

מחרוזות

- הגדרת מחרוזת.
- פונקציות ספרייה למחרוזות.

מחרוזות

- ראינו שניתן לשמור תו בודד במשתנה מסוג char.
- אבל תו בודד לכשעצמו בדר"כ לא מועיל, אנחנו בדר"כ מתעניינים באוסף של תווים שביחד יש להם משמעות – שם, שם משפחה, כתובת וכו'.
- אוסף של תווים נקרא **מחרוזת** (String).
- שפת C מייצגת מחרוזת באמצעות מערך של תווים:

```
char str[10];
```

קלט מחרוזות

- אפשר לקלוט מחרוזת מהמשתמש ע"י פקודת scanf ושימוש בסימן %s

```
char str[10];
```

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

- שימו לב לשני דברים חשובים:

– בקלט המחרוזת אין צורך להשתמש ב-& כמו בכל שאר המשתנים (נראה למה בהמשך הקורס).

– הפקודה קולטת רק רצף של תווים שאינם רווח. למשל, אם המשתמש יקליד Hello World, רק המחרוזת Hello תקלט.

נניח שהוגדרה המחרוזת:

```
char str[10];
```

והמשתמש הקליד את המילה "Hello". כך יראה הזיכרון לאחר הקלט:

H	e	l	l	o	\0				
0	1	2	3	4	5	6	7	8	9

תווי המחרוזת נשמרים החל מתא 0 במערך, אחד אחרי השני. אחרי התו האחרון, מתווסף באופן אוטומטי תו מיוחד שנקרא \0 והוא מסמן את סוף המחרוזת.

למרות שבמערך יש עוד תאים פנויים, המחרוזת מסתיימת בתו \0, כל מה שמעבר לתו זה לא נחשב חלק מהמחרוזת.

נניח שנתון הקוד הבא:

```
char str[3];  
scanf("%s", str);
```

ונניח שהמשתמש הקליד את המחרוזת "abc". מה לדעתכם יקרה בזיכרון?

המחרוזת מוגדרת בגודל 3, כלומר יכולה להכיל שלושה תווים. המשתמש מזין שלושה תווים, אולם בסוף הקלט מתווסף אוטומטית התו \0. כיוון שהתווים abc תופסים את שלושת תאי המערך, התו \0 יתווסף אחרי התא האחרון, כלומר בזיכרון שלא שייך למערך.

לכן חשוב להגדיר מערך מספיק גדול כדי שיוכל להכיל מחרוזות באורכים משתנים!

הדפסת מחרוזת

- ניתן להדפיס מחרוזת ע"י הפקודה printf והסימן `%.s`.
- המחרוזת תודפס החל מתא 0 של המערך ועד ה-`\0` (לא כולל):

```
char s[20];  
scanf("%.s", s);  
printf("%.s", s);
```

אתחול מחרוזת

- ניתן לאתחל מחרוזת בשורת ההגדרה בצורה כזאת:

```
char s[20] = "My Name";
```

- שימו לב לשימוש בגרשיים. כמו כן שימו לב שהמערך מוגדר בגודל 20, אבל המחרוזת מכילה רק 7 תווים שיאוחסנו בתאים 0 עד 6. התו \0 ישתל באופן אוטומטי בתא מספר 7.

כיוון שמחרוזת היא בעצם מערך, ניתן לעבור על התווים שלה בלולאה.

נכתוב פונקציה שמקבלת כפרמטר מחרוזת ומחזירה כמה אותיות (גדולות וקטנות) נמצאות במחרוזת:

```
int countLetters(char s[])
{
    int count = 0, i;
    for(i = 0; s[i] != '\0'; i++)
        if(isLetter(s[i]))
            count++;
    return count;
}
```


כיוון שמחרוזת היא בעצם מערך, ניתן לעבור על התווים שלה בלולאה.

נכתוב פונקציה שמקבלת כפרמטר מחרוזת ומחזירה כמה אותיות (גדולות וקטנות) נמצאות במחרוזת:

```
int countLetters(char s[])
{
    int count = 0, i;
    for(i = 0; s[i] != '\0'; i++)
        if(isLetter(s[i]))
            count++;
    return count;
}
```

* את הפונקציה isLetter ראינו בשיעור על תווים – הפונקציה מקבלת תו ומחזירה TRUE אם התו הוא אות.

פונקציות ספרייה


- מחרוזות הן מבנה נתונים חשוב ושימושי, ולכן פעולות רבות על מחרוזות נכתבו כבר מראש בתור פונקציות ספרייה, לשימוש המתכנתים.
- כדי להשתמש בפונקציות האילו יש להוסיף את הספרייה `string.h` לתוכנית, כך –

```
#include <string.h>
```
- להלן כמה פונקציות שימושיות מהספרייה:

int strlen(char s[])

הפונקציה strlen מקבלת מחרוזת ומחזירה את אורכה (מספר התווים שבה, לא כולל \0):

```
#include <stdio.h>
#include <string.h>

int main()
{
    char s[20] = "Hello";
    int x;
    x = strlen(s);
    printf("%d", x);  5
}
```


נשנה את הפונקציה שסופרת אותיות במחרוזת, כך שתשתמש
בפונקציה strlen:

```
int countLetters(char s[])  
{  
    int count = 0, i;  
    for(i = 0; i < strlen(s); i++)  
        if(isLetter(s[i]))  
            count++;  
    return count;  
}
```

```
void strcpy(char t[], char s[])
```

הפונקציה מעתיקה לתחילת המחרוזת t את המחרוזת s:

```
#include <stdio.h>
#include <string.h>
```

```
int main()
{
    char s[20] = "Hello", t[20];
    strcpy(t, s);
    printf("%s", t);  Hello
}
```

int strcmp(char t[], char s[])

הפונקציה משווה בין שתי המחרוזות s ו-t. אם שתיהן שוות, היא תחזיר 0. אם t יותר קטנה, היא תחזיר -1 ואם t יותר גדולה, היא תחזיר 1. יחס הגודל בין המחרוזות הוא לפי סדר לקסיקוגרפי (מילוני).

```
#include <stdio.h>
#include <string.h>
```

```
int main()
{
    char s[20] = "Hello", t[20] = "Hello";
    if(!strcmp(t, s))
        printf("Equal");
}
```


int strcmp(char t[], char s[])

הפונקציה משווה בין שתי המחרוזות s ו-t. אם שתיהן שוות, היא תחזיר 0. אם t יותר קטנה, היא תחזיר -1 ואם t יותר גדולה, היא תחזיר 1. יחס הגודל בין המחרוזות הוא לפי סדר לקסיקוגרפי (מילוני).

```
#include <stdio.h>
```

```
#include <string.h>
```

```
int main()
```

```
{  
    char s[20] = "Hello", t[20] = "Door";  
    int x = strcmp(t, s);  
    printf("%d", x);  -1  
}
```

פונקציות נוספות

- הספרייה string.h מכילה עוד הרבה פונקציות שימושיות שמאפשרות ביצוע מניפולציות וקבלת מידע רלבנטי על מחרוזות.
- כדאי לקרוא את תיעוד הספרייה כדי להכיר את היכולות המובנות לעבודה עם מחרוזות.

תרגיל

- כתבו פונקציה שחתימתה –

```
int equalIgnoreCase(char s1[], char s2[])
```

- הפונקציה תקבל שתי מחרוזות ותחזיר TRUE אם הן שוות, עד כדי גודל האות. כלומר, לא יהיה הבדל בין אותיות גדולות לקטנות.
- למשל, עבור שתי המחרוזות $s1 = \text{"AbCd"}$ ו- $s2 = \text{"aBcD"}$ הפונקציה תחזיר TRUE.