

נושאים

- כמה הערות על פונקציות
- The C preprocessor
- printf / scanf
- מצביעים

סיכום של פונקציות:

- לפונקציה שם ייחודי אשר מופיע פעם אחת בתוכנית
- לפונקציה יכולים להיות מספר פרמטרים (גם 0)
- לפונקציה יש ערך מוחזר פרט לפונקציות void
- ברירת המחדל לערך מוחזר אם לא הוגדר – int
- לא חייבים להשתמש בערך המוחזר של הפונקציה
- בקריאה לפונקציה – יש לספק את הערכים לכל הפרמטרים
- בקריאה, הערכים מועתקים לפרמטרים של הפונקציה (אבל ראו בהמשך)
- אין מגבלה על שמות הפרמטרים בתוך פונקציה, וניתן להשתמש גם בשדות שבשימוש בתוך פונקציה אחרת.
- ניתן לקרוא לפונקציה מכל מקום בו היא מוגדרת והלאה, ניתן להיעזר ברישום מוקדם (prototype) לשימוש לפני ההגדרה המלאה.

Recursive Functions

פונקציות רקורסיביות

```
double power(double val, unsigned pow)
{
    if (pow == 0)    /* pow(x, 0) returns 1 */
        return 1.;
    else return power(val, pow - 1) * val;
}
```

-> משתנים לוקליים <-

The C Preprocessor

המעבד הראשוני

```
#include "MyDefinitions.h"
```

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

```
int printf(const char* format, ...);
```

```
int scanf(const char* format, ...);
```

Files קבצים

The C Preprocessor

המעבד הראשוני

```
#include <stdlib.h>
```

```
void* malloc(size_t size);
```

```
void free(void* p);
```

```
int abs(int n);
```

```
long labs(long n);
```



הקצבת זכרון

המעבד הראשוני

```
#include <math.h>
```

```
gcc file.c -lm
```

```
double sin(double x);  
double cos(double x);  
double tan(double x);  
double log(double x);  
double log10(double x);  
double asin(double x);  
double acos(double x);  
double atan(double x);  
double atan2(double y, double x);
```

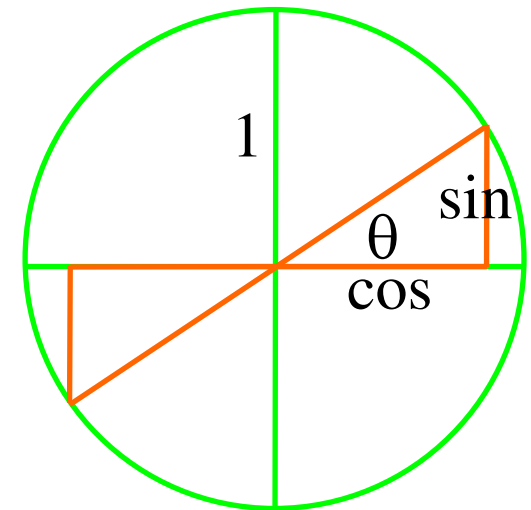
$\tan^{-1}(y/x)$ in $[-\pi, \pi]$

```
double sinh(double x);  
double cosh(double x);  
double tanh(double x);  
double exp(double x);
```

$[-\pi/2, \pi/2]$

$[0, \pi]$

$[-\pi/2, \pi/2]$



המעבד הראשוני

```
#include <math.h>
```

```
gcc file.c -lm
```

```
double pow(double x, double y);
```

 x^y

```
double sqrt(double x);
```

 \sqrt{x}

```
double cbrt(double x); (לפעמים)
```

 $\sqrt[3]{x}$

```
double ceil(double x);
```

```
ceil(5.3) => 6
```

smallest integer not less than x

```
ceil(-0.1) => 0
```

```
double floor(double x);
```

```
floor(5.5) => 5
```

largest integer not greater than x

```
floor(-0.1) => -1
```

```
double fabs(double x);
```

 $|x|$

```
int round(double x) {  
    return (int) ((x < 0) ? (x-.5) : (x+.5)); }  
}
```

cast: הטלה

(type-name) expression

המעבד הראשוני

```
#define name replacement-text
```

```
#define SIZE 100
```

```
double a[SIZE],b[SIZE][10];
```

```
#define Pi 3.1415926535897932385
```

```
area=Pi*r*r;
```

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

```
Food=Pie+Cake;
```

אין החלפה בתוך

מרכאות או שמות

המעבד הראשוני

```
#define name replacement-text
```

```
#define Area(r) Pi*r*r          שגיאה!!!
```

```
x=Area(s+1);
```

```
x=Area(s++);
```