מטלה מספר 2 – שפות תכנות והידור (FLEX)

במטלה זו נרשום Interpreter לשפה BrainFreak.

השפה מבצעת שימוש במערך (בגודל 2048 בתים) ומצביע (מאותחל בכתובת תחילת המערך) שבעזרתו ניתן לנוע על גבי המערך.

השפה מכילה 8 פקודות בלבד:

הפקודה	משמעות
>	קדם את המצביע לתא הבא
<	החזר את המצביע תא אחד לאחור
+	הוסף 1 לתא הנוכחי
-	החסר 1 מהתא הנוכחי
,	קרא התו / המספר הנוכחי ממחרוזת הקלט
.	כתוב את התו עפ"י קוד ה-ASCII של התא הנוכחי לפלט
[	דלג קדימה לפקודה שאחרי ה-[המתאים אם התא הנוכחי שווה ל-0
]	דלג אחורה לפקודה שאחרי ה-] המתאים אם התא הנוכחי שונה מ-0
רווח או \n	ה- Interpreter מתעלם ממנו
כל תו אחר	ה- Interpreter מתריע על שגיאת תחביר

למען הנוחות, לפניכם תרגום של פקודה ב- BrainFreak לפקודה ב- C:

שפת Brainfreak	שפת C
תחילת תוכנית	`char array[2048] = {0};` `char *ptr=array;`
>	`++ptr;`
<	`--ptr;`
+	`++*ptr;`
-	`--*ptr;`
,	`*ptr = getNum () / getChar();` (המספר / תו ייקלט מתוך מחרוזת הקלט)
.	`putchar(*ptr);`
[	`while (*ptr) {`
]	`}`

ה- Interpreter שתבנו יקבל תוכנית בשפת BrainFreak, יריץ אותה (בהתאם לפקודות שתוארו לעיל) ויציג את פלט התוכנית.

מספר הערות בנוגע למטלה:

- יש לבצע את הפקודות +, - כפקודה אחת במקרה והם מופיעים ברצף. משמע, בהינתן הקטע קוד ++++ הפקודה שתתבצע בפועל היא (*ptr)+4 (כמספר מופעי ה- "+") ולא הפעלת הפקודה (*ptr)++ ארבע פעמים. אותו הדבר בנוגע לפקודה "-".
- כאשר ישנה שגיאה (סוגי השגיאות יפורטו בהמשך) יש לעצור את ריצת ה- Interpreter ולהציג הודעה מתאימה.
  - כאשר הופיע תו לא תקין, יש להדפיס את ההודעה:

Unknown command

  - כאשר התבצעה פעולת –, אשר הובילה לכך שהערך בתא יהיה שלילי, יש להדפיס את ההודעה:

Invalid – command

  - כאשר התבצעה פעולת <, אשר הובילה לכך שתהיה חריגה מגבולות המערך, יש להדפיס את ההודעה:

Index Out Of Range

- פקודת הקלט (,) עובדת כך שהתו שמופיע אחרי הפקודה, יאוחסן בתא שעליו המצביע נמצא.
לדוגמא:

,H>,e>,l>,l>,o<<<<.>.>.>.>.

התוכנית לעיל מדפיסה : Hello

- או לחילופין פקודת הקלט (,) עובדת כך שהתו (או תווים) שמופיע אחרי הפקודה הינו מספר, אשר יאוחסן בתא שעליו המצביע נמצא.
לדוגמא:

,33>,H.<.

התוכנית לעיל מדפיסה: !H
מאחר והמספר ה- ASCII של '!' הינו 33.
נבחין כי התוכנית תחילה מאכסנת את הערך של !
לאחר מכן עוברת לתא הבא ומאכסנת בו את התו H.
מדפיסה את התו שמאוכסן בתא הנוכחי (שהוא H)
לאחר מכן חוזרת לתא הוקדם ומדפיסה את ה- !.

- **דוגמא לשימוש בלולאה**

לפנינו קטע קוד אשר מציג את כל הספרות מ- 1 ועד 9.

```
,9>,49<[>.+<-]
```

כדי להסביר את אופן פעולת הקוד, להלן **קוד** שמסביר את מה שקורה.

```
int arr[2048];
arr[0] = 9;
arr[1] = '1';
while(arr[0] != 0)
{
        printf("%c", arr[1]);
        arr[1]++;
        arr[0]--;
}
```

(קטע הקוד בפועל משנה את ערך המשתנה ptr וכך הוא יכול להגיע לתאים במערך).

- כאשר אתם מזהים את התו של תחילת לולאה (משמע התו '[') יש לעבור למצב אחר.

לדוגמא, אם קבעתם מצב

```
%s WHILESTATE
```

אז אני מצפה לראות בקוד שימוש ב:

```
BEGIN(WHILESTATE);
```

רק למען הסר ספק, אני לא רוצה שתאחסנו את **המחרוזת** שבין סימני הלולאה. כאשר אתם מזהים את תחילת הלולאה יש **לעבור למצב חדש**, ומשם להמשיך לפענח את הפקודות שכתובות בתוכה **רק באמצעות FLEX.**

- בפקודת ה- WHILE יש לממש עקרון של **לולאות מקוננות**.

לדוגמא:

```
,a>++[-<[->>+<<]>>+<]>.
```

התוכנית לעיל מדפיסה את הערך c.

- חובה לתת alias לכל ביטוי רגולרי שאתם מגדירים.
- חובה להשתמש ב- states (משמע להגדיר מצבים באמצעות %s או %x).

הנחיות כלליות

- המטלה צריכה לרוץ על שרתי ה- U2 של האוניברסיטה (משמע LINUX).
- הגשה דרך מערכת ההגשות
- יש לצרף בהגשה את הקבצים הבאים:
  - makefile
  - ex2.lex
- בראש קובץ ה- FLEX יש להוסיף הערה (שימוש ב: /* comment */) בה רשום:
  - שם הסטודנט
  - ת.ז.
  - מספר קבוצה
  - שם משתמש
- קובץ ההרצה הסופי צריך להיות קרוי a.out.