב ההמרות ע"פ סדר פעולות חשבון המצוין לעיל (כולל סוגריים).
- כדי לבצע המרה מ Fahrenheit ל Kelvin, יש לבצע קודם המרה מ Fahrenheit ל Celsius ואח"כ מ Celsius ל Kelvin.
- באופן דומה הכיוון ההפוך (Kelvin ל Fahrenheit) צריך לעבור דרך Celsius.

הערות:

- בפעולות חילוק/חיבור/חיסור בין שני טיפוסים טמפ', אם הם לא מאותו טיפוס טמפ', יש לבצע המרה ע"פ טיפוס האופרנד השמאלי. לדוגמא, עבור הביטוי $1000C/300K$, יש לבצע את החישוב תוך המרת האופרנד הימני לטיפוס C. כנ"ל עבור פעולות השוואה בין שני טיפוסים טמפ'.
- בפקודות input אין המרות והקלט המצופה הוא ערך מספרי בלבד (ללא הטיפוס). למשל בפקודת "input x" עבור משתנה x מטיפוס C, אם בקלט מופיע הערך המספרי 50, אז x יקבל את הערך 50 מעלות צלזיוס.

ב2- ביטויים בוליאניים ומבני בקרה

1. ביטוי בוליאני יכול להופיע או בהשמה למשתנה מטיפוס BOOL, או כתנאי (של משפט if או לולאה). כדאי לבצע טיפול מתאים לכל מופע של ביטוי בוליאני ע"פ ההקשר שבו הוא מופיע. בתרגולים לא דנו בפריסת קוד עבור כללי גזירה בהם יש השמה למשתנה מטיפוס BOOL למשל. יש לחשוב על פריסת קוד מתאימה. בהשמה למשתנה בוליאני יש לשמור את הערך הבוליאני בזיכרון. פרט לכך (כאשר BOOL מופיע כתנאי) אין צורך לחשב ערכים בוליאניים לזיכרון אלא לתרגם אותם לסדרת קפיצות כפי שנלמד בתרגול.
2. ערכי המערך של שפת הביניים הם מטיפוס int. אנחנו נממש השמה בוליאנית ע"י הערכים 0 ו 1 בהתאמה. ערך משתנה 0 אם הביטוי הוא false ו-1 אם הביטוי הוא true.
3. משפט IF מהצורה:

```
if(x)
{ ....
//do something }
```

הוא חוקי כאשר x משתנה מטיפוס BOOL ויתבצע אם ערכו של x הוא true. התנאי הבוליאני עשוי להיות מורכב ולכלול למשל פעולות or/and/not בין משתנים בוליאניים ו/או ביטויי השוואה בוליאניים ו/או קבועים בוליאניים. כל זאת ע"פ כללי הגזירה של הדקדוק.

ב3- משפט Break

כאשר יש פקודת break, יש לקפוץ לסוף הלולאה שבתוכה מופיעה הפקודה. כלומר, לפקודה הבאה לאחר סיום הלולאה.

ב4- משפט Switch

את ערכי הקבועים המספריים שמופיעים במשפטי case יש להשוות לערך ה- Exp שב- switch. את ההשוואה יש לבצע ע"פ טיפוס Exp. כלומר, אם טיפוס הערך המספרי ב- case אינו זהה לטיפוס Exp, יש להמיר אותו

לטיפוס של Exp ואז לבצע את ההשוואה.
יש לבצע אך ורק את הפקודות השייכות ל case הראשון שעבורו יש התאמה (שוויון)
בין ערך הטמפ' של ה case לבין ערך הטמפ' של Exp.
אם לא קיים אף case מתאים, לא יתבצע דבר.

חלק ג – לולאת FOREACH מסוג λ

הוסיפו את האסימון עם התבנית : foreach FOREACH, ואת הכללים :
 $Stmt \rightarrow FOREACH LP ID TemperatureType RP OP Exp SC$
כאשר $OP \in \{PLUS, MINUS, MULT, DIV\}$
ניתן להניח ש Exp הוא rvalue.
בדיקות סמנטיות לכלל החדש :
המשתנה ID הוא int והוא הוגדר כבר בתוכנית (מופיע בטבלת הסמלים).
אם לא נעשתה לו עדיין השמה, הוא יקבל את הערך 2.
הקוד של $OP Exp$ יבוצע ID פעמים על כל המשתנים במערכת שהם טיפוסים מוצהרים מסוג $TemperatureType$, ללא קשר למיקום ההצהרה שלהם.
אם המשתנה רק הוצהר ולא קיבל ערך, נאתחל אותו ראשית ל0 ולאחר מכן נבצע את הקוד.
לדוגמא : $int x=3; FOREACH (x K) + 5$.
אז כל משתנה במערכת מסוג K יבצע 3 איטרציות של $id+=5$.

תשתית

הקבצים bp.hpp ו bp.cpp מממשים פעולות בסיסיות עבור backpatching.
אין חובה להשתמש בקבצים אלו.

BVM

הקובץ bvm.pl הוא קובץ perl שיועד להריץ תוכנית בשפת ביניים.
השימוש בקובץ הוא כלהלן :
 $bvm.pl <file>$
כאשר file הוא הקובץ שמכיל את התוכנית בשפת הביניים.
בדיקת התרגיל תתבצע בעזרת קובץ זה.

פלט המנתח

יש להדפיס את קוד הביניים (תוכן הbuffer) ל- **stdout** בסיום הניתוח.

טיפול בשגיאות

בתרגיל זה נתמקד בייצור קוד ביניים ולכן ניתן להניח שקלט הקומפיילר הוא תוכנית חוקית בשפת WEATHER (גם מבחינה לקסלית, גם מבחינה תחבירית וגם מבחינה סמנטית).

הדרכה והנחיות נוספות

1. מומלץ לפתור את התרגיל על פי סדר השלבים שבו הוא נכתב.
2. מומלץ להיעזר במבני נתונים של stl.
3. המחלקה ostream מאפשרת לשרשר מחרוזות בצורה נוחה.

הוראות הגשה

יש להגיש קובץ אחד בשם ID1-ID2.zip, על הקובץ להכיל:

- את כל הקבצים הדרושים לבניית הקומפיילר (כולל bp.cpp, bp.hpp אם עשיתם בהם שימוש).
- Makefile לבניית הקומפיילר.

סדרת הפעולות הבאה צריכה לייצר את הקובץ ex5.exe :

```
unzip ID1-ID2.zip  
make
```

בפרט יש לשים לב שלאחר ביצוע unzip מופיעים בתיקייה הנוכחית כל קבצי ההגשה ולא תיקייה נוספת אשר מכילה את קבצי ההגשה.

לאחר ביצוע make יש לבדוק שנוצר קובץ ההרצה וכי אין שגיאות קומפילציה. בדקו שזה אכן מתקיים עבור קבצי ההגשה שלכם, על השרת stud עליו תתבצע הבדיקה.

הגשות שלא יעמדו בדרישות לעיל יקבלו ציון 0 ללא אפשרות לבדיקה חוזרת !!!
בדקו היטב שההגשה שלכם עומדת בדרישות הבסיסיות הללו, לפני ההגשה.

בדיקה לפני הגשה

באתר הקורס פורסם קובץ טסט לדוגמא.
בדקו אם הקוד שאתם מייצרים בשפת הביניים תקין וכי הפלט של הרצתו הוא הפלט המצופה באופן הבא :

```
./ex5.exe < simple_test.txt > simple_IL.txt  
./bvm.pl simple_IL.txt > output.txt  
diff output.txt simple_test_output.txt
```

מומלץ להכין קבצי בדיקה נוספים שלכם על מנת לבדוק את נכונות המימוש עבור מבני הבקרה השונים וההמרות הנדרשות.

בהצלחה!