

# פרק 1 – מבוא

בפרק מבוא קצר זה נסביר מעט על המחשב, על תפקידיו, על מבנהו ועל אופן השימוש בו לצורך פתרון בעיות מסוגים שונים. ייתכן כי לאלה מביניכם הרגילים בשימוש במחשב, ואולי אף בתכנות, חלק מהמושגים יהיו מוכרים. בכל זאת, סביר שקריאת הפרק תאיר כמה מהמושגים האלה ואת הקשרים ביניהם באור מעט שונה, וסביר שכמה מהמושגים המוצגים בפרק יהיו חדשים גם עבור התלמידים שרגילים בשימוש במחשב.

מחשבים מצויים במקומות רבים: הם מנחים מטוסים, אוניות וספינות חלל; הם מפקחים על מיליוני טלפונים המחוברים ברשת ופזורים על פני מדינות שונות; הם משמשים לחיזוי מזג האוויר, לכתובה ולהדפסה של מסמכים, לבקרה על מערכות ייצור אוטומטיות, להלחנת מוסיקה, למשחקים ולדברים רבים נוספים.

ללא המחשב לא היה מתאפשר מבצע הנחתת אדם על הירח. המבצע דרש פתרון בעיות חישוביות קשות ומסובכות לקביעת מסלול חללית בהשפעת הירח, כדור הארץ והשמש. ביצוע החישובים האלה ללא מחשב היה מצריך עבודת צוות גדולה במשך עשרות שנים. בנוסף, כל שינוי במועד ההמראה או במסלול הטיסה חייב פתרון מחדש של בעיות חישוביות אלו. המבצע לא היה מתאפשר ללא מחשב.

## 1.1 מהו מחשב?

**מחשב** (computer) הוא מכונה אלקטרונית הקולטת נתונים, מעבדת אותם ופולטת מידע הנוצר בתהליך העיבוד. הנתונים שקולט המחשב נקראים **קלט** (input); המידע שפולט המחשב נקרא **פלט** (output); העיבוד המבוצע במחשב מונחה על ידי אוסף הוראות הנקרא **תוכנית מחשב** (computer program).

הנה כמה דוגמאות להמחשת ההגדרות האלו:

- ◆ כשאנו מגיעים לקנות כרטיסים לסרט, הקופאי מקיש בלוח המקשים של המחשב שעל שולחנו את מספר הכרטיסים שאנו מבקשים לקנות. המחשב מעבד נתון זה בבדיקת המקומות הפנויים, בהקצאת מקומות כמבוקש ובסימון המקומות שהוקצו כתפוסים. המדפסת מדפיסה כרטיסים שעליהם מסומנים המקומות שהוקצו. במקרה זה הקלט הוא מספר הכרטיסים המבוקש, והפלט הוא הכרטיסים שעליהם מקומות מסומנים.
- ◆ אמן אנימציה מציין כקלט על צג מחשב שתי נקודות לתנועה של דמות – נקודת התחלה ונקודת סיום. המחשב מעבד נתונים אלה יחד עם נתונים נוספים על הדמות, ומציג על הצג כפלט את תנועת הדמות מנקודת ההתחלה לנקודת הסיום.
- ◆ מנהל חשבונות מקיש כקלט שם של עובד ואת מספר הימים שעבד בחודש האחרון. המחשב מעבד נתונים אלה יחד עם נתונים אחרים השמורים בו (כמו דרגת העובד), ומדפיס כפלט את המשכורת שמגיעה לעובד עבור החודש האחרון.
- ◆ מדען, המשתמש במחשב לצורך חישוב מסלול התעופה של טיל, נותן כקלט למחשב את נקודת השיגור הטיל, את זמן השיגור, את מהירות הטיל ונתונים על הרוחות במסלול מעופו הצפוי של הטיל. המחשב מעבד נתונים אלה בעזרת נוסחאות לחישוב תעופת טיל ובעזרת מידע השמור בו על כדור הארץ, ונותן כפלט את המקום שאמור הטיל לנחות בו.

## שאלה 1.1

הביאו דוגמאות מסביבת הבית ובית הספר לשימוש במחשב, וציינו עבור כל דוגמה את הקלט ואת הפלט.

ייחודה של המכונה "מחשב" לעומת מכונות אחרות הוא במגוון השימושים הרחב שלה. בעוד שבמכונת כביסה אנו משתמשים כדי לכבס, במקרר כדי לשמור על מזון ובמכונת כתיבה כדי להגיע ממקום למקום, הרי שבמחשב משתמשים בקשת רחבה של משימות. זה נובע מכך שהמחשב מבצע פעילויות שונות של עיבוד ואחסון מידע, פעילויות המונחות כולן על ידי תוכניות מחשב מתאימות.

אמנם, קיימים כיום מחשבים אשר מסוגלים לבצע בצורה מוגבלת תפקודים אנושיים, כמו ראייה, שמיעה, דיבור ותנועה, אבל המחשב אינו כל יכול. הוא בסך הכול מכונה. הוא איננו יכול למשל לדמיין, לכאוב או לשמוח. למעשה, גם אם נגביל את הדיון למשימות שאינן מערבות רגשות, ישנן משימות רבות שאינן ניתנות לביצוע על ידי מחשב.

יתרון בולט של מחשב הוא בכך שהוא מבצע פעולות חישוב ועיבוד מידע במהירות עצומה ובדיוק רב. למשל, מחשב יכול לקלוט אלף מספרים בני חמש ספרות כל אחד, ולחשב תוך שבריר שנייה את סכומם. מחשב יכול לקלוט טקסט בן אלף מילים, ולחשב תוך שבריר שנייה את האות השכיחה ביותר בטקסט.

יתרון חשוב נוסף של מחשב הוא שניתן לאחסן בזיכרונו מיליוני נתונים. למשל, במחשב של משרד הפנים מאוחסנים נתונים על כל אחד מתושבי המדינה (שם, מספר זהות, תאריך לידה, מצב משפחתי ועוד). במחשב של בנק מאוחסנים פרטים אישיים על כל לקוח של הבנק, והנתונים על התנועות בכל חשבונות הבנק.

אמנם המחשב מבצע פעולות חישוב ועיבוד בדיוק רב, אך מאחר שהוא לא יותר ממכונה, המבצעת הוראות, לא מובטח שתמיד ייתן המחשב כפלט תוצאות עיבוד נכונה. פלט לא נכון יכול להתקבל כתוצאה מקלט שגוי או מתוכנית מחשב שגויה.

**?** מהי תוכנית מחשב שגויה?

תוכנית מחשב נכתבת על ידי בני אדם. ייתכן שההוראות הכלולות בה אינן מורות על העיבוד הדרוש. במקרה כזה התוכנית שגויה, וייתכן כי כאשר המחשב פועל על פיה הוא ייתן כפלט תוצאות עיבוד לא נכונה. למשל, ייתכן כי מתכנת יכתוב תוכנית לחישוב ממוצע של אלף מספרים, ובה יורה על סיכום המספרים אך יטעה וישכח להורות על חלוקת הסכום המחושב ב-1000. ברור כי עיבוד שיתבצע על פי תוכנית זו לא יבצע חישוב ממוצע כנדרש וייתן פלט שגוי.

אנו מבחינים בין שני צדדים של מחשב: חומרה ותוכנה.

**חומרה (hardware)** היא הרכיבים הפיסיים שמרכיבים את המחשב.

**תוכנה (software)** היא אוסף תוכניות המחשב.

## 1.2 חומרה

בסעיף זה נתאר על קצה המזלג את הרכיבים הפיסיים של המחשב. במהלך לימוד היחידה "יסודות מדעי המחשב" תעבדו בוודאי על מחשבים אישיים (personal computer – pc). מחשבים אישיים הם קטנים יחסית בממדיהם ומיועדים לשרת בו זמנית משתמש אחד בלבד.

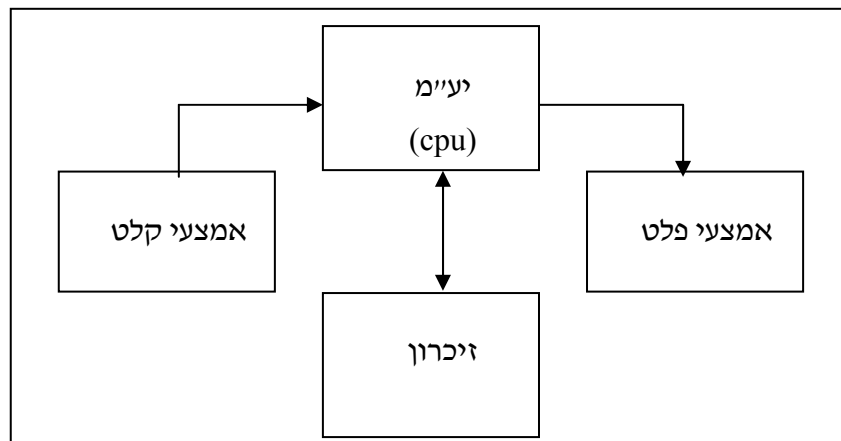
קיימים גם מחשבים גדולים יותר, שיכולים לשרת בו זמנית משתמשים רבים. עם זאת, המבנה הכללי של מחשב אינו תלוי בגודלו.

למחשב כמה יחידות בסיסיות:

- ◆ **יחידת עיבוד מרכזית (central processing unit)**, ובקיצור **יע"מ (CPU)** היא היחידה שאחראית על עיבוד מידע, מבצעת חישובים ומנהלת את כל התהליכים המתבצעים במחשב.
- ◆ **בזיכרון (memory)** נשמרים תוכניות המחשב, המידע שבו משתמש המחשב ותוצאות ביניים של תהליכי עיבוד.
- ◆ **דרך אמצעי קלט (input devices)** מתבצעת קליטת נתונים.
- ◆ **דרך אמצעי פלט (output devices)** ניתן הפלט של תוצאות העיבוד.

איור 1.1 מתאר את היחידות הבסיסיות של מחשב:

#### מחשב



איור 1.1 – יחידות הבסיסיות של מחשב

נרחיב מעט על היחידות הבסיסיות:

**יחידת העיבוד המרכזית** היא ה"מוח של המחשב". היא מבצעת את ההוראות הכתובות בתוכניות המחשב, ביניהן הוראות לביצוע פעולות חישוב, פעולות השוואה ועוד. במהלך ביצוע פעולותיה פונה יחידת העיבוד המרכזית אל הזיכרון ואל אמצעי הקלט והפלט.

**הזיכרון** מורכב מאוסף גדול מאוד של תאים. לכל תא בזיכרון מותאם מספר סידורי, הנקרא מען (כתובת, address). יחידת העיבוד המרכזית משתמשת במידע השמור בזיכרון ולעתים משנה אותו, תוך פנייה לתאי הזיכרון על ידי המען שלהם. יש שני סוגי פניות לזיכרון: כתיבת מידע בזיכרון וקריאת מידע מהזיכרון. תהליך הכתיבה בזיכרון גורם למחיקת מידע ולשמירת מידע חדש במקומו. לעומתו, תהליך הקריאה מהזיכרון גורם להעברת המידע מהזיכרון אל יחידת העיבוד המרכזית, אך איננו גורם למחיקתו. הוא מזכיר את תהליך הקריאה שאנו מבצעים: כאשר אנו קוראים ספר, השורות אינן נעלמות מדפי הספר, אלא רק "מועתקות" לזיכרונו.

המידע השמור בזיכרון מיוצג על ידי סיביות. **סיבית (bit)** – קיצור של **ספרה בינארית (binary digit)** – היא אחת מן הספרות 0 או 1, כמו שספרה עשרונית היא אחת מן הספרות 0, ..., 9.

תא בזיכרון המחשב מכיל סדרה של סיביות, בדרך כלל 16, 32 או 64 סיביות, על פי תכנון החומרה של המחשב. מפתיע ומרשים שכל המשימות המורכבות המתבצעות על ידי מחשב הן בסופו של דבר אוסף של פעולות על סדרות המורכבות מהספרות 0 ו-1! המחשב מפרש סדרות של סיביות כמספרים או כאותיות, לעתים סדרות של סיביות מפורשות כהוראות לביצוע או כטיפוסי מידע אחרים.

למעשה, יחידת העיבוד המרכזית, ה"מוח" של המחשב, מבצעת בסך הכול הוראות פשוטות כגון "קרא את סדרות הסיביות שנמצאות בתאי הזיכרון שמעניהם הם 13 ו-37, התייחס לכל סדרת סיביות כאל מספר, חבר אותם וכתוב את התוצאה בתא שמענו 116".

זיכרון המחשב מורכב למעשה משני חלקים נפרדים:

**זיכרון ראשי** (main memory) משמש לשמירת תוכניות בזמן ביצוען, ולשמירת נתונים ותוצאות ביניים של תוכניות שמתבצעות.

**זיכרון משני** (secondary memory) משמש לאחסון לזמן בלתי מוגבל של מידע ושל תוכניות מחשב.

הזיכרון הראשי קטן משמעותית מן הזיכרון המשני. מהירות הקריאה ממנו והכתיבה אליו גדולה הרבה יותר ממהירותן של פעולות אלו בזיכרון המשני. הזיכרון הראשי ממוקם צמוד ליחידת העיבוד המרכזית, ואילו הזיכרון המשני נמצא באמצעי אחסון חיצוניים כמו דיסק ודיסק קשיח. במהלך ביצועה של תוכנית מחשב, יחידת העיבוד המרכזית מעתיקה את התוכנית מהזיכרון המשני אל הזיכרון הראשי. במובנים רבים, ניתן להתייחס לזיכרון הראשי כאל זיכרון לטווח קצר, ואל הזיכרון המשני כאל זיכרון לטווח ארוך.

### שאלה 1.2

הביאו דוגמה מכיתת בית הספר לחפץ שניתן לדמות את השימוש בו לשימוש בזיכרון ראשי, ולעומתה דוגמה לחפץ שניתן לדמות את השימוש בו לשימוש בזיכרון משני.

**אמצעי קלט** משמשים להעברת נתונים אל המחשב מן המשתמשים בו. למשל, בדרך כלל מחוברים למחשב אישי עכבר ולוח מקשים. שניהם אמצעי קלט המשמשים להעברת נתונים. גם סורק דיגיטלי, מצלמה דיגיטלית או מיקרופון מחוברים למחשב הם אמצעי קלט.

**אמצעי פלט** משמשים להעברת מידע מן המחשב אל המשתמשים בו. למשל, בדרך כלל מחוברים למחשב אישי מסך ומדפסת. גם מקרן או רמקול מחוברים למחשב הם אמצעי פלט.

### שאלה 1.3

מחשבון נועד לבצע פעולות חשבון. מהו אמצעי הקלט למחשבון? מהו אמצעי הפלט למחשבון?

### שאלה 1.4

המחשב קולט מידע, מעבד אותו ונותן כפלט את תוצאת העיבוד. גם מוח האדם קולט מידע, מעבד אותו ופולט את תוצאת העיבוד.

א. הביאו דוגמאות לאיברים קולטי מידע ולאיברים פולטי מידע בגוף האדם.

ב. הביאו דוגמאות למידע השמור בזיכרון האדם.

## 1.3 תוכנה

תוכנה היא אוסף תוכניות מחשב. תוכניות מחשב מנחות את העיבוד המתבצע על ידי החומרה. קיימות תוכניות לביצוע חישובים מתמטיים, לניהול מאגרי מידע, לבקרה על תהליכים, להדמיית מערכות, לעיבוד תמלילים, למשחקים וליישומים שונים ורבים נוספים.

בנוסף לתוכניות המיועדות ליישומים שונים, יש בכל מחשב תוכנית מיוחדת הנקראת מערכת הפעלה, והיא מהווה את הקשר בין התוכנה לחומרה:

**מערכת הפעלה** היא תוכנית המנהלת את שאר התוכניות ומקצה לשימושן את משאבי החומרה השונים (יע"מ, זיכרון, אמצעי קלט ואמצעי פלט).

כל תוכנית מחשב (או בקיצור, תוכנית) נכתבת על ידי מתכנת בשפת תכנות. נסביר את שני המושגים האלה:

**שפת תכנות** (programming language) היא למעשה אוסף של כל הכללים הקובעים כיצד נכתבות ההוראות בתוכנית מחשב, ומה המשמעות של כל הוראה.

**מתכנת** (programmer) הוא אדם הכותב תוכניות בשפת מחשב. עבודתו של מתכנת – תהליך התכנות – כולל ניתוח של משימות המיועדות לביצוע במחשב, כתיבת מתכון לביצוע המשימה, ויישומו של המתכון בשפת מחשב.

כזכור, הפעולות המתבצעות ביחידת העיבוד המרכזית הן פעולות על סיביות. לכן, בעצם, השפה שבאמצעותה ניתן לתקשר עם מחשב, או השפה שבה ניתן לתת לו הוראות שיוכל "להבין", צריכה להיות שפה מאוד פשוטה, הכוללת הוראות לביצוע פעולות על סיביות. שפה כזאת נקראת שפת מכונה:

**שפת מכונה** (machine language) היא שפת תכנות הכוללת הוראות לביצוע פעולות פשוטות מאוד. כל הוראה בשפת מכונה מורה על ביצוע פעולות על סדרות של סיביות (למשל, חיבור שתי סדרות). למעשה, גם ההוראות של שפת מכונה נכתבות בקוד המבוסס על סיביות, ולכן תוכנית בשפת מכונה היא בעצם סדרה ארוכה של סיביות, כלומר, רצף ארוך של 0 ו-1.

לכל סוג מחשב שפת מכונה משלו.

התוכניות הראשונות שנכתבו עבור מחשבים (בסוף שנות ה-40 של המאה העשרים) נכתבו בשפת מכונה. תהליך הכתיבה של תוכניות אלו היה מסורבל מאוד ולא נוח, בגלל החסרונות הרבים שיש לכתיבה בשפת מכונה.

מהם החסרונות של כתיבה בשפת מכונה ?

◆ מאחר שתוכנית בשפת מכונה היא רצף ארוך של 0 ו-1, קשה מאוד לכתוב אותה וקשה עוד יותר לקרוא אותה, לעקוב אחר מהלך ביצועה ולהבין את מטרתה. חשוב להבין שתהליך התכנות לא מסתיים בדרך כלל עם כתיבת התוכנית: לעתים מתגלות שגיאות בתוכנית וצריך לתקנה, לפעמים צריך לעדכן אותה כדי להתאימה לדרישות חדשות של המשימה שהיא

מבצעת. לא תמיד התיקונים והעדכונים מתבצעים על ידי הכותב המקורי, ולכן לנוחות הקריאה וההבנה של תוכנית נתונה יש חשיבות רבה.

◆ לכל סוג מחשב יש שפת מכונה שמתאימה בדרך כלל רק לו. לכן לא ניתן לקחת תוכנית שנכתבה בשפת מכונה של מחשב מסוג אחד, ולהריץ אותה כמו שהיא במחשב מסוג אחר. מעבר בין סוגי מחשבים דורש כתיבה מחודשת של התוכנית.

חסרונות אלה הביאו לפיתוחן של שפות נוחות יותר לכתיבה, לקריאה ולשימוש. שפות כאלו פותחו החל מאמצע שנות ה-50 של המאה העשרים, והן נקראות שפות עיליות. ביניהן למשל השפות פסקל (pascal), ג'אווה (Java), ו-C.

**שפה עילית** (high level language) היא שפת תכנות, אשר ההוראות בה דומות למשפטים בשפה טבעית (כמו אנגלית) או לנוסחאות מתמטיות. למרות הדמיון לשפה טבעית, ההוראות אינן נכתבות בכתיבה חופשית, אלא על פי כללים מוגדרים, שנקראים **כללי התחביר** (syntax) של השפה.

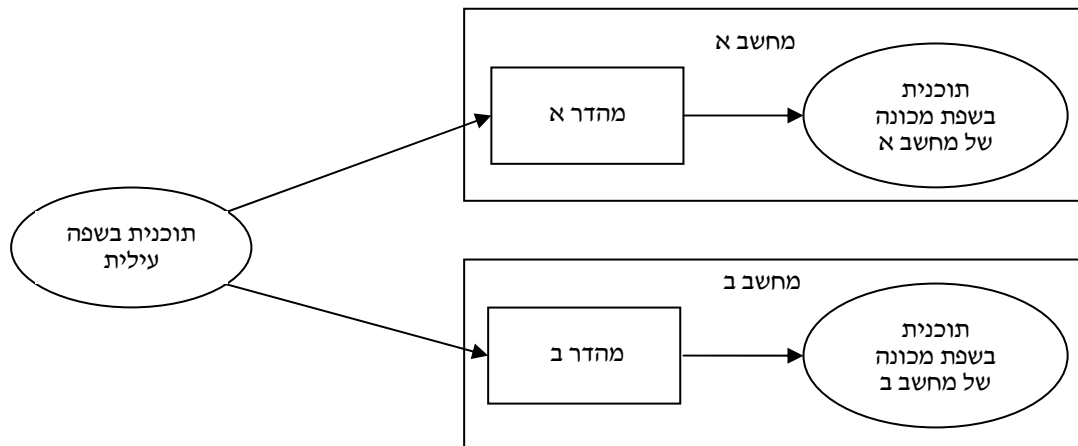
? אם המחשב "מבין" רק שפת מכונה, כיצד ניתן לגרום לו "להבין" תוכנית הכתובה בשפה עילית?

תוכנית הכתובה בשפה עילית עוברת תהליך של תרגום לשפת מכונה. התרגום נעשה על ידי תוכנית מחשב מיוחדת, הנקראת מהדר:

**מהדר** (compiler) היא תוכנית המתרגמת משפה עילית לשפת מכונה. תהליך התרגום נקרא **הידור** או **קומפילציה** (compilation). הקלט של המהדר הוא תוכנית בשפה עילית והפלט שלו הוא תוכנית בשפת מכונה, שהיא התרגום של תוכנית הקלט.

אם כך, כדי לבצע תוכנית מחשב הכתובה בשפה עילית צריכים להתבצע שני שלבים. קודם כל מתבצע שלב ההידור, במהלכו התוכנית בשפה העילית עוברת תרגום לשפת מכונה. רק אחר כך יכול להתבצע שלב ההרצה, במהלכו מתבצעת המשימה עצמה, כלומר, המחשב מבצע את ההוראות הכתובות בשפת מכונה, ונותן את תוצאת העיבוד כפלט.

לכל זוג של שפה עילית ושפת מכונה דרוש מהדר נפרד שיבצע את התרגום ביניהן. אם בכוונתנו להריץ במחשב מסוים תוכניות הכתובות בכמה שפות עיליות, עלינו לדאוג שבמחשב יהיה מותקן מהדר מתאים לכל אחת מהשפות העיליות האלו. גם ההיפך נכון: אם בכוונתנו להריץ תוכניות בשפה עילית מסוימת בכמה מחשבים מסוגים שונים, עלינו לדאוג שיהיו ברשותנו מהדר עבור כל סוג מחשב. איור 1.2 מדגים את הקשרים האלה.



איור 1.2 – הידור של תוכנית בשפה עילית במחשבים מסוגים שונים

אם כך, שפה עילית איננה רק יותר נוחה לקריאה ולכתיבה, אלא היא גם מגשרת על פני ההבדלים בין סוגים שונים של מחשבים. בעזרת מהדר מתאים ניתן לתרגם כל תוכנית בשפה עילית לתוכנית בשפת מכונה של מחשב זה או אחר.

תהליך ההידור מורכב משני שלבים:

1. בדיקת תחביר התוכנית בשפה העילית
2. תרגום התוכנית בשפה העילית לתוכנית בשפת מכונה.

בשלב 1, נערכת בדיקה כי התוכנית בשפה העילית עומדת בכללי התחביר של השפה שבה נכתבה. אם כתיבת התוכנית לא נעשתה בהתאם לכללי התחביר, יש בה **שגיאות תחביר** (syntax errors). הפלט של המהדר אחרי שלב 1 הוא פירוט שגיאות התחביר שמצא. לפני שניתן יהיה לתרגם את התוכנית יש לתקן את כל שגיאות התחביר שבה, כלומר, לעבור בהצלחה את שלב 1. למשל, תוכנית בשפת C# חייבת להכיל בתוכה לפחות פעם אחת את המילה class. אם ננסה לתרגם תוכנית בשפת C# שאינה מכילה את המילה class המהדר יודיע על שגיאת תחביר.

כאשר שלב 1 מסתיים בהצלחה, ואין בתוכנית שגיאות תחביר, מתבצע השלב השני ובו התוכנית מיתרגמת לשפת מכונה, ומתקבלת תוכנית שיכולה לרוץ במחשב.

גם במהלך הריצה של התוכנית עלולות להתגלות שגיאות שיגרמו לעצירת הריצה לפני סיומה המיועד, או להודעות שגיאה. אלו הן **שגיאות ריצה** (run-time errors), שאינן יכולות להתגלות בזמן ההידור. למשל, הניסיון לחלק ערך השמור בזיכרון המחשב ב-0 יגרום לשגיאת ריצה ולהדפסת הודעה מתאימה.

תהליך איתור שגיאות ריצה ותיקונן נקרא **ניפוי** (debugging). מקורו של המונח באנגלית בשלבים המוקדמים של שימוש במחשבים, כאשר מחשבים היו כה גדולים עד כי מחשב אחד מילא אולם שלם. מחשב מסוים חדל לפעול, ולאחר זמן התגלה בין רכיביו חרק גדול שנתקע שם והפריע לפעולתו התקינה של המחשב. מאז נוהגים לקרוא לשגיאה בתוכנית באג (bug – חרק באנגלית).

מאז תחילת פיתוח השפות העיליות, באמצע שנות ה-50 של המאה העשרים, פותח מספר גדול מאוד של שפות. עובדה זו מעוררת את השאלות הבאות:

◆ מדוע יש צורך בכל כך הרבה שפות?

◆ האם לא עדיפה שפה אחת אחידה שבה יכתבו כל התוכניות, ואשר אותה יוכל כל אחד ללמוד בקלות?

◆ מה מבדיל בין השפות השונות?

◆ מי משתמש באילו שפות ולאילו מטרות?

לריבוי השפות שתי סיבות עיקריות. סיבה אחת קשורה להתפתחות שפות תכנות כתגובה לצרכים המתעוררים בשטחים חדשים ושונים של יישומים. לכל שפה מאפיינים ייחודיים משלה: יש שפות המיועדות בעיקר לחישובים מדעיים, יש אחרות המתאימות יותר לעיבוד נתונים מנהלי (הפקת משכורות, הנהלת חשבונות וכו'). הסיבה השנייה היא התקדמות המחקר המדעי העוסק בשפות תכנות ומסייע בשיפור השפות.

את שפות התכנות ניתן לחלק לקבוצות על פי העקרונות המנחים את הכתיבה בשפות אלו. בספר זה ללימוד היחידה "ייסודות מדעי המחשב" נשתמש בשפת C#, השייכת לקבוצת השפות הקרויות **מונחות עצמים** (object oriented). קבוצות אחרות, שאליהן לא נתייחס ביחידה זו, הן קבוצת השפות ה**פרוצדורליות** (כמו פסקל או C), קבוצת השפות ה**פונקציונליות** (כמו scheme) וקבוצת השפות ה**לוגיות** (כמו פרולוג).

## 1.4 התפתחות המחשבים ומדעי המחשב

ההתפתחות הקשורה למחשבים נעשתה בשני מסלולים: ההתפתחות ההנדסית והטכנולוגית שאפשרה בניית מחשבים משוכללים יותר ויותר, וההתפתחות המדעית שניסתה להתמודד בצורה מדויקת, אפילו פורמלית לעתים, עם שאלות הקשורות לפתרון בעיות באמצעות מחשב. בסעיף זה ננסה לתת סקירה קצרה של שני מסלולי ההתפתחות, שכמובן אינם מנותקים זה מזה.

### התפתחות הנדסית וטכנולוגית – חומרה

ציון דרך חשוב בהתפתחות ההנדסית הוא באמצע המאה ה-17. אז פיתח המתמטיקאי הצרפתי בליז פסקל (Pascal) – על שמו נקראת שפת התכנות פסקל – מכונת חיבור וחסור. את המכונה בנה עבור אביו, כדי לסכם סכומי כסף לצורך גביית מסים. פסקל בנה את המכונה כדי לעזור לאנשים שביצעו בדרך כלל את החישובים הדרושים, ובמיוחד כדי להקטין את מספר הטעויות שנגרמו על ידי החישובים האנושיים. המכונה של פסקל מהווה ציון דרך חשוב משום שזו הייתה הפעם הראשונה ששולב מרכיב אוטומטי, באמצעות מכונה, בפעולת חישוב.

המכונה של פסקל זכתה לשיפורים כמה עשרות שנים מאוחר יותר. המדען הגרמני וילהלם לייבניץ (Leibnitz) בנה אף הוא מכונת חישוב. הוא העתיק את מנגנוני החיבור והחסור של מכונתו מהמכונה של פסקל, אך הוסיף גם חלק שמבצע כפל וחילוק. לייבניץ בנה את המכונה שלו משום שלטענתו המדע אמנם לא יכול להתקיים ללא חישוב, אך חישוב הוא פעולה חוזרת על עצמה, משעממת ולא יצירתית, וצריך להעביר את ביצועה למכונות.

בתחילת המאה ה-19, ב-1801, פיתח גם הצרפתי ז'וזף ז'אקאר (Jacquard) מכונה לביצוע אוטומטי של משימות. בניגוד למכונות של פסקל ולייבניץ אלו לא היו משימות חישוב מספריות, אלא משימות אריגה. מכונתו של ז'אקאר היתה למעשה נול אריגה מתוחכם, שיכול היה לארוג במגוון של דוגמאות. הדוגמאות השונות תוארו על ידי כרטיסים מנוקבים, לכל דוגמת אריגה תבנית ניקובים משלה. מנגנון בקרה מיוחד בתוך המכונה חש את הנקבים בכרטיס, ובהתאם לכך פיקח על פעולות המכונה, כגון בחירת חוטים.

התכנון של מה שנחשב היום המחשב הראשון הגיע כשלושים שנה מאוחר יותר. ב-1833, תוכננה לראשונה מכונה כללית יותר, שיכולה לבצע משימות מסוגים שונים. המתמטיקאי האנגלי צ'רלס בבג' (Babbage) תכנן את "המכונה האנליטית" שלו, מכונה שהיתה אמורה לבצע תוכניות שונות מסוגים שונים למטרות שונות. התוכניות היו אמורות להיות מקודדות, בדומה למכונה של ז'אקאר, בעזרת כרטיסים מנוקבים. התכנון של בבג', שכלל צירים, ידיות, גלגלי שיניים ורכיבים מכניים אחרים, לא מומש אף פעם. בניית חלקי המכונה דרשה דיוק טכני רב מדי, שלא ניתן היה להשגה באותו זמן. אבל, אף על פי שלא נבנתה, המכונה האנליטית היא בעלת חשיבות גדולה. למעשה, הרעיונות הגלומים בה הם הבסיס למבנה המחשבים של ימינו ולאופן פעולתם. אפילו בעצם נכתבה אז תוכנית עבור המכונה הלא-בנויה, תוכנית הראויה בהחלט להיחשב כתוכנית המחשב הראשונה בהיסטוריה! את התוכנית כתבה עדה לאבלייס (Lovelace) – על שמה נקראת שפת Ada.

לאחר מותו של בבג' מרכז הפעילות בבניית מכונות חישוב נדד לארצות הברית. העיסוק המוגבר בכך בארצות הברית נבע בין השאר מהצורך שהתעורר מהשטח: ב-1880 נערך בארצות הברית מפקד אוכלוסין, המפקד הבא נועד ל-1890, וב-1886, ארבע שנים לפני המפקד הבא, ושש שנים אחרי המפקד הקודם, עדיין לא סיימו לסכם את תוצאותיו, ומועד סיום הסיכום לא נראה באופק... הרמן הולרית (Hollerith), מהנדס שעבד כפקיד בלשכת מפקד התושבים, גילה יוזמה וב-



1886 הוא הציע כמה מכשירים שפיתח כדי לפתור את הבעיה. סיכום המפקד של 1890 כבר נעשה בעזרת פיתוחו של הולריק וארך לא יותר מחודש! גם פיתוחו של הולריק היה מבוסס על כרטיסים מנוקבים. בעקבות הצלחתו ראה הולריק כי טוב והקים חברה למכונות חישוב. מאוחר יותר, ב-1928, הרחיבה אותה חברה את פעילותה והפכה לחברה בינלאומית למכונות עסקיות, ושינתה את שמה בהתאם ל-International Business Machines, או בקיצור, IBM, המוכרת לנו היטב גם היום.

בשנת 1937 החל המדען האמריקני הווארד אייקן (Aiken), יחד עם חברת IBM, לבנות את המחשב האלקטרו-מכני הראשון. למעשה, אייקן הגשים את חלומו של בבג: התכנון שלו התבסס על רעיונותיו של בבג, ונעזר במכשירים החשמליים והאלקטרו-מכניים אשר בתקופתו של אייקן כבר היו זמינים. בניית המחשב הושלמה ב-1944. שמו היה MARK I וגודלו היה כגודל אולם התעמלות! למעשה, במקביל לבניית MARK I, במהלך מלחמת העולם השנייה, בנו גם הבריטים מחשב, בשם אניגמה (Enigma), שבעזרתו פיצחו צפנים של הגרמנים. אלא שעקב תפקידו הרגיש נשמר קיומו של Enigma בסוד במשך זמן רב. זמן קצר אחר-כך, ב-1946, כבר הושלמה בנייתו של מחשב מהיר בהרבה מ-MARK I. הוא נקרא ENIAC, ותוכן על ידי האמריקנים אקרט (Eckert) ומוצ'לי (Mauchly). הוא היה הרבה יותר מהיר משום שלא התבסס בכלל על תנועות מכניות, ולכן נחשב המחשב האלקטרוני הראשון. אורכו היה שלושים מטרים, רוחבו מטר אחד, גובהו שלושה מטרים, והוא הכיל 18,000 שפופרות ואקום. צריכת החשמל של ה-ENIAC, שהוצב בעיר פילדלפיה, הייתה כה גבוהה, עד שנהגו אז לומר שבכל פעם שהופעל התעממו האורות בעיר כולה!

אותם מחשבים ראשונים השתמשו בסרטי נייר מנוקבים. עבור כל עיבוד קודדו על סרט כזה גם התוכנית לביצוע וגם נתוני הקלט עבורה. כלומר, אם ביצעו אותה תוכנית כמה פעמים, היה צריך להזין למחשב את סרט הנייר שעליו קודדה בכל פעם ופעם. ב-1946 הציע המדען ההונגרי-אמריקני ג'ון פון-נוימן (von Neuman) לשמור את התוכניות בזיכרון המחשב. כתוצאה מכך, מהירות תהליכי העיבוד השתפרה משמעותית. המחשב התעשייתי הראשון שפעל לפי עקרון זה נבנה ב-1951, ונקרא UNIVAC. עקרון זה של פון-נוימן משמש למעשה גם במחשבים של ימינו.

מכאן החלה האצה בקצב ההתפתחות הטכנולוגית. בשנות ה-50 וה-60 של המאה העשרים הוחלפו שפופרות הריק בטרנזיסטורים, ושינוי זה הביא להקטנה בממדי המחשבים, להקטנה בצריכת ההספק החשמלי שלהם ולהגדלה במהירות פעולתם. בנוסף למחשבים הגדולים (mainframes) שהיו עד אז, נבנו גם מחשבים בינוניים (שגודלם כגודל כוננית ספרים) שנקראו מחשבי מידי ומחשבי מיני (midi/mini computers). זמן לא רב אחר-כך קטנו המחשבים אף יותר: בסוף שנות ה-60 ובתחילת שנות ה-70 פותחה טכנולוגיית המעגל המשולב (integrated circuit) ובעקבות כך ירד מחיר המחשבים, מהירותם עלתה ונבנו אפילו מחשבים זעירים, שגודלם לא עלה על גודל קופסת גפרורים – המיקרו-מחשבים (micro computers). משום שהיו כה קטנים וזולים יחסית, החלו להשתמש בהם הרבה לצרכים מגוונים למשל כבקרים במערכות אלקטרוניות שונות. המחשב האישי, המוכר לנו היום, נבנה לראשונה בסוף שנות ה-70 והוא היה מבוסס על מיקרו-מחשב. קפיצת הדרך הזאת, שנעשתה במשך תקופה קצרה יחסית, היא כמעט בלתי נתפסת: היום יש מיקרו מחשבים שכושר העיבוד שלהם עולה בהרבה על זה של אותם מחשבי הענק הראשונים.

## **התפתחות הנדסית וטכנולוגית – תוכנה**

במקביל להתפתחות הטכנולוגית של החומרה, המכונות עצמן, חלה התפתחות גם בתחום התוכנה. המחשבים הראשונים תוכנתו בשפת מכונה. כפי שהזכרנו, כתיבת תוכניות כאלו הייתה כרוכה באי-נוחות רבה. אי-נוחות זו הביאה באמצע שנות ה-50 לפיתוחה של שפת התכנות העילית

הראשונה, פורטרן (Fortran). משום שבאותה תקופה הייתה עלייה בביקוש לתוכניות מחשב המבצעות חישובים מתמטיים, פורטרן הותאמה לכתיבה של חישובים כאלה. אבל היא לא הייתה נוחה לכתיבת תוכניות לניהול מאגרי מידע (ניהול כוח אדם, ניהול מלאי וכו'), ומשום כך פותחה בעקבותיה, בתחילת שנות ה-60, שפת קובול (Cobol). באותה תקופה פותחה שפה נוספת, ליספ (Lisp) שהתאימה לצרכים אחרים. ממנה נגזרה מאוחר יותר שפת לוגו (LOGO) המשמשת בדרך כלל כשפה לימודית לפיתוח הרגלי חשיבה בפתרון בעיות.

מאז פותחו שפות עיליות רבות לצרכים שונים ומגוונים. למשל, שפת בייסיק (Basic) פותחה באמצע שנות ה-60, למטרות לימודיות. פסקל (Pascal) פותחה אף היא כשפה לימודית, בתחילת שנות ה-70, ובאותה תקופה פותחה גם שפת C, שנועדה לכתיבת מערכות הפעלה. ב-1970 פותחה שפת פרולוג (PROLOG) שגישת התכנות בה מבוססת על כללים לוגיים. עם עליית הצורך בכתיבת תוכניות גדולות מאוד ומורכבות מאוד, פותחו בשנות ה-80 שפות שנועדו במיוחד לפיתוח תוכניות גדולות, כגון Modula ו-Ada. בעקבותיהן פותחו שפות נוספות שהתמקדו בפיתוח תוכניות גדולות, והתבססו על מה שקרוי תכנות מונחה-עצמים. בין אלו ניתן למצוא את ++C, את SmallTalk, את Ada95, את Eiffel ואת Java.

## התפתחות מדעית

כאמור, במקביל להתפתחות ההנדסית והטכנולוגית, הן בתחום החומרה והן בתחום התוכנה, חלה גם התפתחות מדעית. החל מאמצע שנות ה-30 של המאה העשרים (עוד לפני שנבנה המחשב הראשון!), נעשתה עבודה תיאורטית חשובה, שהניחה את הבסיס לתחום המדעי הקרוי היום מדעי המחשב. עבודה זו נעשתה על ידי מתמטיקאים, שניסו להגדיר בצורה מתמטית מהו תהליך של חישוב, ולנתח בצורה מתמטית, פורמלית ומדוייקת, את מגבלותיהם של תהליכי חישוב, ובפרט את מגבלותיהן של מכוונות המבצעות תהליכי חישוב. בין המתמטיקאים האלה ניתן למנות את אלן טיורינג (Turing) האנגלי, שהיה מעורב מאוחר יותר בפרויקט האניגמה, את קורט גדל (Gödel) הגרמני, את אנדריי מרקוב (Markov) הרוסי, ואת האמריקנים אלונזו צ'רץ' (Church), אמיל פוסט (Post) וסטיבן קלין (Kleene). חוקרים אלה ידעו להצביע כבר אז על כך שיש בעיות חישוביות שלא יצליחו לעולם להיפתר על ידי מכונה חישובית, וזאת כאמור עוד לפני שנבנה המחשב הראשון.

מאז חלה התפתחות מדעית עצומה, כאשר לעתים ההתפתחויות הטכנולוגיות הניעו וזירזו התפתחויות מדעיות ולעתים דווקא ההתפתחויות המדעיות גרמו להתפתחות טכנולוגית. כיום קיימת מחלקה למדעי המחשב כמעט בכל מוסד אקדמי, והפעילות המחקרית במדעי המחשב היא רבה ומגוונת: תורת החישוביות העוסקת באפיון של בעיות שניתנות או לא ניתנות לפתרון; תורת הסיבוכיות העוסקת באפיון של בעיות על פי כמות המשאבים (זמן וזיכרון) הנדרשים לפתרונן; קריפטוגרפיה העוסקת בהצפנות מסוגים שונים; חישוב מקבילי ומבוזר, העוסק בפתרון בעיות שנועדו להתבצע במערכות שבהן כמה מחשבים עובדים ביחד לפתרון משימה אחת; תורת התקשורת העוסקת באפיונים של רשתות תקשורת (כמו האינטרנט) ופתרון בעיות הקשורות לרשתות תקשורת; בינה מלאכותית העוסקת במערכות שנועדו לדמות פעילות אנושית, ועוד תחומים רבים נוספים.

במסגרת לימודי מדעי המחשב בבית הספר התיכון ניתן כמובן להציג רק מקצת מתחומי הפעילות השונים במדעי המחשב. בכל זאת תוכנית הלימודים התיכונית במדעי המחשב נוגעת במגוון רחב למדי של נושאים, גם בחלק מאלה שהוזכרו לעיל.

## סיכום

בפרק זה תיארונו בקצרה מהו מחשב, מהן היחידות הבסיסיות שמהן הוא בנוי, וכיצד נכתבות ומתבצעות תוכניות מחשב. תיארונו גם את ההתפתחות ההנדסית והטכנולוגית של המחשבים ואת ההתפתחות המדעית של תחום מדעי המחשב.

**מחשב** הוא מכונה אלקטרונית הקולטת נתונים, מעבדת אותם ופולטת מידע שנוצר בתהליך העיבוד.

הנתונים שקולט המחשב נקראים **קלט**.

המידע שפולט המחשב נקרא **פלט**.

העיבוד המבוצע במחשב מונחה על ידי קבוצת הוראות הנקראת **תוכנית מחשב**.

הרכיבים הפיסיים של המחשב נקראים **חומרה** ואוסף תוכניות המחשב נקרא **תוכנה**. החומרה מחולקת לכמה רכיבים בסיסיים: יחידת עיבוד מרכזי, זיכרון, אמצעי קלט ואמצעי פלט. תוכנה של מחשב כוללת בין השאר את מערכת ההפעלה, את המהדרים ותוכניות ליישומים שונים.

**יחידת העיבוד המרכזית** מנהלת את כל התהליכים המתבצעים במחשב.

**הזיכרון** שומר מידע, תוכניות, ותוצאות ביניים של תהליכי עיבוד. הזיכרון מתחלק ל**זיכרון משני** ול**זיכרון ראשי**. המידע השמור בזיכרון מיוצג באמצעות סיביות (**סיבית** – אחת מן הספרות 0 או 1).

**אמצעי הקלט** אחראים על קליטת נתונים ו**אמצעי הפלט** אחראים על פליטת מידע.

תוכנית מחשב נכתבת על ידי **מתכנת** בשפת תכנות. **לשפת תכנות** כללים המכתיבים את אופן כתיבת ההוראות בתוכנית. כיום נכתבות תוכניות מחשב בשפה עילית.

**בשפה עילית** המשפטים דומים למשפטים בשפה טבעית, כמו אנגלית. **שפת מכונה** היא השפה אותה "מבין" המחשב, וההוראות בה מקודדות על ידי סיביות.

לפני ביצוע תוכנית בשפה עילית עליה לעבור שני שלבים: הידור והרצה. **הידור** (קומפילציה) הוא התהליך של תרגום תוכנית בשפה עילית לשפת מכונה. תהליך זה מתבצע על ידי **מהדר** (קומפילר).

**שגיאות תחביר** מתגלות בשלב ההידור. **שגיאות ריצה** מתגלות בזמן ההרצה. תהליך איתור שגיאות ריצה ותיקונן נקרא **ניפוי שגיאות**.

## שאלות נוספות

1. מדוע אי-אפשר לכתוב תוכנית למחשב בשפה טבעית, כמו אנגלית או עברית?
2. מדוע שפת תכנות עילית נקראת בשם זה?
3. למה מתכוונים כאשר אומרים כי מחשב מבין שפת מכונה?
4. כמה מהדרים דרושים להידור תוכנית בשפת C, תוכנית בשפת פסקל, ותוכנית בשפת בייסיק בשלושה מחשבים מסוגים שונים?