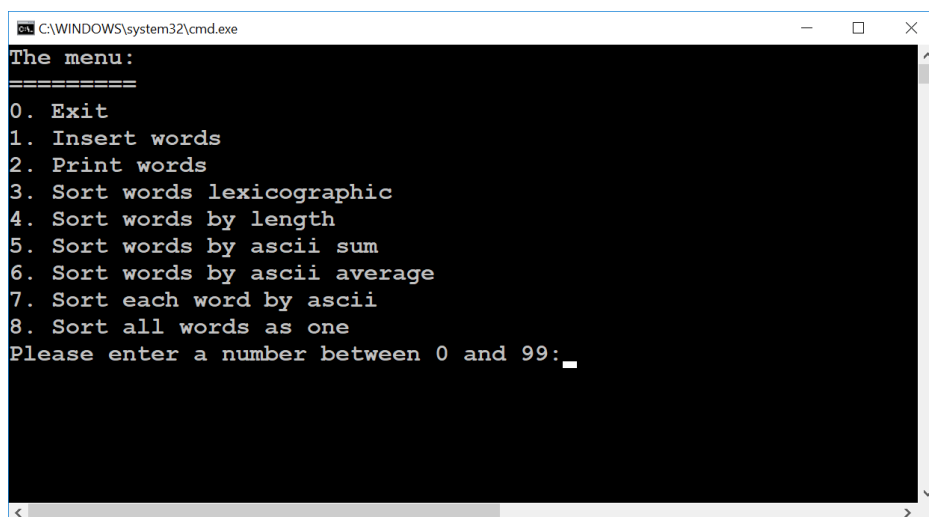


תרגיל מספר 1 בשפת C

תשפ"ג סמסטר א

עליכם לכתוב תכנית המציגה את התפריט הבא:



```
C:\WINDOWS\system32\cmd.exe
The menu:
=====
0. Exit
1. Insert words
2. Print words
3. Sort words lexicographic
4. Sort words by length
5. Sort words by ascii sum
6. Sort words by ascii average
7. Sort each word by ascii
8. Sort all words as one
Please enter a number between 0 and 99: _
```

גם בסיום כל סעיף (מלבד 0) יש להדפיס את התפריט.

סעיף 0:

יציאה מהתכנית תוך מתן הודעה למשתמש (נמצאת בקבצי הקלט פלט), כמו כן חשוב לשים לב כי כל מספר – חיובי ועד דו ספרתי - בתפריט יתקבל (כל מספר שאינו כזה יבקש שוב ושוב קלט מהמשתמש), אך כל מספר שהתקבל ושאינו אחד מהסעיפים יחשב כטעות, 5 טעויות מצטברות יסיימו את התכנית עם הודעת שגיאה מתאימה.

סעיף 1:

הזנת מילים, יש להזין את מספר המילים שיוזנו לתוכנה, מספר תקין הוא בין 2 ל 10, כל מספר לא תקין יבקש קלט שוב ושוב, לאחר הזנת מספר תקין N, המשתמש יתבקש להכניס N מחרוזות, על כל מחרוזת להיות בת מקסימום 20 תווים (כולל ה '\0'), ולהכיל רק אותיות (a-z או A-Z), כל נתון אחר לא יתקבל. ניתן להניח כי לא יהיה קלט של מעל 200 תווים.

סעיף 2:

יפעל רק אחרי שסעיף 1 הורץ לפחות פעם אחת – הדפסת כל המחרוזות הקיימות לפי סדרן, כאשר לפני כל אחת מפורטים מספר סידורי, סכום קודי ה ascii וממוצע קודי ה ascii של כל מחרוזת.

סעיף 3:

יפעל רק אחרי שסעיף 1 הורץ לפחות פעם אחת – מיון מחרוזות לקסיקוגרפי, המיון משנה את הסדר הקודם אך לא מדפיס.

סעיף 4:

יפעל רק אחרי שסעיף 1 הורץ לפחות פעם אחת – מיון כל המחרוזות הקיימות על פי גודלן בסדר עולה.

סעיף 5:

יפעל רק אחרי שסעיף 1 הורץ לפחות פעם אחת – מיון כל המחרוזות הקיימות על פי סכום קודי ה ascii של המחרוזת.

סעיף 6:

יפעל רק אחרי שסעיף 1 הורץ לפחות פעם אחת – מיון כל המחרוזות הקיימות על פי ממוצע קודי ה ascii של המחרוזת.

סעיף 7:

יפעל רק אחרי שסעיף 1 הורץ לפחות פעם אחת – מיון מחרוזת קיימת בפני עצמה כך שכל תו במחרוזת יהיה נמוך או שווה בערכו ה ascii מהתו שמימינו.

סעיף 8:

יפעל רק אחרי שסעיף 1 הורץ לפחות פעם אחת – מיון כל המחרוזות כאילו היו מחרוזת אחת ואחר כך הוחזרו למחרוזות המקוריות על פי גודלן, לדוגמא עבור המחרוזות הקיימות הבאות:

```
01:(ascii sum 486, ascii avg 81.00)AaAaAa
02:(ascii sum 394, ascii avg 78.80)BbBbB
03:(ascii sum 332, ascii avg 83.00)CcCc
The menu:
```

לאחר מיון זה יהפכו למחרוזות הבאות (כל אחת תישאר באותו הגודל והמיקום אבל התווים כאילו כל המחרוזות ברצף היו מחרוזת אחת והיא מוינה):

```
The words are:
=====
01:(ascii sum 393, ascii avg 65.50)AAABBB
02:(ascii sum 425, ascii avg 85.00)CCaaa
03:(ascii sum 394, ascii avg 98.50)bbcc
The menu:
=====
0. Exit
```

בעבודה זו חובה להשתמש **לפחות** בפונקציות הבאות, יש לעבוד עם חתימות מדויקות וניתן להעתיקן מדף הוראות זה:

```
void printMenu();
```

פונקציה המדפיסה את התפריט הראשי.

```
int checkString(char[]);
```

פונקציה המקבלת מחרוזת וקולטת מחרוזת מהמשתמש, אם המחרוזת שהתקבלה תקינה על פי הגדרות העבודה היא תועתק למחרוזת שהתקבלה ותחזיר 1, אחרת לא תועתק ותחזיר 0. מחרוזת אינה תקינה במקרה שהיא מכילה תו שאינו אות או יותר מ 20 תווים כולל ה '\0'.

```
void printWords(char[][20],int);
```

פונקציה המקבלת מערך מחרוזות ואת גודלו ומדפיסה את כל המילים הקיימות במערך כרגע, כל אחת בשורה משלה עם מספר סידורי המתחיל מ 1 וגם פרטים על סכום קודי ה ascii וממוצעם.

```
void insertStrings(char[][20],int);
```

פונקציה המקבלת מערך מחרוזות וגודל שיוגדר כ N ומאתחלת את N המחרוזות הראשונות במערך למילים תקינות על פי הגדרת התכנית.

```
int getSize(int,int);
```

פונקציה המקבלת ערך מינימלי ומקסימלי וקולטת מהמשתמש ערכים (שכל עוד הם אינם מספרים או אינם בטווח הם יקלטו שוב ושוב – תוך מתן הודעה מתאימה), מרגע שהוכנס ערך ראשון תקין ובטווח הפונקציה מחזירה אותו ללא הודעה, הטווח כולל את ערכי הקצה שנשלחו. המשתמש יכול להקליד כל קלט עד 200 תווים.

```
int asciiSum(char[]);
```

פונקציה המקבלת מחרוזת ומחזירה את סכום כל ערכי ה ascii של כל התווים במחרוזת.

```
double asciiAvg(char[]);
```

פונקציה המקבלת מחרוזת ומחזירה את ממוצע כל ערכי ה ascii של התווים במחרוזת.

```
void sortStringsLexicographic(char[][20],int);
```

פונקציה המקבלת מערך מחרוזות ואת גודלו וממיינת את המחרוזות מיון מילוני – לקסיקוגרפי.

```
void sortStringsByLength(char[][20],int);
```

פונקציה המקבלת מערך מחרוזות ואת גודלו וממיינת את המחרוזות מיון עולה לפי גודלן.

```
void sortStringsByAsciiAvg(char[][20],int);
```

פונקציה המקבלת מערך מחרוזות ואת גודלו וממיינת את המחרוזות מיון עולה לפי ממוצע ערכי ה ascii של כל מחרוזת.

```
void sortStringsByAsciiSum(char[][20],int);
```

פונקציה המקבלת מערך מחרוזות ואת גודלו וממיינת את המחרוזות מיון עולה לפי סכום ערכי ה ascii של כל מחרוזת.

```
void sortEachString(char[][20],int);
```

פונקציה המקבלת מערך מחרוזות ואת גודלו וממיינת כל מחרוזת בו (בלי קשר לאחרות) לפי תווי ה ascii כך שמימין לכל תו יהיה תו עם קוד ascii גדול או שווה לו.

```
void sortString(char[]);
```

פונקציה המקבלת מחרוזת וממיינת אותה לפי תווי ה ascii כך שמימין לכל תו יהיה תו עם קוד ascii גדול או שווה לו.

```
void sortAllAsOne(char[][20],int);
```

פונקציה המתייחסת למערך כל המחרוזות כאל מחרוזת משורשרת אחת גדולה ושמה, ממיינת אותו לפי קודי ה ascii ומחזירה למחרוזות המקוריות לפי הגדלים המקוריים.

הערות:

1. ניתן להניח שכל קלט מסתיים בלחיצת enter.
2. לא ניתן להניח קלטים מספרים תקינים, יש לבדוק זאת.
3. קלט מספרי תקין נחשב קלט עם ספרות רצופות, רווחים וטאבים, כאשר הרווחים והטאבים נמצאים או בתחילת או בסוף הקלט (או בשניהם) ברצף והמספר צריך להיות בטווח המבוקש, כל קלט אחר יחשב כלא תקין.
4. ניתן לקבל הערות קומפילציה על פקודות: scanf, strlen, strcpy, strcat, strcmp, gets.
5. ניתן לקבל הערות קומפילציה על (1) while וגם על single line comment.
6. לא ניתן לקבל הערות קומפילציה על שום פקודה או סיטואציה אחרת.
7. ניתן להשתמש בחומר הנלמד עד לפרסום העבודה ורק בחומר זה, ובפירוט: תנאים, לולאות, מערכים, מחרוזות (הפונקציות שנלמדו בלבד) ומערכים דו ממדיים.
8. בקלט המילים והמספרים ניתן להניח כי המשתמש לא מזין יותר מ 200 תווים בקלט.
9. שימו לב להערות על כל סעיף וסעיף, על הפלט להיות זהה לחלוטין לזה של קובץ הריצה.
10. בכל מקום בו מזהה חריגה בין ההסברים לבין קובץ הריצה יש לידע אותי במייל או בפורום, ובכל מקרה יש לנהוג על פי קובץ הריצה.
11. בכל המיונים שני נתונים בעלי ערך מיון זהה ישמרו על הסדר המקורי שלהם גם לאחר המיון (חישבו אלו מיונים שלמדתם מתנהגים כך).
12. חובה להשתמש בכל הפונקציות שהוגדרו שימוש לא פיקטיבי, ניתן להוסיף פונקציות משלכם אך אין לגרוע ממה שהוגדר.
13. הגשה בזוגות, יש לכתוב מספרי תעודות זהות בהערה בתוך קובץ ההגשה.
14. שם הקובץ ex1.c
15. יש לוודא שהתוכנית עובדת באופן תקין גם בקומפיילר gcc.

בהצלחה