

המכללה הטכנולוגית של חיל האוויר
באר – שבע

מגמת מכונות – התמחות מכטרוניקה

חוברת עבודה בתכנות

כתב וערך: שמעוני גבי – תשע"ז

תוכן עניינים

3	הקדמה
4	הדפסות מידע למסך המחשב
7	הגדרה ושימוש במשתנים בתוכנית
12	קבלת נתונים מהמשתמש דרך מקלדת המחשב
15	קבלת החלטות בתוכנית בעזרת תנאי if

במסגרת הלימודים בהתמחות מכטרוניקה מבצעים התלמידים פרויקטים בהם הם נדרשים לתכנת בקר מסוג Arduino. העבודה על הפרויקטים נמשכת בכיתה י"א וי"ב וכוללת תכן מכני, בקרה, בניית מעגלים אלקטרוניים ותכנות בקר שמיישם את כל תהליך הבקרה של המערכת המפותחת.

כדי להקל על האתגרים העומדים בפני התלמידים בכיתה י"א, הוקדם חלק הלימודים הקשור לתכנות לכיתה י"א לאפשר לתלמידים להיחשף ולהתמודד עם עולם התכנות בנפרד מכל אתגרי הפרויקט.

בחוברת זו אספתי את הנושאים המרכזיים שנלמדים במקצוע תכנות בקר Arduino והכנתי תרגילים שחלקם יבוצעו בכיתה והיתר יבוצעו בזמנו הפנוי של הלומד בבית.

חשוב! זו אינה חוברת שמלמדת תכנות. בחוברת יהיה תקציר של כל פקודה או פונקציה שנלמדה, תוכנית דוגמא בסיסית ושורה של תרגילים ברמות קושי הולכות וגדלות.

כדי להקל על אתגר הפרויקט מומלץ לנסות ולעשות כמה שיותר תרגילים מתוך החוברת מעבר לאלו שיבוצעו בכיתה או ינתנו כעבודת בית.

במקביל לחוברת זו ישנו אתר אינטרנט המלווה את תוכנית הלימודים ובו יש יותר חומר הסבר ודוגמאות לכל דורש. כתובת האתר היא : <http://007arduino.info>

בהצלחה

שמעוני גבי

הסבר

כתיבת מידע למסך המחשב מחייבת שהבקר יהיה מחובר למחשב בעזרת כבל USB. מערכת בקרה אמיתית לרוב אינה מחוברת קבוע למחשב מאחר והיא מותקנת בכל מיני מקומות ומצבים שלא מאפשרים חיבור למחשב.

הדפסת מידע למסך המחשב הוא כלי מרכזי בו נעשה שימוש במהלך פיתוח התוכנה. כלי זה מאפשר לי לעקוב אחר התקדמות התוכנית והחישובים שהיא מבצעת בזמן ריצה. פעולת מעקב זו נקראת בשפה המקצועית debug או ניפוי שגיאות.

לדוגמא: בסמרטפון של כל אחד יש חיישן מרחק בראש המסך שמודד מרחק מעצם מתקרב. בזמן ביצוע שיחה ראשו של המשוחח קרוב מאוד לחיישן וזה גורם לכיבוי המסך וחיסכון בסוללה. במהלך פיתוח התוכנית שמיישמת עיקרון זה רוצים לראות האם החיישן מודד ערכים נכונים ולכן מחברים את המערכת למחשב ומדפיסים את הערכים שמודד החיישן למסך. אם הערכים אינם נכונים, יבוצע תהליך ניתוח הבעיה, זיהוי הגורם ותיקון.

החיבור של הבקר למחשב בעזרת כבל USB נקרא חיבורי טורי – Serial. כדי לשלוח מידע מהבקר למסך המחשב וחזרה אנחנו משתמשים בספריה שנקראת Serial והיא מכילה כלים תכנותיים ליצירת הקשר והעברת המידע. כשמשתמשים באחד מהכלים האלה בתוכנית, מתחילים לכתוב את שם הסיפריה, אחר כך שמם נקודה ואחר כך שם הפונקציה או הכלי בשימוש עם פרמטרים רלוונטים המופיעים בין סוגרים עגולים.

יצירת קשר בין הבקר למחשב:

```
Serial.begin(9600);
```

begin – זוהי פונקציה שמאתחלת את הקשר בין הבקר ולמחשב דרך כבל USB. היא מגדירה שהמידע שיועבר ישלח במהירות של 9600 ביט לשנייה (כמות זכרון לשנייה). ישנם מצבים בהם תוגדר מהירות אחרת.

חשוב! יש להקפיד על אותיות גדולות וקטנות בכתיבת הפקודה.

כל שורה מסתיימת בסימן ; (נקודה פסיק) מלבד שורות שמסתיימות בסימן } (סוגריים מסולסלים)

יש לוודא שגם בצד המחשב מהירות הקליטה תתאים למהירות המוגדרת בתוכנית. הגדרת מהירות הקליטה נעשית במסך Serial Monitor בסביבת הפיתוח.

הדפסת מידע למסך :

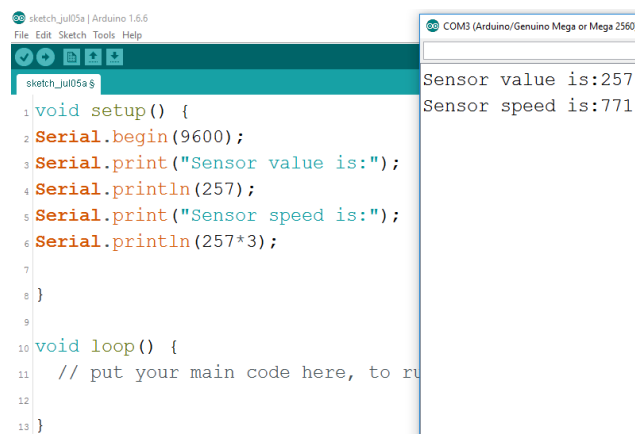
`Serial.print(text);`

`Serial.println(text);`

`print` או `println` - שתי פונקציות אלה שולחות אל מסך המחשב את התוכן אותו נרצה להציג. ניתן לשלוח משפט שתחום בין גרשיים, למשל: "Sensor Value is=" (אין תמיכה בעברית בשליחת מידע למסך המחשב!). ניתן לשלוח ערכים מספריים או תוצאות של פעולה חשבונית, למשל: 127 או $4+2*3$. ניתן להדפיס גם תוכן של מידע ששמרנו בזיכרון בתוך משתנים (יורחב בהמשך).

הפונקציה `print` מדפיסה את המידע ששלחנו ומשאירה את הסמן על המסך בקצה השורה וממתינה לפקודת ההדפסה הבאה. הפונקציה `println` מדפיסה את המידע ששלחנו ומורידה את הסמן לתחילת השורה הבאה (כמו לחיצה על כפתור ENTER במקלדת) וממתינה לפקודת ההדפסה הבאה.

תוכנית לדוגמא



The screenshot shows the Arduino IDE interface. On the left, the sketch editor displays the following code:

```
1 void setup() {  
2   Serial.begin(9600);  
3   Serial.print("Sensor value is:");  
4   Serial.println(257);  
5   Serial.print("Sensor speed is:");  
6   Serial.println(257*3);  
7  
8 }  
9  
10 void loop() {  
11   // put your main code here, to run over and over again  
12  
13 }
```

On the right, the Serial Monitor window is open, showing the output of the sketch:

```
Sensor value is:257  
Sensor speed is:771
```

לכל אחד מהסעיפים כיתבו תוכנית נפרדת, הריצו ובידקו שהתקבלה התוצאה הנדרשת. בתוכניות יש להשתמש בפונקציות print או println לפי הצורך.

1. תוכנית המדפיסה מילה המצוייה בין גרשיים. מה קיבלת?
2. תוכנית המדפיסה מילה בלי גרשיים. מה קיבלת?
3. תוכנית המדפיסה משפט בן 3 מילים לבחירתכם בעזרת פקודת print אחת
4. תוכנית המדפיסה משפט בן 3 מילים לבחירתכם בעזרת 3 פקודות print, אחת לכל מילה תוך שמירה על רווחים בין המילים
5. תוכנית המדפיסה משפט בן 3 מילים לבחירתכם כך שהמילה הראשונה תהיה בשורה נפרדת משתי המילים האחרות
6. תוכנית המדפיסה משפט בן 3 מילים לבחירתכם כך ששתי המילים הראשונות יהיו בשורה נפרדת מהמילה השלישית
7. תוכנית המדפיסה משפט בן 3 מילים לבחירתכם כך שכל מילה תהיה בשורה נפרדת
8. תוכנית המדפיסה פעולה חשבונית המצוייה בין גרשיים (למשל: "17+5")
9. תוכנית המדפיסה תוצאת פעולה חשבונית (למשל: 17+5)
10. תוכנית המדפיסה פעולה חשבונית המצוייה בין גרשיים ואת תוצאתה באותה שורה על המסך (רמז: יש לבצע בעזרת שתי פקודות)
11. **אתגר:** תוכנית המדפיסה מספרים לבחירתכם המאורגנים בטבלה בת 3 עמודות ו-4 שורות כך שכל ספרות היחידות יהיו מיושרים לימין
12. תוכנית המדפיסה ערך מספרי עשרונים בדיוק של 4 מקומות אחרי הנקודה
13. תוכנית המדפיסה ערכים בבסיס 2 או בבסיס 16
14. תוכנית המדפיסה משפט אחד בלולאה אינסופית בתור שורה אחת
15. תוכנית המדפיסה משפט אחד בלולאה אינסופית בשורות נפרדות

הגדרה ושימוש במשתנים בתוכנית

הסבר

משתנים בתכנות הם כלי לגשת לזכרון הבקר ולשמור שם מידע או לשלוף משם מידע ששמרנו קודם. הזיכרון בנוי כמו מגירות ובכל מגירה ניתן לשמור רק סוג אחד של מידע, למשל: מגירה לשמירת מספרים שלמים, מגירה לשמירת מספרים עשרוניים, מגירה לשמירת אותיות. בכל מגירה שומרים רק מספר אחד ולכן יכולות להיות מספר רב של מגירות בהם שומרים אותו סוג מידע אבל שונה.

אנחנו נתרכז בשלוש סוגי מספרים בשלב ראשון (בהמשך לפי הצורך נכיר סוגים נוספים)

מספר שלם (קטן) integer –

משתנה זה מאפשר לשמור בתוכו מספרים שלמים בין התחום -32768 ועד 32767. המגבלה הזו קיימת בגלל גודל הזכרון המוקצה לשמירת המספר. חשוב! למספר זו תכונה מעגלית, אם נגדיל את הערך המקסימלי 32767 באחד, נגיע אל הערך המינימלי -32768. אם נפחית אחד מהערך המינימלי -32768 נגיע לערך המקסימלי 32767.

כדי להודיע לתוכנית שברצוני לשמור מספר שלם בזכרון אני צריך לכתוב את השורה הבאה:

```
int varName;
```

int – זה מציין שבכוונתי להקצות מקום בזכרון למספר שלם מסוג integer.

varName – זהו השם שנתתי למשתנה ובעזרתו אני יכול לגשת אל המקום שהוקצה בזכרון כדי לשמור בו ערך או כדי לקרוא את הערך ששמור בו. שם המשתנה יכול להיות מורכב מאותיות, ספרות והתו _ . באתר האינטרנט של המקצוע ניתן לראות מידע מפורט על המגבלות במתן שמות.

כדי לשמור ערך מספרי בתוך משתנה כותבים שורה כמו זו:

```
varName = 53;
```

שימו לב לסיום השורה עם ;

מספר שלם (גדול) – long

משתנה זה מאפשר לשמור בתוכו מספרים שלמים בין התחום $-2,147,483,648$ ועד $2,147,483,647$.

כדי להודיע לתוכנית שברצוני לשמור מספר שלם בזכרון אני צריך לכתוב את השורה הבאה:

```
long varName;
```

– long זה מציין שבכוונתי להקצות מקום בזכרון למספר שלם מסוג long integer. הוא מקצה בזכרון הבקר יותר מקום כדי לאכסן מספרים גדולים יותר.

varName – זהו השם שנתתי למשתנה ובעזרתו אני יכול לגשת אל המקום שהוקצה בזכרון כדי לשמור בו ערך או כדי לקרוא את הערך ששמור בו.

כדי לשמור ערך מספרי בתוך משתנה כותבים שורה כמו זו:

```
varName = 53L;
```

שימו לב לאות L המופיעה מימין לערך 53, תפקידה הוא להודיע לבקר לשמור את המספר בסוג long למרות שגודל המספר מתאים ל-int.

מספר עשרוני float –

משתנה זה מאפשר לשמור בתוכו מספרים עשרוניים (כאלה שיש להם חלק שקטן מאחד מימין לנקודה העשרונית) בין התחום -3.4×10^{38} ועד 3.4×10^{38} .

כדי להודיע לתוכנית שברצוני לשמור מספר עשרוני בזכרון אני צריך לכתוב את השורה הבאה:

```
float varName;
```

– float זה מציין שבכוונתי להקצות מקום בזכרון למספר עשרוני מסוג float. הוא מקצה בזכרון הבקר יותר מקום כדי לאכסן מספרים גדולים יותר.

varName – זהו השם שנתתי למשתנה ובעזרתו אני יכול לגשת אל המקום שהוקצה בזכרון כדי לשמור בו ערך או כדי לקרוא את הערך ששמור בו.

כדי לשמור ערך מספרי בתוך משתנה כותבים שורה כמו זו:

```
varName = 1547.175;
```

```
1  // כאן מגדירים משתנים גלובליים
2  int a;
3  int b=10;
4  int c,d,e;
5  long m=100L;
6  float q=12.75;
7
8  void setup() {
9  // כאן מגדירים משתנים לוקלים
10 int a1 = 4;
11 int a2 = 6;
12 float myFloatVar = 125.365;
13 long loNgVar = 572L;
14
15 Serial.begin(9600);
16
17 Serial.println(a1);
18 Serial.println(a1+a2);
19 a2=a1*5;
20 Serial.println(a2);
21 q=a1/a2;
22 Serial.println(q);
23
24 }
25
26 void loop() {
27
28 // כאן מגדירים משתנים לוקלים
29 int b1 = 19;
30 int b2 = 2546;
31 float myFloatVar1 = 5624.12458;
32 long loNgVar1 = 32900;
33
34 Serial.println(c);
35 Serial.println(d+e);
36 e=c*d;
37 Serial.println(e);
38 myFloatVar1=q*b;
39 Serial.println(myFloatVar1);
40
41 }
```

לכל אחד מהסעיפים כיתבו תוכנית נפרדת, הריצו ובידקו שהתקבלה התוצאה הנדרשת.

1. תוכנית הכוללת ב-setup הגדרת משתנה int לוקלי ללא אתחול והדפסת ערכו
2. תוכנית הכוללת ב-setup הגדרת משתנה int לוקלי עם אתחול בשורת ההגדרה והדפסת ערכו
3. תוכנית הכוללת ב-setup הגדרת משתנה int לוקלי ומתן ערך למשתנה בתוך התוכנית והדפסת ערכו
4. תוכנית הכוללת ב-setup הגדרת משתנה int לוקלי ומתן ערך למשתנה בתוך התוכנית והדפסת ערכו בתוך loop.
5. תוכנית הכוללת ב-loop הגדרת משתנה int לוקלי ומתן ערך למשתנה בתוך התוכנית והדפסת ערכו
6. תוכנית הכוללת הגדרת משתנה גלובלי int עם אתחול והדפסת ערכו בשתי הפונקציות הראשיות (setup ו-loop)
7. תוכנית הכוללת ב-setup הגדרת שני משתנים int לוקלים עם אתחול וביצוע פעולה חשבונית על אחד המשתנים כך שהתוצאה תשמר במשתנה השני ותדפיס את ערכו למסך
8. תוכנית הכוללת ב-setup הגדרת שני משתנים int לוקלים עם אתחול. התוכנית תבצע ב-loop פעולה חשבונית על אחד המשתנים כך שהתוצאה תשמר במשתנה השני ותדפיס את ערכו למסך
9. תוכנית הכוללת הגדרת שני משתנים int גלובלים עם אתחול. התוכנית תבצע ב-loop פעולה חשבונית על אחד המשתנים כך שהתוצאה תשמר במשתנה השני ותדפיס את ערכו למסך
10. תוכנית הכוללת הגדרת 3 משתנים int גלובלים. התוכנית תבצע ב-setup פעולה חשבונית עם שני משתנים ותשמור את התוצאה בשלישי. התוכנית תדפיס את הערך בפונקציה loop
11. תוכנית הכוללת הגדרת משתנה int, אתחול לערך מקסימילי של int, ביצוע פעולה חשבונית שתוצאתה חורגת מגבולות הערכים של int והדפסתה
12. תוכנית הכוללת הגדרת משתנה long, אתחול לערך מקסימילי של int, ביצוע פעולה חשבונית שתוצאתה חורגת מגבולות הערכים של int והדפסתה
13. תוכנית הכוללת הגדרת משתנה long ואתחול, ומשתנה int. ביצוע פעולה עם long ושמירה ב-int והדפסתה
14. תוכנית הכוללת ביצוע פעולה של long ו int יחד - הוספת L אחרי ערך מספרי. הדפסת התוצאה

15. תוכנית הכוללת ביצוע פעולה חשבונית שתוצאתה לא מספר שלם והדפסתו.
16. תוכנית הכוללת הגדרת משתנה float, ביצוע פעולה חשבונית שתוצאתה לא מספר שלם והדפסתו.
17. תוכנית הכוללת ביצוע פעולות בין משתנה float לבין משתנה int ושמירת התוצאה במשתנה int והדפסתו (למשל: $7.25 * 17$)
18. תוכנית הכוללת ביצוע פעולות בין משתנה float לבין משתנה int ושמירת התוצאה במשתנה float והדפסתו (למשל: $7.25 * 17$)
19. **אתגר:** תוכנית הכוללת ביצוע type cast של משתנה int למשתנה float – המרת סוג הערך לסוג אחר (למשל: הפיכת השלם 10 למספר עשרוני 10.00) שמירת התוצאה במשתנה int והדפסתו. חזור על התוכנית אך הפעם שמירת התוצאה במשתנה float והדפסתו.
20. **אתגר:** תוכנית לחישוב שורשים של משוואה ריבועית על ידי שימוש בפעולות מתמטיות מתקדמות pow, sqrt. הגדירו משתנים לפי הנדרש, הדפיסו תוצאות ביניים ותוצאות סופיות

קבלת נתונים מהמשתמש דרך מקלדת המחשב

הסבר

תהליך בקרה מורכב משלושה שלבים : קבלת מידע, עיבוד מידע והפעלת רכיבים. תכנות הבקר מהווה את ישום השלב השני בתהליך הבקרה. במערכת בקרה מציאותית מקבל הבקר מידע מחישנים. בשלב זה אנחנו עוד לא מכירים חישנים ולכן נדמה קבלת מידע על ידי הקלדת מספרים במקלדת המחשב וקליטתם על ידי הבקר לטובת עיבוד המידע והדפסת התוצאות למסך.

קבלת המידע מהמחשב נעשית גם כן על ידי שימוש בחיבור USB בין הבקר למחשב, וכמו שראינו קודם, חובה עלינו לאתחל את הגדרות החיבור לבקר על ידי שימוש בפקודה :

```
Serial.begin(9600);
```

תהליך קבלת המידע מורכב מכמה שלבים :

- האזנה לקו התקשורת כדי לזהות מידע שמגיע
- בדיקה כמה מספרים ממתנינים לקריאה
- קריאת המידע שהגיע ושמירתו בתוך משתנים
- עיבוד המידע

חשוב! בשלב זה לא יוסברו כל הפעולות לקריאת נתונים מהמקלדת. יוסבר רק מה צריך לעשות כדי לקבל את המידע ולשמור אותו.

תוכנית לדוגמא

```
1 int num = 0;
2 void setup() {
3   Serial.begin(9600);
4 }
5
6 void loop() {
7   if (Serial.available() > 0) {
8     num = Serial.parseInt();
9   }
10  Serial.println(num);
11 }
```

`Serial.available()` – זו פונקציה בספריית `Serial` המחזירה לנו את כמות המספרים שממתנינים לקריאה, כלומר אם הקלדתי את המספר 10 אז הפונקציה תחזיר לי 1 מכיוון

שיש רק מספר אחד הממתין לקריאה. אם עכשיו אקליד מספר נוסף (בלי שקראתי את הקודם) הפונקציה תחזיר לי 2 מכוון שיש שני מספרים שממתינים לקריאה.

כאשר אקרא את המספר הראשון מבין השנים, ערך הפונקציה יקטן ל-1 מכוון שנשאר רק עוד ערך אחד שלא קראנו. כשנקרא את הערך השני, הפונקציה תחזיר לי אפס, כי אין שום מידע שממתין לקריאה.

בתוכנית הדוגמא בשורה מספר 7 כתוב: `אים הפונקציה Serial.available()` מחזירה ערך שגדול מאפס (כלומר יש מידע הממתין לקריאה) אז תבצע את הפקודה בשורה 8. אם הפונקציה `Serial.available()` מחזירה ערך שווה לאפס, אז התוכנית תמשיך לשורה 10. הסבר מפורט של שימוש בתנאי `if` יוסבר בהמשך.

`Serial.parseInt()` - זו פונקציה בספריית `Serial` המחזירה לנו את הערך שקראנו מהמקלדת כמספר מסוג `int`. אם הקלדנו ערכים שאינם מתאימים לסוג `int` נקבל ערכים שונים לפי המרה מסויימת שהבקר מבצע ולא תוסבר כאן.

הדבר החשוב כאן, שהגדרנו משתנה גלובלי בשם `num` ואת הערך שקראנו מהמקלדת אני שומר במשתנה זה.

תרגילים

לכל אחד מהסעיפים כיתבו תוכנית נפרדת, הריצו ובידקו שהתקבלה התוצאה הנדרשת.

1. תוכנית המדפיסה בקשה למשתמש להכניס ערך ב- `setup`. ב- `loop` התוכנית קולטת את הערך, ומדפיסה אותו למסך
2. תוכנית הקולטת שני ערכים `a, b` שהם המקדמים של משוואת קו ישר $(Y=aX+b)$, מבצעת חישוב ערך `Y` עבור ערך `X` שתקבעו כרצונכם ותדפיס למסך את `X` ו- `Y`
3. תוכנית שמקבלת מחיר בדולרים וממירה לש"ח לפי יחס המרה של 3.5 ש"ח לדולר אחד. התוכנית תדפיס למסך משפט שאומר: "המחיר בשקלים הוא:" ואת הערך המחושב
4. תוכנית המקבלת אורכים של שני ניצבים במשולש וחישוב אורך היתר תוך שימוש במשפט פיתגורס ופונקציות מתקדמות (`sqrt` לחישוב שורש ריבועי)
5. תוכנית המקבלת אורכי ניצב ויתר ומחשבת ומדפיסה את הזווית הכלואה ביניהם על ידי שימוש בפונקציה מתקדמת (`acos`), (למשל: אם הניצב 2 והיתר 4 מתקבלת זווית

(60

6. **אתגר :** תוכנית שמקבלת מהירות סיבוב גלגל (RPM) והיקפו וחישוב המהירות הקווית והדפסתה.
7. **אתגר :** תוכנית המקבלת זמן הלוקח לגל קול לנוע אל עצם ובחזרה (ביחידות של שניות), ומחשבת ומדפיסה את המרחק שעבר גל הקול תוך ידיעת ערך מהירות הקול (חפשו את ערך מהירות הקול בגוגל)
8. תוכנית המקבלת ערך מספרי עשרוני parseFloat ומדפיסה למסך
9. חזור על סעיפים 1-7 עם מספרים עשרוניים
10. תוכנית המקבלת ערך כלשהוא בעזרת הפונקציה Serial.read() ומדפיסה אותו
11. **אתגר :** הרחב את התוכנית מסעיף 10 שתכלול המרה מ ascii לערך מספרי של סיפרה יחידה והדפסה למסך
12. חזור על סעיף 11 אבל התוכנית מקבלת מספר דו סיפרתי
13. חזור על סעיף 11 אבל התוכנית מקבלת מספר מספר תלת סיפרתי

קבלת החלטות בתוכנית בעזרת תנאי if

הסבר

החלק המרכזי בתהליך העיבוד שמבצע הבקר הוא קבלת החלטה על בסיס המידע שקלט. למשל: מזגן מודד את הטמפרטורה בחדר וצריך להחליט האם להמשיך לקרר או להפסיק?

קבלת החלטות בתכנות ניתן לביצוע במספר דרכים. הדרך הבסיסית היא בעזרת תנאי if.

התנאי מורכב מפעולה לוגית, שיכולה להיות אמת או שקר. למשל: התנאי בודק האם הטמפרטורה עכשיו נמוכה מ-20 מעלות? אם החיישן מדד טמפרטורה של 25 מעלות אז התנאי לא מתקיים ולכן הוא שקר, אבל אם החיישן מדד 18 מעלות אז התנאי מתקיים והוא אמת.

בתוכנית אנחנו בודקים את התנאי ומגדירים מה צריך לעשות אם התנאי אמת ומה לעשות אם התנאי שקר.

יש 6 סוגי תנאים לוגיים שניתן להשתמש בתוכנית:

== השוואה בין שני ערכים, אמת אם הם שווים, שקר אם הם שונים

!= שוני בין שני ערכים, שקר אם הם שווים, אמת אם הם שונים

$A > B$ אמת אם A יותר גדול מ-B, ושקר אם A קטן או שווה ל-B

$B < A$ אמת אם A יותר קטן מ-B, ושקר אם A גדול או שווה ל-B

$A \geq B$ אמת אם A יותר גדול או שווה מ-B, ושקר אם A קטן מ-B

$A \leq B$ אמת אם A יותר קטן או שווה מ-B, ושקר אם A גדול מ-B

תחביר התנאי

```
if( condition is true)
{
    Statements to run if true;
}
else
{
    Statements to run if false;
}
```

```

1 int a = -13;
2 int b = 152;
3 void setup() {
4     Serial.begin(9600);
5     if(a==b){
6         Serial.println("a is equal to b");
7     }
8     if(a!=b){
9         Serial.println("a is not equal to b");
10    }
11    if(a>b){
12        Serial.println("a is greater than b");
13    }
14    if(a<b){
15        Serial.println("b is greater than a");
16    }
17    if(a>=b){
18        Serial.println("a is greater or equal than b");
19    }
20    if(a<=b){
21        Serial.println("b is greater or equal than a");
22    }
23 }
24
25 void loop() {
26 }

```

לפי הערכים של המשתנים המוגדרים בראש התוכנית, יתקיימו התנאים בשורות 8, 14 ו-20 והתוכנית תדפיס על המסך את המשפטים הבאים :

```

COM3 (Arduino/Genuino Mega or Mega 2560)
a is not equal to b
b is greater than a
b is greater or equal than a

```

חשוב! כאשר תנאי מסויים מתקיים (אמת) אז התוכנית תבצע את כל הפקודות שנמצאות בין הסוגריים המסולסלים שמיד אחרי שורת ה `if`, למשל בתוכנית הדוגמא התנאי בשורה 8 מתקיים והבקר מבצע את כל הפקודות בין הסוגריים המסולסלים בשורה 8 ועד שורה 10. ניתן לכתוב יותר מפקודה אחת לביצוע במקרה שהתנאי מתקיים בין שני הסוגריים המסולסלים.

אם התנאי לא מתקיים אז התוכנית תדלג להמשך התוכנית לאחר הסוגריים המסולסלים שסוגרים את השורות לביצוע, בתוכנית הדוגמא התנאי בשורה 5 לא מתקיים ולכן התוכנית דילגה עד שורה 8.

בדוגמא הבאה רואים תנאי ופקודות לביצוע (שורות 6-8) אם התנאי מתקיים וסט פקודות אחרות לביצוע (שורות 10-12) אם התנאי לא מתקיים:

```
1 int a = -13;
2 int b = 152;
3 void setup() {
4     Serial.begin(9600);
5     if (a == b)
6     {
7         Serial.println("a is equal to b");
8     }
9     else
10    {
11        Serial.println("a is not equal to b");
12    }
13 }
14 }
15
16 void loop() {
17 }
```

תרגילים

לכל אחד מהסעיפים כיתבו תוכנית נפרדת, הריצו ובידקו שהתקבלה התוצאה הנדרשת.

1. תוכנית הבודקת האם שטח ריבוע שאורך צלעו 10 ס"מ גדול או שווה ל-100 סמ"ר. התוכנית תדפיס את המילה GreaterEqual אם התנאי מתקיים, אחרת לא מדפיסה כלום

2. הרחב את התוכנית מסעיף 1 שתדפיס את המילה Less במקרה והתנאי לא מתקיים

3. תוכנית הבודקת האם משולש שאורך צלעות באורכים 52 ס"מ, 74 ס"מ, 126 ס"מ הן צלעות משולש ומדפיסה למסך Triangle אם הוא משולש. אם אינו משולש תדפיס Not Triangle

רמז: סכום שתי צלעות במשולש גדול מהצלע השלישית

4. תוכנית הבודקת האם משולש שאורך צלעותיו 3 ס"מ, 4 ס"מ, 5 ס"מ הוא משולש ישר זווית ומדפיסה למסך Right Angle אם הוא ישר זווית. אם אינו ישר זווית תדפיס Not Right Angle

רמז: יש להשתמש במשפט פיתגורס

5. תוכנית שמגדירה שלושה משתנים int ומאתחלת אותם לערכים שונים. התוכנית תבדוק ותדפיס את המספר הקטן מבין השלושה
6. תוכנית שמגדירה שלושה משתנים int ומאתחלת אותם לערכים שונים. התוכנית תבדוק ותדפיס את המספר הגדול מבין השלושה
7. תוכנית שמגדירה שלושה משתנים int ומאתחלת אותם לערכים שונים. התוכנית תבדוק ותדפיס את שלושת המספרים כשהם מופיעים מהקטן לגדול
8. שנה את התוכנית מסעיף 7 כך שתקלוט 3 מספרים מהמשתמש (מקלדת) ותדפיס את שלושת המספרים כשהם מופיעים מהגדול לקטן
9. **אתגר** : תוכנית שתקלוט 3 מספרים מהמשתמש (מקלדת) המייצגים את שלושת המקדמים של משוואה ריבועית a, b, c . התוכנית תחשב את השורשים של המשוואה ותדפיס תשובה בהתאם. המצבים לבדיקה : שני שורשים, שורש אחד כפול, אף שורש, רמז :

If determinant > 0,	$\text{root1} = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$
	$\text{root2} = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$
If determinant = 0,	$\text{root1} = \text{root2} = \frac{-b}{2a}$
If determinant < 0,	No roots