

1. כתבו פונקציה המקבלת מחרוזת לא-ריקה, ומחזירה את התו האחרון שלה.

2. כתבו תכנית הקולטת 15 מחרוזות, ומציגה כפלט את אלו מהן שהתו הראשון שלהן שווה לתו האחרון שלהן (היעזרו בפונקציה שכתבתם).

3. לפניכם תכנית חלקית:

```
#include <stdio.h>
int main()
{
    char st[100];
    ...
    gets(st);
    ...
    return 0;
}
```

התכנית מכניסה לתוך המחרוזת st קלט שיכול להכיל

רק את האותיות A..Z ואת התו רווח.

במחרוזת יש מילים המורכבות מאותיות בלבד,

ואחרי כל מילה (כולל האחרונה) יש רווח אחד בלבד.

התו הראשון במחרוזת הוא אות.

השלימו את התכנית החלקית, כך שתמנה ותציג כפלט את מספר המילים המסתיימות באות Y.

לדוגמא, בעבור הקלט "HAPPY BIRTHDAY TO YOU", יוצג הפלט 2.

4. כתבו תכנית הקולטת שתי מחרוזות. אם אורך המחרוזת הארוכה היא לפחות פי שניים מאורך המחרוזת הקצרה, תודפס המחרוזת הארוכה, אחרת תודפס המחרוזת הקצרה.

5. כתבו תכנית הקולטת שם מלא (פרטי ומשפחה) מהמשתמש, ומדפיסה את אורכו. השם הפרטי ושם המשפחה מופיעים בשורות, נפרדות של הקלט, ואסור שהשם המלא יכיל יותר מ-12 אותיות. במידה והוא ארוך מדי, תוצג הודעת שגיאה.

6. נתונה כותרת הפונקציה (prototype) הבאה:

void reverse (char str[])

הפונקציה מקבלת מחרוזת s, והופכת אותה ללא שימוש במערך עזר.

לדוגמא, אם ערכה של המחרוזת str הוא "abc",

אז לאחר הזימון reverse(str), יהיה ערכה "cba".

השלימו את גוף הפונקציה.

7. כתבו פונקציה שכותרתה

unsigned count_occurrences (const char s1[], const char s2[])

הפונקציה תחזיר את מספר הפעמים שהמחרוזת s2 מופיעה בשלמותה ב-s1.

לדוגמא, אם s1 = "ababaabbbabab", s2 = "aba", אזי הפונקציה תחזיר 3.

הערה: מותר שיהיו חפיפות בין ההופעות. למשל aba מופיעה פעמיים במחרוזת ababab.

8. כתוב פונקציה המקבלת מחרוזת המכילה אותיות ומספרים ומחזירה את סכום המספרים שהופיעו במחרוזת. דוגמא: עבור המחרוזת ab12d580cw600, תחזיר הפונקציה 1192 (12+580+600).

9. כתבו פונקציה בעלות המפרט הבא:

```
void delete_char(char src[], char c);
```

הפונקציה מקבלת מחרוזת ותו מפריד. היא מוחקת מהמחרוזת את כל ההופעות של התו הזה (כלומר המחרוזת תצטופף על חשבון ההופעות של התו הזה, ראו דוגמא למטה).
הדרכה: ניתן להגדיר בפונקציה מערך-עזר. הניחו כי הקלט חוקי וכי אורך המחרוזת המקסימלי הוא 200 תווים.

דוגמא: בעקבות קטע-התוכנית הבא:

```
char src[] = "e-r-e-t-z-n-e-h-e-d-e-r-e-t", sep='-';  
delete_char(src, sep);  
printf("%s\n", src);
```

יודפס:
eretznehederet

10. צופן קיסר

צופן (cipher) הוא אלגוריתם הצפנה, המקבל טקסט קריא ומפתח - ומחזיר טקסט מוצפן. צופן קיסר מבוסס על רעיון החלפת האותיות של הטקסט הקריא לשם יצירתו של הטקסט המוצפן:

האלפבית המשמש להצפנה מוסט מעגלית במספר קבוע של 'מקומות' מן האלפבית הרגיל.

לפי עדויות היסטוריות יוליוס קיסר עשה בשיטה זו שימוש נרחב.

המפתח (key) = מספר מקומות ההסטה.

למשל, במפתח של 3 מקומות המילה BABY תתורגם למילה EDEB. שימו לב שהאות Y הפכה ל-B בעקבות המעגליות.

כתוב פעולה המקבלת מחרוזת ומפתח. הפונקציה תדפיס את המילה המוצפנת.

שאלה 11

עליכם לקלוט מחרוזת ע"

```
scanf("%s", string);
```

כתבו תוכנית המחשבת את "הגימטרייה" של כל מילה במשפט. ניתן להניח כי המילים מופרדות ע"י נקודה יחידה ('.')

איך לחשב את "הגימטרייה": נחשב את הסכום של כל ערך ASCII של כל אות במילה.

דוגמה:

עבור:

Please enter a sentence:

The.quick.brown.fox.jumps.over.the.lazy.dog

נקבל:

```
The geometria of word #0 is: 289
The geometria of word #1 is: 541
The geometria of word #2 is: 552
The geometria of word #3 is: 333
The geometria of word #4 is: 559
The geometria of word #5 is: 444
The geometria of word #6 is: 321
The geometria of word #7 is: 448
The geometria of word #8 is: 314
```

הסבר:

T = 84

h = 104

e = 101

The = 84 + 104 + 101 = 289

שאלה 12

נתונה מחרוזת המכילה מילים המופרדות ברווח יחיד. כתבו פונקציה `int longest_word(char s[])` המקבלת מחרוזת של מילים ומחזירה את אורך המילה הארוכה ביותר.

ניתן להניח כי בין כל 2 מילים יש רווח יחיד, וכי המחרוזת חוקית (מסתיימת ב – '\0') וכי אין רווחים מיותרים בתחילת ובסוף המחרוזת.

למשל עבור המחרוזת

G	O	O	D		B	Y	E	\0
---	---	---	---	--	---	---	---	----

הפונקציה תחזיר 4, כי המילה הארוכה ביותר היא הראשונה ואורכה 4.

הערות חשובות

1. אסור לשנות את המחרוזת.
2. אין להשתמש ברקורסיה כלל.

סעיף א:

כתבו פונקציה: `int does_create(char str[], char sub_str[])` המקבלת מחרוזת חוקית `str` וכן מחרוזת נוספת `sub_str`.

הפונקציה תחזיר 1 אם שרשור של מערך התווים `sub_str` בונה את המחרוזת `str` ו-0 אחרת.

לדוגמא:

עבור המחרוזת `s "abcabcabcabc"`, מערך התווים `abc` הפונקציה תחזיר 1 כי שרשור `abc` 4 פעמים בונה את המחרוזת.

עבור המחרוזת `s "abcabcabcab"`, מערך התווים `abc` הפונקציה תחזיר 0 כי כל שרשור של `abc` לא בונה את המחרוזת.

אפשר להניח כי הקלט מורכב רק מאותיות ומספרים.

הערה חשובה: ניתן להשתמש בפונקציות ספריה באופן חופשי אך מומלץ מאוד לממשן בעצמכם. כמובן שכל פונקציה אותה מימשתם יש לבדוק אותה בנפרד לפני שימוש בתוכנית אחרת.

סעיף ב:

רישא של מחרוזת תווים היא מערך תווים הנמצאת בתחילתה של המחרוזת.

למשל: למחרוזת `"abca"` ישנם 4 רישות שונות, והן:

a – התו הראשון

ab – שני התווים הראשונים

abc – שלושת התווים הראשונים

abca – כל אברי המחרוזת.

כתבו פונקציה `int shortest_str(char s[])` המקבלת מחרוזת חוקית לא ריקה `s` ומחזירה את אורכה של הרישא הקצרה ביותר אשר שרשורה (ראה סעיף א) בונה את המחרוזת. אפשר להניח כי הקלט מורכב רק מאותיות ומספרים.

לדוגמא: עבור `s "abcabc"` הפונקציה תחזיר 3 כי `abc` היא הרישא הקצרה ביותר הבונה את המחרוזת. עבור `s "abcab"` הפונקציה תחזיר 5 כי `abcab` היא הרישא הקצרה ביותר הבונה את המחרוזת.

```
int shortest_str(char s[])
{
    /* implement this function */
}
```