



מבוא למדעי המחשב

תירגול 2: מבוא לתיכנות ב-C



תוכנייה

- משתנים
- פלט
- משפטי תנאי





משתנים



מבוא למדעי המחשב מ' - תירגול 2

משתנים - תמונת הזיכרון

הזיכרון

לכל תא בזיכרון
יש כתובת

כתובת התא	תא
#1000	10
#1004	-4
#1008	לא מאותחל
#1012	67

ערך השמור בתא

- "משתנה" הוא תא בזיכרון, שניתן לו שם ויכול להכיל ערך.
- כמעט כל הפעולות בתכנית מחשב דורשות גישה לזיכרון לקריאה/כתיבה. גישה זו מתאפשרת בעזרת המשתנים, מה שהופך אותם לחלק מרכזי בכל תכנית.



משתנים- הגדרה ואתחול

- לפני שניתן להשתמש במשתנה, יש להגדיר אותו (לתת לו שם וטיפוס).
- הגדרת המשתנה בשפת C מתבצעת בעזרת הכרזה בסגנון:

המלה `int` מסמנת את טיפוס המשתנה. טיפוס, הוא חלק קבוע מהגדרה. טיפוס `int` מגדיר משתנה ששומר ערכים שלמים. טיפוס `double` מגדיר משתנה ששומר ערכים עם שבר עשרוני. עוד על טיפוסים- בתרגול הבא

```
int x=5;
```

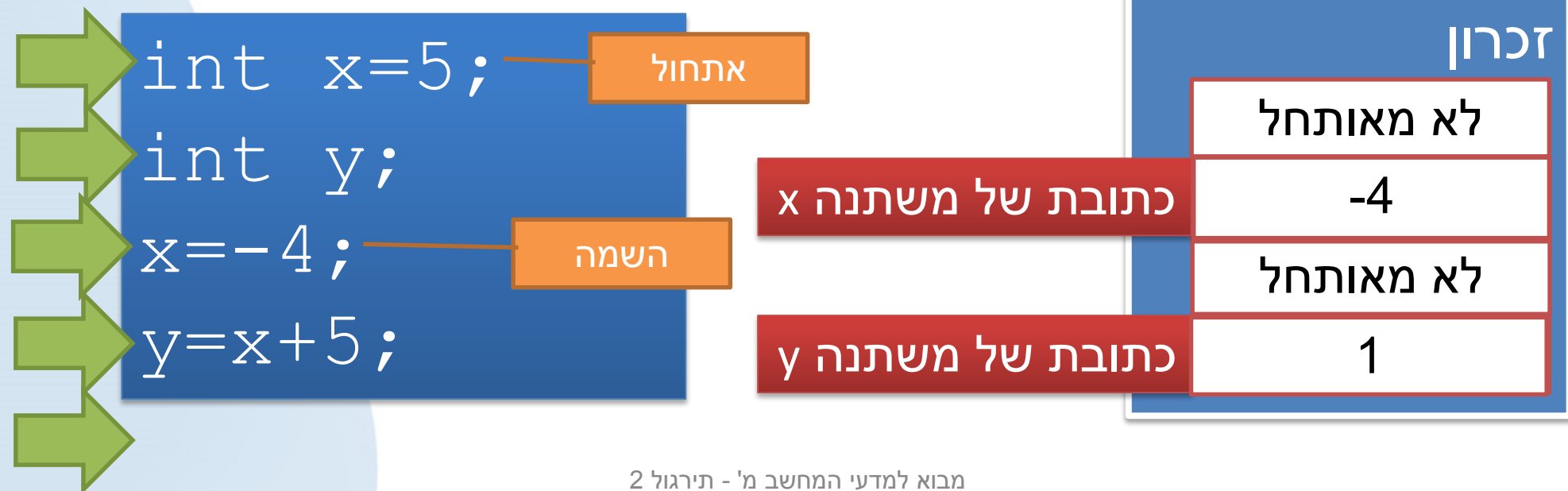
ההגדרה מסתיימת בנקודה-פסיק.

המלה `x` היא שם המשתנה. לא כל מילה יכולה להיות שם של משתנה. למשל: שם משתנה אינו יכול להתחיל בספרה.
דרישה בקורס: יש לתת למשתנים שם המלמד על המשמעות שלהם בתוכנית!

ערך התחלתי אופציונלי!
אם לא קובעים ערך התחלתי, המשתנה מכיל "זבל" – ערך כלשהו שלא ידוע מראש והוא שונה מריצה לריצה.

משתנים- השמה וקריאה

- לאחר הגדרת המשתנה, ניתן לעדכן את תוכן תא הזיכרון על-ידי כתיבה (השמה) למשתנה (המסומנת על-ידי סימן "=").
- כמו כן, ניתן לקרוא מתא הזיכרון באמצעות המשתנה.
- דוגמא: נעקוב אחרי תוכן הזיכרון במהלך ריצת התוכנית הבאה:



משתנים- חישובים בסיסיים

- ב-C ישנן חמש פעולות חשבון יסודיות:

+	חיבור
-	חיסור
*	כפל
/	חילוק
%	שארית

- ניתן להפעיל פעולות אלה הן על מספרים מפורשים, והן על משתנים שמכילים מספרים, כרצוננו.
שפת C מחשבת את הפעולות הללו על פי סדרן המתמטי (כלומר כפל לפני חיבור וכו').

פעולות חילוק ושארית

- כאשר מתבצעת פעולת חילוק (/) בין שני שלמים, התוצאה **מומרת למספר שלם** באמצעות קיצוץ הספרות שלאחר הנקודה (ולא עיגול!)
- פעולת השארית (%) (מודולו) מוגדרת על שלמים בלבד ומחזירה את השארית של פעולת החילוק בין שני הפרמטרים שלה.

±11 / ±5 לפי שיטת השארית		
-11	11	<div>a b</div>
-2 ∟ -1	2 ∟ 1	5
2 ∟ -1	-2 ∟ 1	-5

תוצאת החילוק משמאל

ותוצאת השארית מימין

משתנים- תרגילים

- תרגיל 1: כתבו תכנית המגדירה 2 משתנים בעלי ערך 19 ו 88, ומכניסה את הסכום למשתנה שלישי.

תרגיל 1 - פתרון

```
int main()
{
    int num1 = 19, num2 = 88;
    int sum=num1+num2;
    return 0;
}
```

- בעיה: מי שמפעיל את התוכנית לא רואה כלום!
- כעת נלמד כיצד להדפיס למסך את התוצאה.



פלט



מבוא למדעי המחשב מ' - תירגול 2

printf - פלט

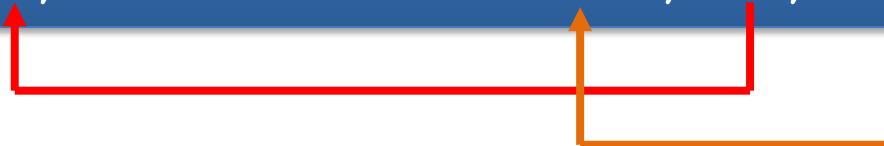
- הדפסת מחרוזת קבועה (משפט שאינו משתנה):

```
printf("I don't care about your variables!");
```

- הדפסת תוכן של משתני התוכנית באמצעות % (משמש להדפסת משתנה מסוג int ו-%f משמש להדפסת משתנה מסוג double):
סדר ההדפסה הוא לפי סדר העברת הפרמטרים ל-printf

```
int x=5;  
double d=3.2;  
printf("x is: %d, and d is: %f", x, d);
```

מה אם המשתנים לא היו מאותחלים?



printf - פלט

- תווים מיוחדים מודפסים באמצעות התו '\'

מדפיס תו של ירידת שורה (ללא הוספת התו הזה מפורשות ההדפסות ימשיכו להיות באותה השורה!)

`\n`

מדפיס טאב (כמו לחיצה על מקש tab במקלדת)

`\t`

מדפיס את התו "

`\"`

- ואיך נדפיס את התו '\'

מדפיס את התו '\'

`\\`

חובה להוסיף בתחילת הקוד
כדי להשתמש ב printf

דוגמא ל-printf

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

הפלט:

```
int main()
```

```
{
```

```
    int x=5;
```

```
    double d=3.2;
```

```
    printf("Hello world");
```

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

```
    printf("x is: %d, d is: %f\n",x,d);
```

```
    printf("x+3 is: %d, d is: %.2f\n",x+3,d);
```

```
    printf("To print an int use %%d\n");
```

```
    return 0;
```

```
}
```

```
Hello worldHello world
x is: 5, d is: 3.200000
x+3 is: 8, d is: 3.20
To print an int use %d
```

הקובץ `stdio.h`

- `printf` אינה מילה שמורה בשפת C (כמו `return`, `int`).
אלא, היא פונקציה שיש להגדירה והגדרתה נמצאת בקובץ `stdio.h` (קיצור של Standard Input/Output)
- לכן, יש להוסיף בתחילת התכנית (ממש בהתחלה, עוד לפני תחילת `main`) את השורה:

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

- פקודה זו מורה למהדר להשתמש באוסף הפונקציות המוגדר בקובץ `stdio.h` (שהוא חלק מהספריה הסטנדרטית).

משתנים וקלט/פלט

- תרגיל 1: כתבו תכנית המדפיסה את הסכום של 19 ו 88.

```
#include <stdio.h>

int main()
{
    printf("This program knows to add!:\n");
    int x=88,y=19;
    printf("The sum is: %d",x+y);
    return 0;
}
```


משתנים וקלט/פלט

- תרגיל 2: כתבו תוכנית המחשבת ומדפיסה את השטח וההיקף של מעגל ברדיוס 3.5.

```
#include <stdio.h>
#define PI 3.14
```

```
int main()
{
    double radius=3.5;

    //doing calculations
    double perimeter = 2*PI*radius;
    double area = PI*radius*radius;

    //printing the result
    printf("The perimeter is: %f and the area is: %f",
           perimeter, area);

    return 0;
}
```

שימו לב להגדרת הקבוע פאי כ-define.
לא רק קבועים ידועים כמו פאי יוגדרו ב-define,
אלא כל ערך קבוע בעל משמעות בתוכנית.



משפטי תנאי



משפטי תנאי

- בשפת C אנו יכולים לכתוב פקודה שתתבצע רק במצבים מסוימים:

```
if (x < 0) {  
    printf("Is negative");  
}
```

- קיימת האפשרות לציין בנוסף גם פקודות שיבוצעו במקרה שהתנאי אינו מתקיים.

```
if (x < 0) {  
    printf("Is negative");  
}  
else {  
    printf("Isn't negative");  
}
```



משפטי תנאי

- במקרה הפרטי שבו ישנו חישוב ערך בודד בבלוק ה-`if` וה-`else` ניתן להשתמש באופרטור המקוצר `?:`

```
if ( x+2 < 0 ) {  
    y=x+2;  
}  
else {  
    y=-(x+2);  
}
```



```
y = (x+2 < 0) ? x+2 : -(x+2);
```

תנאי לבדיקה

ערך במקרה שהתנאי נכון

ערך במקרה שהתנאי אינו נכון

מבוא למדעי המחשב מ' - תירגול 2



משפטי תנאי

- תרגיל 3: כתבו תכנית המדפיסה את המקסימלי מבין 2 משתנים (במקרה של שוויון לא משנה מי יודפס).





```
#include <stdio.h>

int main()
{
    int x=54,y=121;

    if (x>y) {
        printf("%d",x);
    }
    else {
        printf("%d",y);
    }
    return 0;
}
```

פתרון- אפשרות נוספת

- את קטע הקוד הזה:

```
if (x>y) {  
    printf("%d", x);  
}  
else {  
    printf("%d", y);  
}
```

- ניתן להחליף בזה:

```
printf("%d", x>y ? x : y);
```



קליטת מספרים – הצצה לתרגול 3

```
#include <stdio.h>

int main()
{
    int x = 0, y = 0;
    scanf("%d%d", &x, &y);

    if (x > y) {
        printf("%d", x);
    }
    else {
        printf("%d", y);
    }
    return 0;
}
```

- הקוד קצת משעמם – אנחנו יודעים איזה משתנה יותר גדול.
- בהמשך, נלמד להשתמש בפונקציה `scanf` שמקבלת מספר מהמשתמש.
- במצב כזה, לא ידוע מראש מה יותר גדול.
- `scanf("%d%d",&x,&y)` מכניס ל `x` ול `y` את הערכים שהמשתמש הכניס.

משפטי תנאי

- אם קיימות יותר משתי אופציות ניתן לקנן (=לכתוב אחד בתוך השני) כמה בלוקים של if-else:

```
if (a<0) {  
    ...  
}  
else if (a==0) {  
    ...  
}  
else if (a>0) {  
    ...  
}  
else {  
    printf("???");  
}
```



מבנה
else-if



כתיבה
רגילה

```
if (a<0) {  
    ...  
}  
else  
    if (a==0) {  
        ...  
    }  
    else  
        if (a>0) {  
            ...  
        }  
        else {  
            printf("???");  
        }  
}
```



משפטי תנאי

- תרגיל 4: כתבו תכנית המדפיסה את המקסימלי מבין שני מספרים מסוג `int`. במקרה של שיוויון יש להדפיס הודעה המודיעה כי יש שיוויון.



```
if (x>y) {  
    printf("%d", x);  
}  
else {  
    printf("%d", y);  
}
```



```
if (x>y) {  
    printf("%d", x);  
}  
else if (y>x) {  
    printf("%d", y);  
}  
else {  
    printf("Numbers are equal!");  
}
```

משפטי תנאי

- חפשו את ההבדלים:

```
if (x<3){  
    printf("first\n");  
}  
else if (x<7){  
    printf("second\n");  
}  
else {  
    printf("third\n");  
}
```

```
if (x<3){  
    printf("first\n");  
}  
if (x<7){  
    printf("second\n");  
}  
else {  
    printf("third\n");  
}
```



משפטי תנאי

- במקרה הפרטי שבו יש לבחור בין פעולות שונות לפי ערכים קבועים של ביטוי כלשהו, נשתמש ב-switch:

```
switch (expression)
{
    case value_1 : פקודה 1
                  פקודה 2
                  ...
                  break;

    case value_2 : פקודה 1
                  ...
                  break;

    ...

    case value_n : ( ... )
    default       : ( ... )
}
```

ביטוי כלשהו שמחזיר מספר שלם
(למשל: שם של משתנה מסוג int)

ערך מספרי שלם קבוע כלשהו,
שהביטוי יכול לקבל

default הוא אופציונלי



משתני תנאי

תרגיל 5: כתבו תכנית הבוחנת משתנה של מספר שלם בטווח 1-10. אם המספר הוא ראשוני התכנית תדפיס זאת ואחרת תדפיס התכנית האם המספר הינו זוגי או אי-זוגי. אם המספר אינו בטווח בין 1-10 יש להדפיס הודעת שגיאה.

משתני תנאי

מה יקרה אם נסיר את ה-break מסופי השורות?

```
int num=3; //or use scanf
switch (num)
{
    case 1 : printf("Number is odd\n"); break;
    case 2 : printf("Number is prime\n"); break;
    case 3 : printf("Number is prime\n"); break;
    case 4 : printf("Number is even\n"); break;
    case 5 : printf("Number is prime\n"); break;
    case 6 : printf("Number is even\n"); break;
    case 7 : printf("Number is prime\n"); break;
    case 8 : printf("Number is even\n"); break;
    case 9 : printf("Number is odd\n"); break;
    case 10 : printf("Number is even\n"); break;
    default : printf("Error!\n"); break;
}
```

ניתן לכתוב מס' פקודות בשורה אחת.
מה שמפריד בין הפקודות זה ";" ולא אנטר



תרגילי סיכום (חזרה על כל הנושאים)

תרגיל 6: כתבו תכנית הבוחנת מספר שלם בן שלוש ספרות (אין צורך לוודא זאת) ומדפיסה את ספרת האחדות, העשרות והמאות של המספר.

תרגילי סיכום (חזרה על כל הנושאים)

```
int num = 234; //or use scanf
```

```
//calculating units digit
```

```
int units = num % 10;
```

```
//calculating tens digit
```

```
int tens = num / 10;
```

```
tens = tens % 10;
```

```
//calculating hundreds digit
```

```
int hundreds = num / 100;
```

```
hundreds = hundreds % 10;
```

```
printf("Units digit: %d, Tens digit: %d, Hundreds digit: %d",  
      units, tens, hundreds);
```

שימו לב להערות המסבירות את הקוד.
בתרגילי הבית חובה להשתמש בהערות
כדי להסביר מה אתם מנסים לעשות!

שימו לב להבדל בין הגדרת משתנה לבין השמה

תרגילי סיכום (חזרה על כל הנושאים)

תרגיל 7: כתבו תכנית הבוחנת שני מספרים שלמים ומדפיסה את סכום הערכים המוחלטים שלהם.

תרגילי סיכום (חזרה על כל הנושאים)

```
int x = -9, y = 6; //or use scanf

//put absolute value of x in x
if (x < 0){
    x = -x;
}
//put absolute value of y in y
if (y < 0){
    y = -y;
}

printf("Sum of absolute values: %d\n", x+y);
```

תרגילי סיכום (חזרה על כל הנושאים)

תרגיל 8: כתבו תכנית אשר בודקת אם שנה הנתונה במשתנה היא שנה מעוברת. שנה מעוברת מתרחשת כל 4 שנים, מלבד בכל שנה שהיא כפולה של 100, אך לא של 400.

```
int year = 2100; //or use scanf
```

```
if( year%4 == 0) {  
    if( year%100 == 0) {  
        if ( year%400 == 0) {  
            printf("%d is a leap year.", year);  
        }  
        else {  
            printf("%d is not a leap year.", year);  
        }  
    }  
    else {  
        printf("%d is a leap year.", year );  
    }  
}  
else {  
    printf("%d is not a leap year.", year);  
}
```

שימו לב איך בכל פעם שפותחים סוגר מסולסל
כל הקוד עד לסגירתו מוזז פנימה. בצורה כזו
קל להבין את מבנה הקוד מהסתכלות עליו.
בתרגילי הבית חובה להקפיד על כך!!

תרגילי סיכום (חזרה על כל הנושאים)

תרגיל 9: כתבו תכנית הבוחנת שני משתנים המייצגים תאריך (יום וחודש. ניתן להניח כי הקלט הוא תאריך חוקי), ומדפיסה את התאריך שבוע לאחר מכן, כאשר:

א. ניתן להניח כי בכל חודש 30 ימים בדיוק.

ב. בכל חודש מס' הימים האמיתי של אותו החודש (ינואר-31, פברואר-28, מרץ-31, אפריל-30, מאי-31, יוני-30, יולי-31, אוגוסט-31, ספטמבר-30, אוקטובר-31, נובמבר-30, דצמבר-31)

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

תרגילי סיכום (חזרה על כל הנושאים)

```
int day=25; //or use scanf
int month=12; //or use scanf

int new_day = (((day - 1) + 7) % 30) + 1;
int next_month = (((month - 1) + 1) % 12) + 1;
int new_month = (new_day < 8) ? next_month : month;

printf("A week later the date will be %d/%d\n", new_day, new_month);
```



```
int days_in_month = 0;
switch (month)
{
    case 1 : days_in_month = 31; break;
    case 2 : days_in_month = 28; break;
    case 3 : days_in_month = 31; break;
    case 4 : days_in_month = 30; break;
    case 5 : days_in_month = 31; break;
    case 6 : days_in_month = 30; break;
    case 7 : days_in_month = 31; break;
    case 8 : days_in_month = 31; break;
    case 9 : days_in_month = 30; break;
    case 10 : days_in_month = 31; break;
    case 11 : days_in_month = 30; break;
    case 12 : days_in_month = 31; break;
}
new_day = (((day - 1) + 7) % days_in_month) + 1;
next_month = (((month - 1) + 1) % 12) + 1;
new_month = (new_day < 8) ? next_month : month;
```