

# מחרוזות



קרון כליף

# ביחידה זו נלמד:

הספריה `string.h` □

■ `strlen`

■ `strcpy`

■ `strcmp`

■ `strcat`

מערך של מחרוזות □

הפונקציות: □

□ `sprintf`

□ `sscanf`

הפקודות: □

■ `getchar`

■ `putchar`

■ `getch`

■ `getche`

מהי מחרוזת □

איתחול מחרוזת □

הפקודות: □

□ `gets`

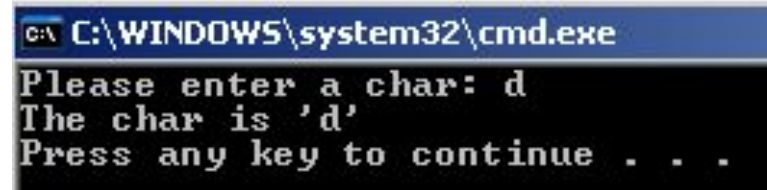
□ `puts`

# הפקודות getchar ו- putchar

- הפקודה getchar קולטת תו מהמקלדת
- הפקודה putchar מדפיסה תו למסך

```
#include <stdio.h>
void main()
{
    char ch;

    printf("Please enter a char: ");
    ch = getchar();
    printf("The char is: ");
    putchar(ch);
    printf("\n");
}
```



```
C:\WINDOWS\system32\cmd.exe
Please enter a char: d
The char is 'd'
Press any key to continue . . .
```

```
scanf("%c", &ch);
printf("The char is: '%c'\n",
      ch);
```

# הפקודות getch ו- getche

□ שתי פקודות אלו קוראות נתון מהמקלדת ומיד מחזירות את השליטה לתוכנית, כלומר:

■ לא אוגרות נתון בתוך ה- buffer / לא מחכות ל- ENTER

□ ההבדל ביניהן:

■ getch לא מציגה את התו המוקלד למסך

■ getche כן מציגה את התו המוקלד למסך

□ כדי להשתמש בהן יש לבצע:

```
<include <conio.h>
```

# הפקודות getch ו- getche - דוגמא

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

```
void main()
{
```

```
    char ch;
```

```
    printf("Please enter a char: ");
```

```
    ch = getche();
```

```
    printf("\nThe printed char is %c\n", ch);
```

```
    printf("Please enter a char: ");
```

```
    ch = getch();
```

```
    printf("\nThe printed char is %c\n", ch);
```

```
}
```

```
Please enter a char: k
The printed char is k
Please enter a char:
The printed char is e
Press any key to continue .
```

רואים את התו שהוקלד על המסך.  
מיד לאחר ההקלדה השליטה חזרה

לא רואים את התו שהוקלד על המסך.  
מיד לאחר ההקלדה השליטה חזרה לתוכנית.

# מהי מחרוזת

- מחרוזת הינה טיפוס שנועד לאחסן מילים שלמות
- מחרוזת היא מקרה פרטי של מערך מטיפוס char
- אבל התו האחרון במחרוזת תמיד יהיה `'\0'` (קוד ה-ASCII שלו הוא 0)

- לכן מחרוזת גם נקראית Null Terminated String
- מערך של תווים:

```
char name[5] = {'h', 'e', 'l', 'l', 'o'};
```

- מחרוזת:

```
char name[6] = {'h', 'e', 'l', 'l', 'o', '\0'};
```

- העובדה שמחרוזת נגמרת ב- `'\0'` מאפשרת לנו לרוץ עליה מבלי לדעת את גודלה

# איתחול מחרוזת

ראינו כי אתחול מחרוזת יעשה כך:

```
char str[] = {'h','e','l','l','o','\0'};
```

אך ניתן לאתחלה גם באופן הבא:

```
char str[] = "hello";
```

```
char str[30] = "hello";
```

במקרה זה מוקצה מערך בגודל 30, ורק ב-5 התאים הראשונים יש תווים, ובשאר 0, כמו איתחול חלקי של מערך

```
void main()
```

```
{
```

```
    char str[] = "hello";
```

```
    int x = 4;
```

```
}
```

char[]: str	'h'	1000
	'e'	1001
	'l'	1002
	'l'	1003
	'o'	1004
	'\0'	1005
int: x	4	1006

# ריצה על מחרוזת

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

```
void main()
```

```
{
```

```
    char str[] = "hello";
```

```
    int i=0;
```

```
    while (str[i] != '\0')
```

```
    {
```

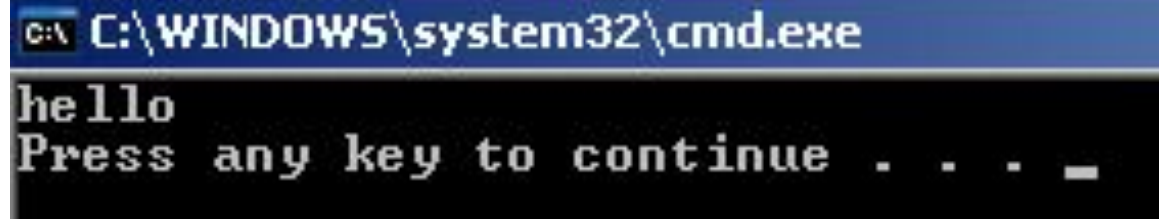
```
        printf("%c", str[i]); // same as: putchar(str[i]);
```

```
        i++;
```

```
    }
```

```
    printf("\n");
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
```

```
hello
```

```
Press any key to continue . . . _
```



# הדפסת מחרוזת

מחרוזת היא אמנם מערך, אבל כדי להדפיס את איבריה אין צורך בלולאה! □

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

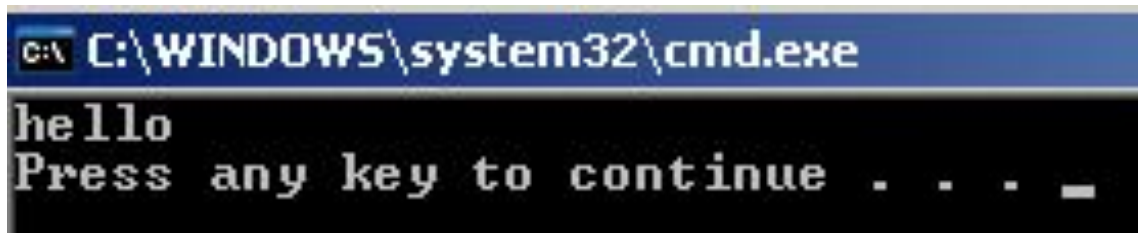
```
void main()
```

```
{
```

```
    char str[] = "hello";
```

```
    printf("%s\n", str);
```

```
}
```

A screenshot of a Windows command prompt window. The title bar at the top is blue and contains the text "C:\WINDOWS\system32\cmd.exe". The main area of the window has a black background with white text. It displays the word "hello" on the first line, followed by the prompt "Press any key to continue . . . \_" on the second line.

# הדפסת מחרוזת – נשים לב להבדל

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

```
void main()
```

```
{
```

```
    char str1[] = {'h','e','l','l','o'}; // doesn't end with '\0'!!
```

```
    char str2[] = "hello";
```

```
    char str3[] = {'h','e','l','\0','l','o'}; // note the '\0' in the middle
```

```
    char str4[] = {'h','e','l','0','l','o'}; // note the 0 in the middle
```

```
    printf("%s\n", str1);
```

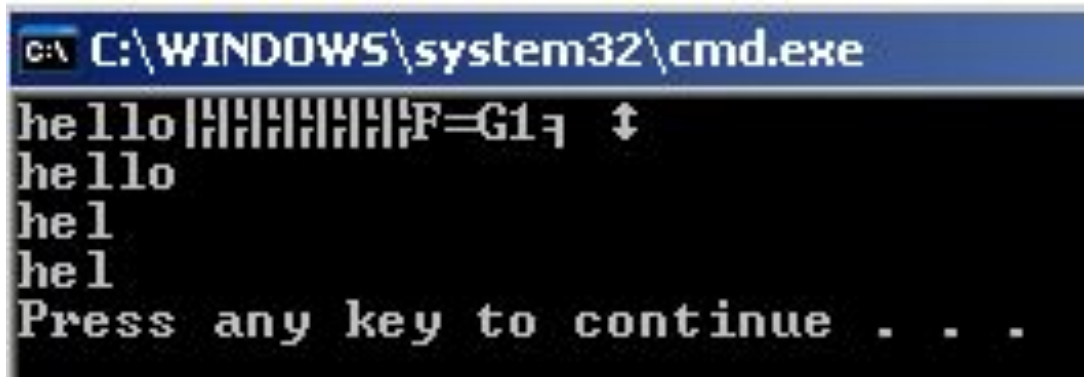
```
    printf("%s\n", str2);
```

```
    printf("%s\n", str3);
```

```
    printf("%s\n", str4);
```

```
}
```

כאשר מדפיסים מחרוזת ומשתמשים ב- `s%`  
יודפסו איברי המערך עד אשר הקומפיילר יתקל  
בתא שיש בו את הערך 0 (כלומר ערך ה- `ASCII`  
של `'\0'`)

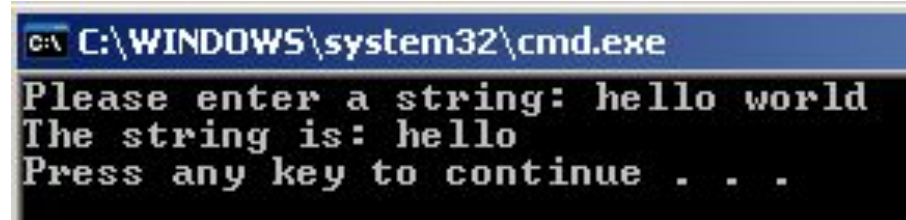


```
C:\WINDOWS\system32\cmd.exe
hello|F=G1  ↑
hello
hel
hel
Press any key to continue . . .
```

# קליטת מחרוזת

גם כדי לקלוט מחרוזת לא צריך לולאה:

```
void main()
{
    char str[10];
    printf("Please enter a string: ");
    scanf("%s", str); // no need &!!
    printf("The string is: %s\n", str);
}
```



כאשר הפקודה `scanf` מקבלת `s%` היא יודעת לקרוא מחרוזת (אוסף תווים עד אשר יוקלד רווח מטיפוס כלשהו שיסמל את ה-  
(`'0\'`)

נשים לב שאורך המחרוזת שנכניס לא יהיה יותר גדול מגודל המערך שהגדרנו, כדי לא לדרוס תאים שלא הוקצו עבורו

# הפקודות gets ו- puts

- עד כה ראינו כיצד ניתן לקרוא מילה שלמה (ללא רווחים), לשמור אותה במערך ולהדפיסה
- כעת נראה כיצד ניתן לבצע פעולות אלו גם עבור משפט (מחרוזת הכוללת רווחים)

```
void main()
```

```
{
```

```
    char str[50];
```

```
    printf("Please enter a sentence: ");
```

```
    gets(str);
```

```
    printf("The sentence is: ");
```

```
    puts(str);
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
Please enter a sentence: Hello World!
The sentence is: Hello World!
Press any key to continue . . . _
```

קולטת משפט עד ירידת שורה ('\n'), ולא רק עד רווח

מדפיסה מחרוזת למסך + ירידת שורה בסוף

# דוגמא: כמה מילים יש במשפט

(הנחה: רווח אחד בלבד)

מפריד בין מילה למילה ויש לפחות מילה אחת במשפט)

```
void main()
```

```
{
```

```
    char str[50];
```

```
    int numOfWords=0, i=0;
```

```
    printf("Please enter a sentence: ");
```

```
    gets(str);
```

```
    while (str[i] != '\0')
```

```
    {
```

```
        if (str[i] == ' ')
```

```
            numOfWords++;
```

```
        i++;
```

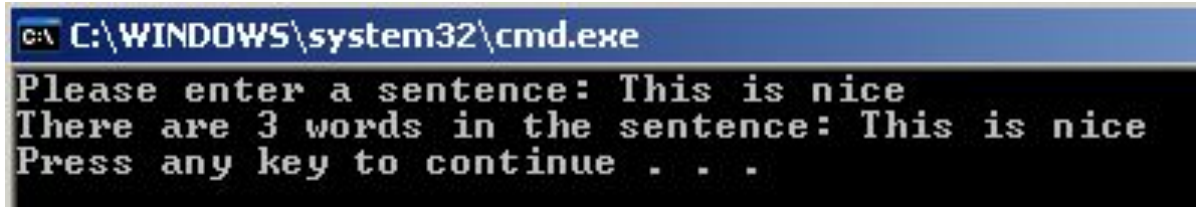
```
    }
```

```
    numOfWords++;
```

```
    printf("There are %d words in the sentence: ", numOfWords);
```

```
    puts(str);
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
Please enter a sentence: This is nice
There are 3 words in the sentence: This is nice
Press any key to continue . . .
```

# דוגמא: כמה מילים יש במשפט

(הנחה: רווח אחד בלבד)

מפריד בין מילה למילה ויש לפחות מילה אחת במשפט)

```
void main()
```

```
{
```

```
    char str[50];
```

```
    int numOfWords=0, i=0;
```

```
    printf("Please enter a sentence: ");
```

```
    gets(str);
```

```
    while (str[i] != '\0')
```

```
    {
```

```
        if (str[i++] == ' ')
```

```
            numOfWords++;
```

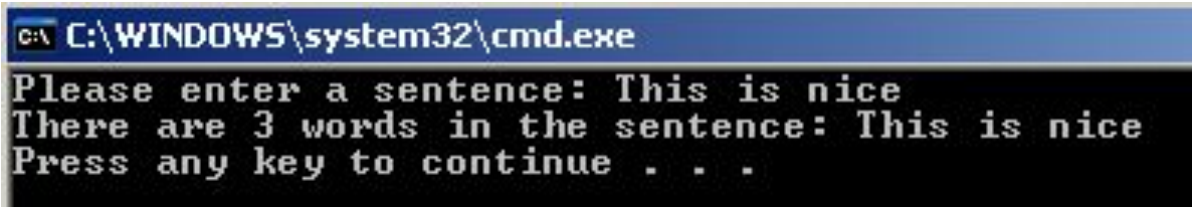
```
    }
```

```
    numOfWords++;
```

```
    printf("There are %d words in the sentence: ", numOfWords);
```

```
    puts(str);
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
Please enter a sentence: This is nice
There are 3 words in the sentence: This is nice
Press any key to continue . . .
```

תזכורת שאפשר גם כך:

# דוגמא: כמה מילים יש במשפט (בלי הנחות, כל מקרי

הקצה מטופלים 😊)

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

```
void main()
```

```
{
```

```
    char str[50];
```

```
    int numOfWorks=0, i=0;
```

```
    printf("Please enter a sentence: ");
```

```
    gets(str);
```

```
    while (str[i] != '\0')
```

```
    {
```

```
        if (str[i] != ' ') // check if this letter might be a beginning of a new word
```

```
        {
```

```
            if ( (i==0) || // if this is the first letter
```

```
                (str[i-1] == ' ') ) // of if is a letter with space before it
```

```
                numOfWorks++;
```

```
            {
```

```
                i++;
```

```
        }
```

```
    printf("There are %d words in the sentence: |%s|\n", numOfWorks, str);
```

```
}
```

```
Please enter a sentence: dfgdg fgd
There are 2 words in the sentence: | dfgdg fgd |
Press any key to continue . . . _
```

```
Please enter a sentence: bla bla
There are 2 words in the sentence: |bla bla|
Press any key to continue . . . _
```

# הספריה string.h

- יש כל מיני פעולות נפוצות שניתן לבצע על מחרוזות, כגון: חישוב אורך, העתקה, שירשור וכד'
- מאחר ואלו פעולות נפוצות, שפת C מספקת לנו ספריה הנקראת string.h המכילה פונקציות שעושות את העבודה בשבילנו
- פונקציה היא קופסא שחורה המקבלת נתונים ומחזירה נתון אחר
  - למשל, פונקציה המחזירה אורך של מחרוזת:

5" ←  ← "hello"



# הפונקציה strlen

הפונקציה strlen מקבלת מחרוזת ומחזירה את מספר התווים עד ה- '\0' הראשון שבו נתקלה

```
#include <stdio.h>
#include <string.h>
void main()
{
```

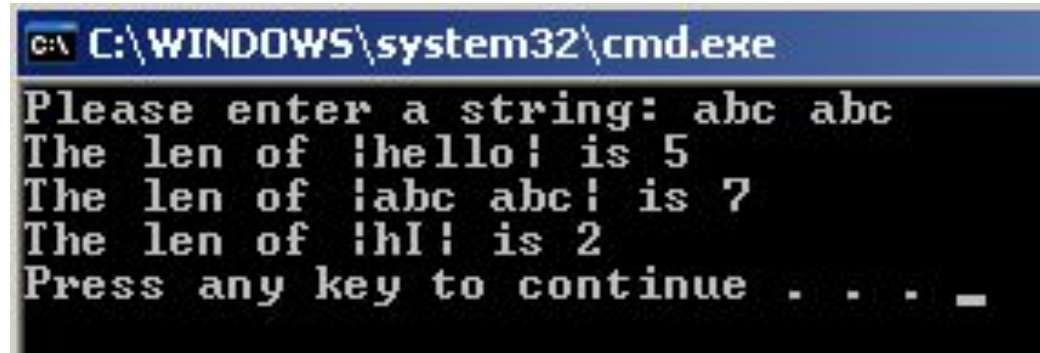
```
    char str1[] = "hello", str2[20], str3[]={ 'h','I','\0','h','I' };
    int len1, len2, len3;
```

```
    printf("Please enter a string: ");
    gets(str2);
```

```
    len1 = strlen(str1);
    len2 = strlen(str2);
    len3 = strlen(str3);
```

```
    printf("The len of |%s| is %d\n", str1, len1);
    printf("The len of |%s| is %d\n", str2, len2);
    printf("The len of |%s| is %d\n", str3, len3);
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
Please enter a string: abc abc
The len of |hello| is 5
The len of |abc abc| is 7
The len of |hI| is 2
Press any key to continue . . . _
```

# הפונקציה strlen

וכמובן שאפשר גם כך:

```
#include <stdio.h>
#include <string.h>
void main()
{
```

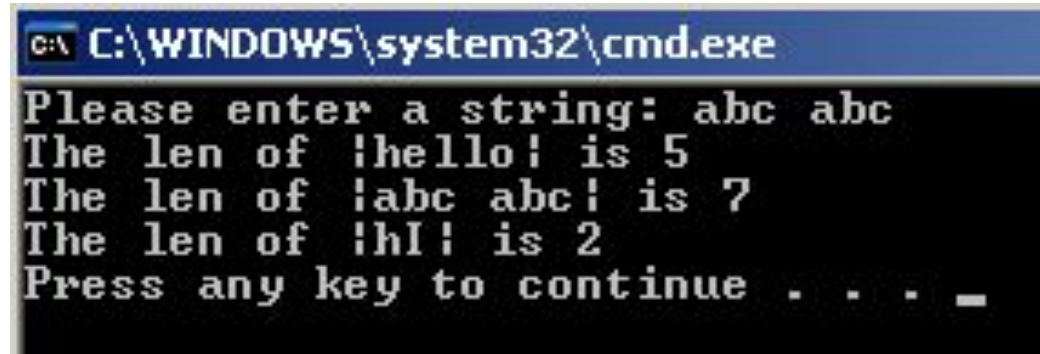
```
    char str1[] = "hello", str2[20], str3[]={'h','I','\0','h','I'};
    // int len1, len2, len3;
```

```
    printf("Please enter a string: ");
    gets(str2);
```

```
    // len1 = strlen(str1);
    // len2 = strlen(str2);
    // len3 = strlen(str3);
```

```
    printf("The len of |%s| is %d\n", str1, strlen(str1));
    printf("The len of |%s| is %d\n", str2, strlen(str2));
    printf("The len of |%s| is %d\n", str3, strlen(str3));
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
Please enter a string: abc abc
The len of |hello| is 5
The len of |abc abc| is 7
The len of |hI| is 2
Press any key to continue . . . _
```

# הפונקציה strcpy

□ ראינו שכדי להעתיק מערכים צריך לעבור בלולאה איבר-איבר

□ עבור מחרוזות הספריה string.h מספקת לנו פונקציה המעתיקה תוכן מחרוזת אחת לאחרת בפקודה אחת:

`strcpy(dest, src)`

■ `dest` היא המחרוזת אליה נרצה להעתיק

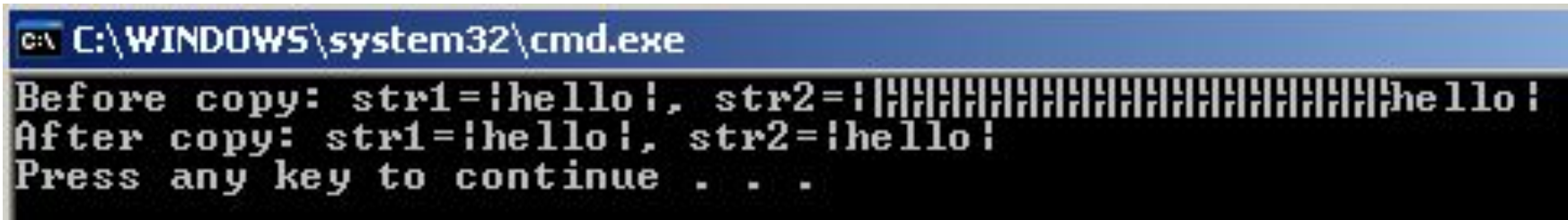
■ `src` היא המחרוזת ממנה נרצה להעתיק

□ במקרה זה "הקופסא השחורה" מקבלת 2 מחרוזות, מעדכנת את המחרוזת `dest` ומחזירה אותה



# הפונקציה strcpy - דוגמא

```
#include <stdio.h>
#include <string.h>
void main()
{
    char str1[]="hello", str2[10];
    printf("Before copy: str1=|%s|, str2=|%s|\n", str1, str2);
    strcpy(str2, str1);
    printf("After copy: str1=|%s|, str2=|%s|\n", str1, str2);
}
```



```
C:\WINDOWS\system32\cmd.exe
Before copy: str1=|hello|, str2=| | | | | | | | | |hello|
After copy: str1=|hello|, str2=|hello|
Press any key to continue . . .
```

# הפונקציה strcpy - דגשים

---

□ גם פה יש חשיבות לכך שהמחרוזת מסתיימת ב- `'\0'` וההעתקה מבוצעת עד ה- `'\0'` בלבד

□ אחריות המתכנת לוודא כי ב- `dest` מספיק מקום להכיל את `src`! הקומפיילר לא מתריע על כך, ובזמן ריצה אנו עלולים לדרוס זיכרון שאינו שלנו!

## הפונקציה `strcpy` - דגשים

```
#include <stdio.h>
#include <string.h>
void main()
{
    char str1[]={ 'h','e','l','l','o',0}, str2[8];
    printf("Before copy: str1=|%s|, str2=|%s|\n",
           str1, str2);
    strcpy(str2, str1);
    printf("After copy: str1=|%s|, str2=|%s|\n",
           str1, str2);
}
```

```
C:\WINDOWS\system32\cmd.exe
Before copy: str1=|he|, str2=|
After copy: str1=|he|, str2=|he|
Press any key to continue . . .
```

char[]: str1	'h'	1000
	'e'	1001
	0	1002
	'l'	1003
	'l'	1004
	'o'	1005
	0	1006
char[]: str2	'h'	1007
	'e'	1008
	0	1009
	??	1010
	??	1011
	??	1012
	??	1013
	??	1014

# הפונקציה strcat

□ פונקציה זו משרשרת מחרוזת אחת לסופה של אחרת:

`strcat(dest, src)`

■ `dest` היא המחרוזת אליה נרצה לשרשר לסופה

■ `src` היא המחרוזת אותה נרצה להעתיק

□ גם במקרה זה "הקופסא השחורה" מקבלת 2 מחרוזות, מעדכנת את המחרוזת `dest` ומחזירה אותה



# הפונקציה strcat - דוגמא

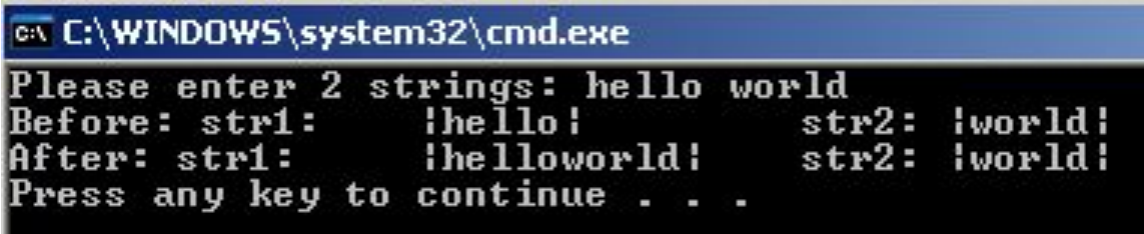
```
#include <stdio.h>
#include <string.h>
void main()
{
```

```
    char str1[20], str2[10];
```

```
    printf("Please enter 2 strings: ");
    scanf("%s %s", str1, str2);
```

```
    printf("Before: str1:\t |%s|\t str2: |%s|\n", str1, str2);
    strcat(str1, str2);
    printf("After: str1:\t |%s|\t str2: |%s|\n", str1, str2);
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
Please enter 2 strings: hello world
Before: str1:      |hello|      str2: |world|
After: str1:      |helloworld|   str2: |world|
Press any key to continue . . .
```



# הפונקציה strcat – הוספת רווח

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

```
void main()
```

```
{
```

```
    char str1[20], str2[10];
```

```
    printf("Please enter 2 strings: ");
```

```
    scanf("%s %s", str1, str2);
```

```
    printf("Before: str1:\t |%s|\t str2: |%s|\n", str1, str2);
```

```
    strcat(str1, " ");
```

```
    strcat(str1, str2);
```

```
    printf("After: str1:\t |%s|\t str2: |%s|\n", str1, str2);
```

```
}
```

```
Please enter 2 strings: hello world
Before: str1:      !hello!      str2: !world!
After: str1:      !hello world!  str2: !world!
Press any key to continue . . .
```

# הפונקציה strcat - דגשים

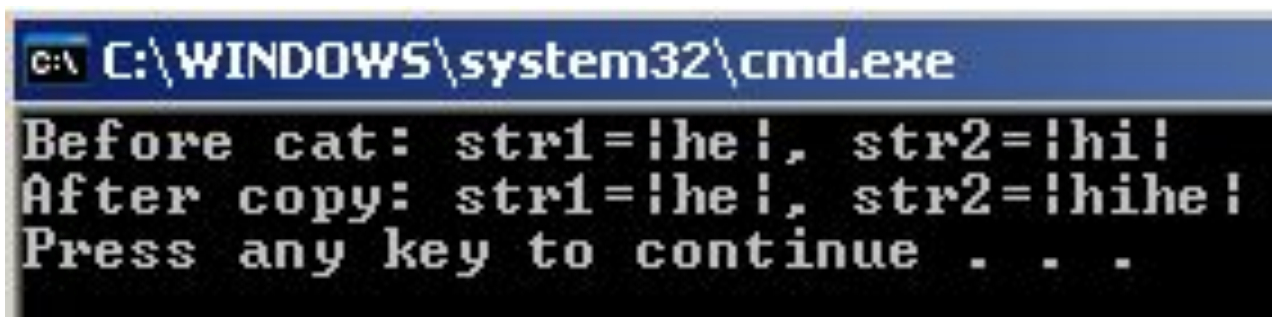
□ בדיוק כמו הדגשים של strcpy:

- גם פה יש חשיבות לכך שהמחרוזת מסתיימת ב- '\0' והשירשור מבוצע עד ה- '\0' של המחרוזת המועתקת
- בשירשור אנו דורסים את ה- '\0' של המחרוזת אליה משרשרים
- אחריות המתכנת לוודא כי ב- dest מספיק מקום להכיל את src! הקומפיילר לא מתריע על כך, ובזמן ריצה אנו עלולים לדרוס זיכרון שאינו שלנו!

# הפונקציה strcat - הרצה

```
#include <stdio.h>
#include <string.h>
void main()
{
    char str1[]={ 'h','e','0','l','l','o','0'}, str2[10]="hi";

    printf("Before cat: str1=|%s|, str2=|%s|\n",
           str1, str2);
    strcat(str2, str1);
    printf("After copy: str1=|%s|, str2=|%s|\n",
           str1, str2);
}
```



```
C:\WINDOWS\system32\cmd.exe
Before cat: str1=|he|, str2=|hi|
After copy: str1=|he|, str2=|hihe|
Press any key to continue . . .
```

char[]: str1	'h'	1000
	'e'	1001
	0	1002
	'l'	1003
	'l'	1004
	'o'	1005
	0	1006
char[]: str2	'h'	1007
	'i'	1008
	'h'	1009
	'e'	1010
	0	1011
	0	1012
	0	1013
	0	1014

# דוגמא לשימוש בערך המוחזר מ- strcat

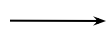
- עפ"י ההצהרה הרשמית, הפונקציות strcpy ו- strcat מחזירות את המחרוזת אותן שינו
- ברוב המקרים לא משתמשים בערך מוחזר זה (כי הנתון המקורי בכל מקרה התעדכן)

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

```
void main()  
{
```

```
    char str1[20]="hello", str2[20]="world";
```

```
    strcat(str1, " ");  
    strcat(str1, str2);
```



```
    strcat(strcat(str1, " "), str2);
```

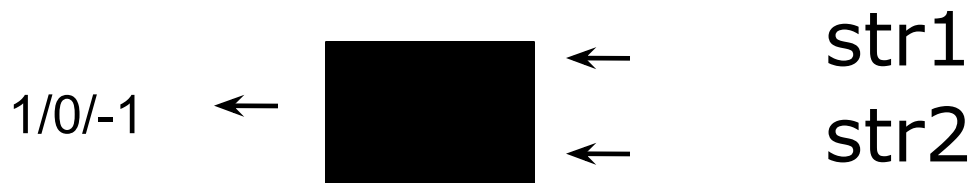
```
    printf("str1 : |%s|\n", str1);
```

```
{
```

```
str1 : !hello world!  
Press any key to continue . . .
```

# הפונקציה strcmp

- השוואה בין מחרוזות היא השוואה לקסיקוגרפית (ולא לפי אורך המילה)
- כדי להשוות בין מספרים או תווים השתמשנו באופרטורים  $<$ ,  $>$ ,  $=$ ,  $<=$ ,  $>=$ ,  $!=$
- כדי להשוות בין מחרוזות לא נשתמש באופרטורים אלו, אלא בפונקציה strcmp, שהיא "הקופסא השחורה" הבאה:



## הפונקציה strcmp (2)

□ הפונקציה מקבלת 2 מחרוזות, ומחזירה:

- 0 אם הם שוות
- 1 אם הראשונה גדולה מהשניה (כלומר במילון str1 תופיע אחרי str2)
- -1 אם הראשונה קטנה מהשניה (כלומר במילון str1 תופיע לפני str2)

□ דוגמאות:

- "str1="hello", str2="world" □ יוחזר -1
- "str1="world", str2="hello" □ יוחזר 1
- "str1="hello", str2="hello" □ יוחזר 0
- "str1="zz", str2="aaa" □ יוחזר 1
- "str1="bb", str2="bbb" □ יוחזר -1

## הפונקציה strcmp - דוגמא

```
#include <stdio.h>
#include <string.h>
void main()
```

```
{
    char str1[10], str2[10];
    int res, fContinue=1;
    while (fContinue)
    {
        printf("Please enter 2 strings, or '!' to exit: ");
        scanf("%s %s", str1, str2);
        if (strcmp(str1, "!") == 0 && strcmp(str2, "!") == 0)
            fContinue = 0;
        else
        {
            res = strcmp(str1, str2);
            printf("The result of strcmp(str1, str2): %d\n", res);
        }
    }
}
```

```
Please enter 2 strings, or '!' to exit: hello world
The result of strcmp(str1, str2): -1
Please enter 2 strings, or '!' to exit: world hello
The result of strcmp(str1, str2): 1
Please enter 2 strings, or '!' to exit: hello hello
The result of strcmp(str1, str2): 0
Please enter 2 strings, or '!' to exit: zz aaa
The result of strcmp(str1, str2): 1
Please enter 2 strings, or '!' to exit: bb bbb
The result of strcmp(str1, str2): -1
Please enter 2 strings, or '!' to exit: ! !
Press any key to continue . . . _
```

# אפסים שונים

□ המספר 0 ייוצג בזכרון ב- 4 בתים כמספר 0:

00000000 | 00000000 | 00000000 | 00000000

□ התו '0' (ערכו האסקיי 48) ייוצג בזכרון בבית אחד:

00110000

□ התו '\0' (ערכו האסקיי 0) ייוצג בזכרון בבית אחד:

00000000

□ המחרוזת "0" תיוצג בזכרון ע"י 2 בתים, מאחר וזו מחרוזת שהתו הראשון שלה מייצג את התו '0' והתא השני את התו '\0'

00110000 | 00000000



אם אי פעם שאלתם את עצמכם מה ההבדל בין 0 ל null...



אהבתי • הגיבי • שתף/שתפי

# מערך של מחרוזות

□ אם נרצה לשמור טקסט בשורות נפרדות, נשתמש במערך של מחרוזות

■ זוהי למעשה מטריצה של תווים

□ למשל מטריצה עם LINES שורות, ובכל שורה מקסימום MAX\_LETTERS תווים:

```
char text[LINES][MAX_LETTERS];
```

□ כדי לפנות לתא מסוים במטריצה נפנה ע"י text[i][j]

□ כדי לפנות לשורה שלמה (מחרוזת אחת), שהיא

למעשה איבר במערך של מחרוזות נפנה ע"י text[i]

```
#define LINES 3
#define MAX_LETTERS 81
void main()
{
```

```
    char text[LINES][MAX_LETTERS], ch;
    int i, j;
```

```
    printf("Please enter %d lines:\n", LINES);
    for (i=0 ; i < LINES ; i++)
        gets(text[i]);
```

```
    printf("Please enter a charchter to search: ");
    ch = getchar();
```

```
    for (i=0 ; i < LINES ; i++)
    {
        for (j=0 ; text[i][j] != '\0' ; j++)
        {
            if (text[i][j] == ch)
            {
                printf("The char %c appears in the line: |%s|\n", ch, text[i]);
                break; // stops inner for..
            }
        }
    }
}
```

# מערך של מחרוזות – דוגמא

## שורות בהן מופיע תו מסוים

C:\WINDOWS\system32\cmd.exe

Please enter 3 lines:

good morning

hi hi hi

hello world

Please enter a charchter to search: d

The char d appears in the line: !good morning!

The char d appears in the line: !hello world!

Press any key to continue . . . \_

# מערך של מחרוזות – דוגמא – מצא את השורה

הארוכה ביותר (הנחה: יש לפחות שורה אחת בטקסט)

'h'	'e'	'l'	'l'	'o'	' '	'w'	'o'	'r'	'l'	'd'	'0\'	0	0
'g'	'o'	'o'	'd'	' '	'm'	'o'	'r'	'n'	'i'	'n'	'g'	'0\'	0
'h'	'i'	' '	'h'	'i'	' '	'h'	'i'	'0\'	0	0	0	0	0

```
#define LINES 3
#define MAX_LETTERS 81
void main()
{
```

```
    char text[LINES][MAX_LETTERS];
    int i, maxLineLen=0, maxLineIndex, currentLen;
```

```
    printf("Please enter %d lines:\n", LINES);
    for (i=0 ; i < LINES ; i++)
        gets(text[i]);
```

```
    // find the longest line
```

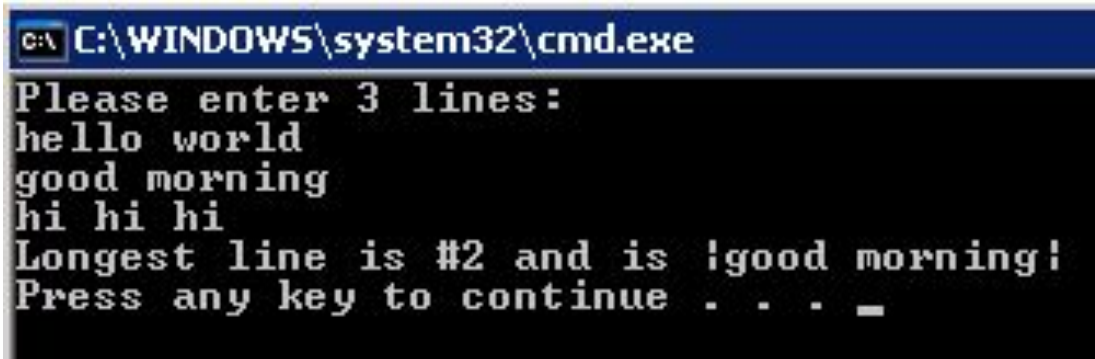
```
    for (i=0 ; i < LINES ; i++)
    {
        currentLen = strlen(text[i]);
        if (currentLen > maxLineLen)
        {
```

```
            maxLineLen = currentLen;
            maxLineIndex = i;
```

```
        }
```

```
    }
    printf("Longest line is #%d and is |%s|\n", maxLineIndex+1, text[maxLineIndex]);
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
Please enter 3 lines:
hello world
good morning
hi hi hi
Longest line is #2 and is !good morning!
Press any key to continue . . . _
```

התוכנית משום-מה קלטה שורה אחת פחות,  
ובהדפסה דילגה על השורה הראשונה..

## בעיה נפוצה

```
void main()
{
    int num, i;
    char text[3][20];

    printf("Enter a number: ");
    scanf("%d", &num);

    printf("Enter 3 lines: \n");
    for (i=0 ; i < 3 ; i++)
        cin.getline(text[i], 20);

    printf("The number is %d and the text is:\n", num);
    for (i=0 ; i < 3 ; i++)
        printf( "|%s|\n", text[i]);
}
```

```
Enter a number: 27
Enter 3 lines:
aaa
bbb
The number is 27 and the text is:
!!
!aaa!
!bbb!
Press any key to continue . . .
```

# הפתרון

```
void main()
{
    int num, i;
    char text[3][20];

    printf("Enter a number: ");
    scanf("%d", &num);
    flushall();

    printf("Enter 3 lines: \n");
    for (i=0 ; i < 3 ; i++)
        cin.getline(text[i], 20);

    printf("The number is %d and the text is:\n", num);
    for (i=0 ; i < 3 ; i++)
        printf( "|%s|\n", text[i]);
}
```

```
Enter a number: 87
Enter 3 lines:
aaa
bbb
ccc
The number is 87 and the text is:
|aaa|
|bbb|
|ccc|
Press any key to continue . . .
```

ניקוי האנטר שבבאפר

# הפונקציה sprintf

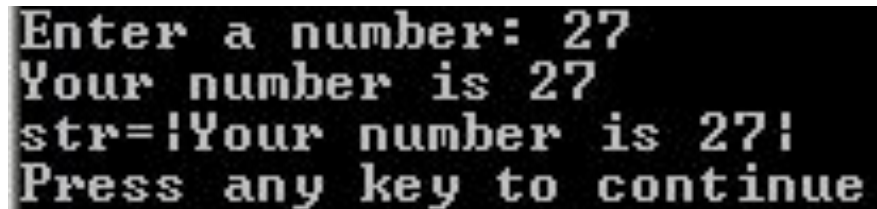
- מאפשרת לייצר פלט לתוך משתנה, במקום למסך
- בדומה ל-printf, אך מקבלת כפרמטר ראשון את המחרוזת אליה רוצה להעתיק את הנתונים

```
void main()  
{  
    char str[30];  
    int num;
```

```
    printf("Enter a number: ");  
    scanf("%d", &num);
```

```
    printf("Your number is %d\n", num);  
    sprintf(str, "Your number is %d", num);  
    printf("str=|%s|\n", str);
```

```
}
```



```
Enter a number: 27  
Your number is 27  
str=|Your number is 27|  
Press any key to continue
```

# הפונקציה sscanf

- מאפשרת לקרוא קלט מתוך משתנה, במקום מהמקלדת
- בדומה ל- scanf, אך מקבלת כפרמטר ראשון את המחרוזת ממנה נקרא את הנתונים

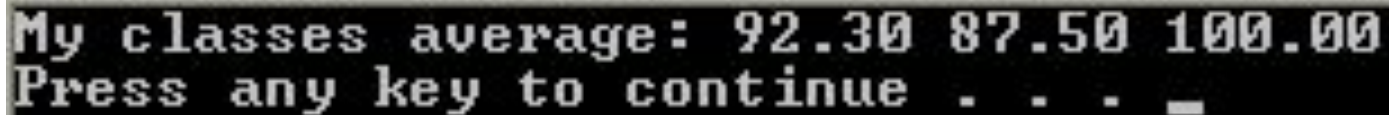
```
void main()
```

```
{
```

```
    char classesAverage[] = "92.3  87.5  100";
```

```
    float avg[3];
```

```
    int i;
```



```
My classes average: 92.30 87.50 100.00
Press any key to continue . . . _
```

```
    sscanf(classesAverage, "%f%f%f", &avg[0], &avg[1], &avg[2]);
```

```
    printf("My classes average: ");
```

```
    for (i=0 ; i < 3 ; i++)
```

```
        printf("%.2f ", avg[i]);
```

```
    printf("\n");
```

```
{
```



## הפונקציה (2) sscanf

□ השימוש בפונקציה גם מאפשר לדלג על כמות תווים כלשהי בעת קריאת המחרוזת, ע"י שימוש ב- `s*%`

```
void main()
```

```
{
```

```
    char str[30] = "Anna is 2 years old";
```

```
    char name[10];
```

```
    int age;
```

קריאת המילה בתחילת הטקסט

דילוג על המחרוזת הבאה ללא איכסונה במשתנה

```
    sscanf(str, "%s %*s %d", name, &age);
```

```
    printf("My dog's name is %s and she is %d years old\n",  
           name, age);
```

```
}
```

```
My dog's name is Anna and she is 2 years old
Press any key to continue . . . _
```

# ביחידה זו למדנו:

הספריה `string.h` □

■ `strlen`

■ `strcpy`

■ `strcat`

■ `strcmp`

מערך של מחרוזות □

הפונקציות: □

□ `sprintf`

□ `sscanf`

הפקודות: □

■ `getchar`

■ `putchar`

■ `getch`

■ `getche`

מהי מחרוזת □

איתחול מחרוזת □

הפקודות: □

□ `gets`

□ `puts`

# תרגיל 1:

- הגדר מטריצה של תווים (מערך של מחרוזות) וקרא לתוכו קלט
- הגדר מחרוזת וקרא לתוכה קלט
- הדפס כמה שורות במטריצה זהות למחרוזת
- דוגמא:

```
Please enter 4 lines:
hi
good morning
hi
this is nice
Please enter a word: hi
2 lines in the text are equal to !hi!
Press any key to continue . . .
```

## תרגיל 2:

□ כתוב תוכנית המגדירה 3 מערכים של מחרוזות (מטריצות) (text1, text2, text3) באותו הגודל וקרא נתונים ל-2 המטריצות הראשונות

□ הפונקציה תשים מחרוזות ב-text3 באופן הבא:

- במידה והאורך הכולל של השורה המתאימה ב-text1 וב-text2 קטן מאורך שורה אפשרית, נעתיק את השורה המתאימה מ-text1 ל-text3 ואח"כ נשרשר את השורה המתאימה מ-text2. אחרת נשים ב-text3 שורה ריקה.

```
Please enter 3 lines <maximum 10 chars>
hi
good
abc

Please enter 3 lines <maximum 10 chars>
hihi
morning
def

text3 is:
hihihi

abcdef
Press any key to continue . . . _
```

□ לבסוף התוכנית תדפיס את text3

□ שימו לב: אין "להמציא את הגלגל מחדש"!!

## תרגיל 3:

- כתוב תוכנית המגדירה מערך של מחרוזות. יש להזיז את איבר המערך כך שבשורה השניה תהיה המחרוזת הראשונה, בשורה השלישית המחרוזת השניה וכו'.
- השורה האחרונה תועתק במקום המחרוזת הראשונה
- לבסוף התוכנית תדפיס את המערך המעודכן

```
Please enter 5 lines <maximum 10 chars>
aaa
bb
ccccc
dd
ee
text after rotation:
ee
aaa
bb
ccccc
dd
Press any key to continue . . . _
```