

הגשה דרך moodle. הגשה בזוגות (לא ביחידים).

מה צריך להגיש: קובץ zip הכולל את הקוד שאתם כתבתם ודוגמא לקלט עם הפלט המתאים. אין להגיש את הקובץ שיצר flex. שם קובץ ה- zip צריך להכיל את שמות המגשים. למשל JohnSmith\_MaryContrary\_exercise1.zip. רק אחד מהשותפים יגיש את התרגיל. יש לצרף בנוסף קובץ README עם שמות המגשים ומספרי הזהות שלהם.

התוכנה של flex (גרסה שמתאימה ל- windows) נמצאת בתיקה flex ב- moodle. (יש שם גם סיכום בעברית על flex ו- manual באנגלית).

**יש לכתוב בעזרת flex תוכנית שכותבת לפלט תאור של האסימונים שהיא מזהה בקלט שלה.**

הקלט כולל מידע על מקצועות ספורט שהופיעו באולימפיאדות והשנים בהם התחרו בכל מקצוע. הקלט יהיה בקובץ טקסט שינתן כ- command line argument לתוכנית שלכם. תאור מפורט יותר של הקלט עם דוגמא מופיע בהמשך.

פלט התוכנית תכתוב לפלט (ל- standard output) תאור של האסימונים שהיא מוצאת בקלט. עבור כל אסימון שמופיע בקלט יופיע בפלט שם האסימון וה- lexeme (המחרוזת שהופיעה בקלט שמהווה את האסימון) התאור של כל אסימון יופיע בשורה נפרדת.

התוכנית תשתמש במנתח לקסיקלי - הפונקציה yylex() -- שתכתב ע"י flex. yylex() תחזיר מספר המציין את סוג האסימון שמצאה.

#### סוגי אסימונים

סוג אסימון (token type)	מה מופיע בקלט
TITLE	Olympic Sports
SPORT	<sport>
YEARS	<years>
NAME	שם הספורט מוקף בגרשיים למשל "Archery"
YEAR_NUM	השנה למשל 2020. השנה המוקדמת ביותר שעשויה להופיע בקלט היא 1896
COMMA	פסיק
HYPHEN	מקף (hyphen) או המחרוזת to
SINCE	since
ALL	all

white space (רווחים, טאבים ו- newlines) לא יוגדרו כאסימונים (אם כי רווחים עשויים להופיע כחלק ממחרוזת שהיא אסימון מסוג NAME). אבל כל תו אחר שמופיע בקלט אמור להיות חלק מאסימון.

במקרה שמופיע בקלט תו שאינו חלק מאסימון חוקי יש לכתוב ל- standard error הודעת שגיאה המפרטת את השורה בה נפלה השגיאה ואת התו השגוי. לאחר מכן התוכנית תמשיך ותזהה אסימונים.

**שימו לב שכאן מתבקש לכתוב רק מנתח לקסיקלי. הסדר שבו מופיעים האסימונים אינו מענינו של המנתח הזה. ה- parser הוא שמוודא שאסימונים מופיעים בסדר הנכון (כלומר בהתאם לכללי הדקדוק) אבל התרגיל אינו כולל parser.**

### תאור הקלט

הקלט כולל רשימה של מקצועות ספורט. עבור כל מקצוע מופיע בקלט שם המקצוע ורשימה של השנים בהם התחרו בו.

בקלט יכול להופיע גם whitespace (רווחים, טאבים ו- newlines).

**הנה דוגמא לקלט (הנתונים נלקחו מ-**

[https://en.wikipedia.org/wiki/Summer\\_Olympic\\_Games](https://en.wikipedia.org/wiki/Summer_Olympic_Games)

Olympic Sports

<sport> "Archery" <years> 1900-1908, 1920, since 1972

<sport> "Athletics" <years> all

<sport> "BasketBall" <years> since 1936

<sport> "Tug Of War" <years> 1900 to 1920

<sport> "Karate" <years> 2020

עבור דוגמא זו הפלט (עבור שתי השורות הראשונות בקלט) יהיה:

TOKEN	LEXEME
TITLE	Olympic Sports
SPORT	<sport>
NAME	"Archery"
YEARS	<years>
YEAR_NUM	1900
HYPHEN	-

```

YEAR_NUM      1908
COMMA          ,
YEAR_NUM      1920
COMMA          ,
SINCE          since
YEAR_NUM      1972

```

**תזכורת: הכנת תוכנית בעזרת flex.**  
(ההערות מתייחסות ל-Windows ול-Linux)

נניח שברשותנו קובץ קלט ל-flex שהכנו בעזרת text editor (למשל Notepad++). נקרא לקובץ olympics.lex.

נריץ את הפקודות הבאות:

1. מריצים את flex

```
flex olympics.lex
```

נוצר קובץ lex.yy.c (הכולל את הפונקציה yylex())

הערה: כדי לקבל הודעות דיבוג כשמריצים את yylex אפשר להריץ את flex עם האופציה -d בצורה כזאת:

```
flex -d olympics.lex
```

2. יש לקמפל את קובץ ה-C ש-flex יצר עבורנו. (כמובן שאם התוכנית שלנו כוללת קבצים נוספים יש לקמפל גם אותם). לצורך כך ניתן להשתמש בכל קומפיילר לשפת C.

אם נשתמש בקומפיילר gcc (קומפיילר פופולרי של GNU) הפקודה היא:

```
gcc -o olympics.exe lex.yy.c
```

כאן האופציה -o מציינת את שם הקובץ שהוא התוצר של הקומפילציה (במקרה זה שם הקובץ הוא olympics.exe אם עובדים עם Linux אז נשמיט את הסיומת .exe).

3. נכין קובץ טקסט שנקרא לו test\_olympics.txt ובו נכתוב קלט לדוגמא למשל

```
Olympic Sports
<sport> "Archery" <years> 1900-1908, 1920, since 1972
```

. . .

נריץ את הפקודה

```
olympics.exe test_olympics.txt
```

ב-Linux הפקודה תהיה (בהנחה שקובץ ההרצה olympics נמצא בתיקיה הנוכחית):

```
./olympics test_olympics.txt
```

ובפלט תופיע רשימה של אסימונים :

TOKEN	LEXEME
TITLE	Olympic Sports
SPORT	<sport>
NAME	"Archery"
YEARS	<years>

. . .

### דוגמאות לתוכניות שהוכנו בעזרת flex

השאלה הראשונה בכל בחינה ישנה (ראו בתיקה "בחינות ישנות" ב-moodle) עוסקת ב-flex & bison. יש שם הרבה דוגמאות לקוד שמיועד ל-flex (התעלמו לעת עתה מהקוד שמיועד ל-bison).

יש גם בתיקה של bison ב-moodle דוגמאות לתוכניות שנכתבו בעזרת flex (ו-bison).

בתיקת ה-flex ב-moodle יש גם סיכום בעברית על flex (וגם manual באנגלית).

### הערות אחרונות :

אל תשכחו לכתוב את הפקודה הבאה בחלק הראשון של קובץ הקלט ל-flex שתכינו :

```
%option noyywrap
```

זה אומר ל-flex שאנו לא משתמשים בפונקציה yywrap (למי שמעונין, אפשר למצוא ב-manual הסבר על הפונקציה).

זכרו ש-flex "רגיש" לרווחים בקלט שלו. את כל הביטויים הרגולריים יש להצמיד לתחילת שורה. אם הביטוי הרגולרי שלכם צריך לכלול רווח אז יש להקיף אותו בגרשיים או בסוגריים מרובעות או לשים לפניו backslash וזה כדי שהרווח ייצג את התו רווח ולא יתפרש בטעות כסוף הביטוי הרגולרי. ניתן להפעיל את flex עם אופציה -d כדי לקבל הודעות דיבוג כש-yylex רצה. לדוגמא

```
flex -d olympics.lex
```

הודעות הדיבוג מפרטות (עבור כל התאמה לביטוי רגולרי) איזה ביטוי רגולרי הותאם ומה היתה המחרוזת בקלט שהתאימה לו.

**בהצלחה!**