בית ספר אפי ארזי למדעי המחשב
המרכז הבינתחומי הרצליה

מבוא למדעי המחשב

מבחן אמצע + פתרון

2009

הנחיות:

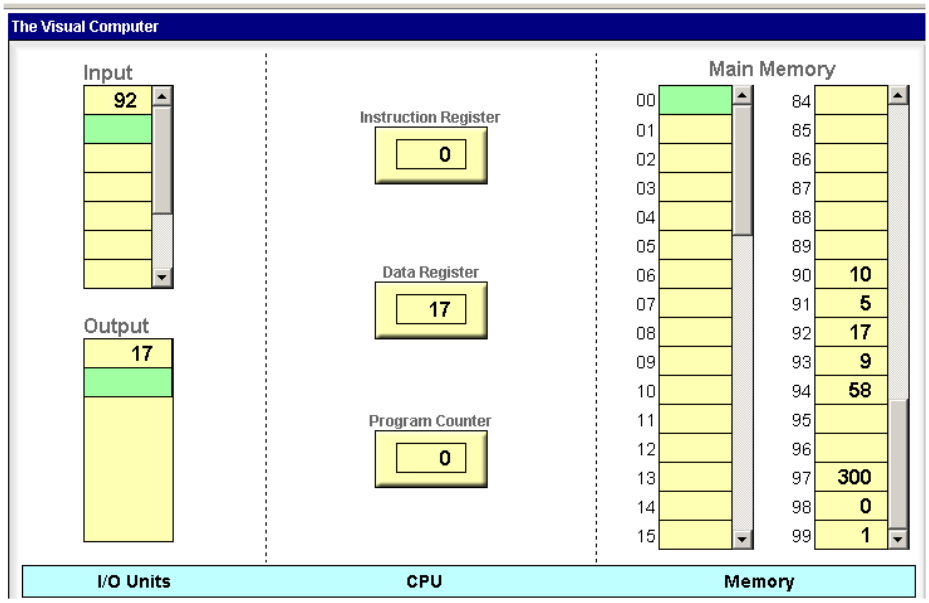
- זמן המבחן: שעתיים. לא תהיה הארכה.
- זמן הבחינה מוגבל, ויש לעבוד ביעילות. אם נתקעים בסעיף מסוים, כדאי לעזוב אותו ולרוץ הלאה.
- חומר סגור. השימוש במחשבים או במחשבונים אסור.
- ענו על כל השאלות על טופס המבחן. מומלץ להשתמש במחברות הבחינה כטיוטא. ניתן גם לכתוב תשובות במחברת הבחינה ולהפנות אליהן בצורה ברורה ומפורשת מטופס המבחן. חובה למחוק באופן ברור כל דבר שאינכם רוצים שנבדוק.
- בגמר הבחינה יש להחזיר את טופס המבחן + מחברת המבחן + כל דפי העזר הנלווים.
- אם תרגישו צורך לעשות הנחה מסוימת כדי לענות על שאלה מסוימת, ניתן לעשות זאת, כל עוד ההנחה היא סבירה ומנוסחת בדיוק ובבהירות.
- אם אתם לא מסוגלים לתת תשובה מלאה לשאלה מסוימת, תנו תשובה חלקית. תשובה נכונה באופן חלקי תקבל ניקוד חלקי.
- התשובות חייבות להיות קצרות ולעניין. כתב היד חייב להיות קריא וברור. תשובות לא קריאות יקבלו ציון 0.
- אם התבקשתם לכתוב תכנית שאמורה לפעול על קלט מסוים, אזי התכנית לא אמורה לבדוק אם הקלט תקין, אלא אם כן נאמר כך בשאלה במפורש.
- התכניות שתכתבו תישפטנה, בין היתר, לפי האורך והאלגנטיות שלהן. תוכניות ארוכות או מסורבלות ללא צורך יקבלו פחות נקודות, אפילו אם הן ממלאות את המשימה שהוגדרה בשאלה.
- כל החומרים להם תזדקקו במהלך המבחן מתועדים בדפי העזר שקיבלת.
- לא יורדו נקודות על טעויות סינטקס טריוויאליות.

בהצלחה!

1. (8 נקודות) כיתבו תכנית בשפת המכונה של Vic שקוראת מספר כלשהו בין 0 ל 99, ומדפיסה את ערך תא הזיכרון שכתובתו הוא המספר הזה. למשל, בהנחה שתמונת המצב של המחשב היא כפי שמופיע למטה, הקלט 92 יגרום לתכנית לכתוב את המספר 17, שהוא תוכן התא שכתובתו 92.

<u>שימו לב:</u>

- התכנית שתכתבו יכולה להניח שלפני שהיא מתחילה לרוץ, תא 97 מכיל את המספר 300.
- הניחו שהתכנית שתכתבו תיטען לזיכרון המחשב מכתובת 00 ואילך.
- <u>התכנית חייבת להיכתב בשפת המכונה של Vic.</u> כלומר – כל פקודה צריכה להיות מספר בן 3 ספרות.



<u>כיתבו את התכנית כאן:</u>

```
800  // read the address, say xx
197  // add 300 to it, creating the instruction 3xx
403  // store this instruction in the next cell
000  // (here you can write any number you want)
900  // write the result
000  // stop
```

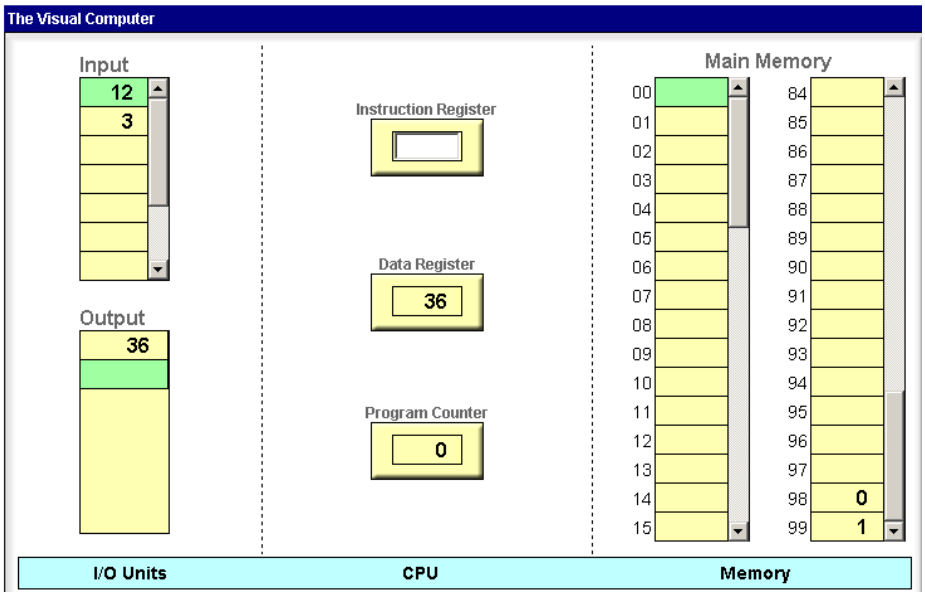
<u>הערות:</u>

תשובות חלקיות שנותנות קוד מוזר אחר או פסבדו-קוד (לאו דווקא עובד) שמבוסס על אותו רעיון תקבלנה ניקוד חלקי.

לא התבקשתם לכתוב הערות (comments) בתכנית זאת, ובכל תכנית אחרת במבחן. אנו כותבים כאן הערות כדי להסביר את הפתרונות.

2. (16 נקודות) כיתבו תכנית Vic שקוראת שני מספרים לא שליליים וכותבת את מכפלתם. למשל, כפי שרואים בתמונת המחשב שמופיעה למטה, הקלט 3,12 יגרום לתכנית לכתוב את המספר 36, שהוא המכפלה של 12 ו 3.

<u>התכנית חייבת להיכתב בשפה הסימבולית של Vic.</u> כל הפקודות והכתובות חייבות להיות סימבוליות.



<u>כיתבו את התכנית כאן:</u>

```
// Multiplies two numbers
  load zero
  store sum    // sum stores the partial product
  read
  store x      // x stores the first multiplication operand
  read
  store n      // n stores the second multiplication operand
NEXT:
  load n
  gotoz END    // if n=0, we are done
  load sum     // sum = sum + x
  add x
  store sum
  load n       // n = n - 1
  sub one
  store n
  goto NEXT
END:
  load sum
  write
  stop
```

<u>הערה כללית על תכניות שאתם מתבקשים לכתוב במבחן:</u>

תיתכנה יותר מתכנית נכונה אחת לפתרון משימה כלשהי. בכל שאלה כזאת אנו נותנים כאן פתרון אפשרי אחד, שהוא בדרך כלל קצר ואלגנטי. אם נתתם פתרון דומה אחר, תקבלו את מלוא הנקודות.

לא יורדו נקודות על טעויות סינטקס וטעויות טריביאליות אחרות.

3. (3 נקודות) האם זמן הריצה של התכנית שכתבת תלוי בסדר שבו נטענים שני המספרים בקלט? למשל, האם התכנית תרוץ באותה מהירות אם הקלט הוא 3,12 לעומת 12,3? הסבר במשפט אחד:

תשובה: כן, זמן הריצה תלוי בסדר הקלט. אחד הקלטים משמש כמונה שקובע את מספר הפעמים שהלולאה תתבצע. אם המספר הזה יהיה הקטן בין השניים, התכנית תרוץ יותר מהר.

4. (3 נקודות) איזה שינויים צריך להכניס לתכנית כדי שזמן הריצה שלה לא יהיה רגיש לסדר בו מופיעים המספרים בקלט? הסבר במשפט אחד:

תשובה: צריך לקרוא את שני המספרים מהקלט, למיין אותם כך שהמספר הקטן בין השניים ישמש כמונה, ורק אז להתחיל את לולאת הכפל.

ענו על אחת מתוך שאלות 5 ו- 6 (30 נקודות):

5. בהינתן מחרוזת (String) כלשהי, אנו מגדירים את המילה שמתחילה במקום ה- n במחרוזת כרצף האותיות שמתחיל במקום ה- n עד שנתקלים ברווח, או עד שמגיעים לסוף הטקסט. למשל, נניח שהטקסט הוא "a tale of two cities". במקרה זה, המילה שמתחילה ב 0 היא "a". המילה שמתחילה ב 7 היא "of". המילה שמתחילה ב 10 היא "two". וכן הלאה. שימו לב: בהחלט ייתכן שבין שתי מילים מפריד יותר מרווח (space) אחד.

5-א. (10 נקודות) כיתבו מתודה בשם word שמקבלת שני פרמטרים: פרמטר s מטיפוס String ופרמטר n מטיפוס int. המתודה מחזירה ערך String שמכיל את המילה ב s שמתחילה במקום n.
למשל, אם s מייצג את הטקסט "a tale of two cities", אזי אם n=0, המתודה תחזיר "a". אם n=7, המתודה תחזיר "of". וכולי.

כיתבו את המתודה כאן:

```
// Returns the next word in s starting from location n
private static String word(String s, int n) {
    int nextSpaceLocation = s.indexOf(' ',n);
    if (nextSpaceLocation == -1) {
      return s.substring(n,(s.length()));
    } else
    return s.substring(n,nextSpaceLocation);
  }
}
```

הערה: אפשר היה להניח ש n נמצא בתחילת מילה, כלומר לא ברווח. יחד עם זאת, מתודה שמניחה שיכולים להיות רווחים ומנקה אותם תזכה במלא הנקודות גם כן.

5-ב. (20 נקודות) כיתבו מתודת main שפועלת על משתנה String ששמו text כדלקמן. המתודה מדפיסה כל מילה בטקסט בשורה נפרדת. לאחר מכן, המתודה מדפיסה את המילה הארוכה ביותר בטקסט.

לדוגמא, בהינתן הטקסט "Tom played in the park" המתודה תדפיס את הפלט הבא:

```
Tom
played
in
the
park
played
```

מתודת main שתכתבו צריכה להשתמש בשירותים של מתודת word. שימו לב: גם אם לא הצלחתם לכתוב את המתודה word, אתם עדיין יכולים להשתמש בה כאילו היא קיימת ופועלת כהלכה.

כיתבו את מתודת main כאן (כתבנו עבורכם את השורות הראשונות של המחלקה והמתודה):

```
import java.util.Scanner;

public class PrintWords {

  public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    System.out.print("Enter text:");
    String text = scan.nextLine();
    // write your code here:

    // Leading spaces can be trimmed from the text, but we didn't ask it.
    int k = 0;
    int max = 0;
    int maxIndex = 0;
    // Breaks the text into words, prints each word in a separate line
    while (k < text.length()) {
      String w = word(text, k);
      System.out.println(w);
      // Check if this word has maximal length (so far)  // 5 points
      if (w.length() > max) {
        max = w.length();
        maxIndex = k;
      }
      // Advance k to the next word, skipping spaces
      k = k + w.length();
      if (k<text.length())
        while (text.charAt(k) == ' ') k++;
    }
    System.out.println(word(text, maxIndex));  // Prints the maximal word
  }

  // The code of the word method is repeated here, just for reference.
  private static String word(String s, int n) {
    int nextSpaceLocation = s.indexOf(' ',n);
    if (nextSpaceLocation == -1) {
      return s.substring(n,(s.length()));
    } else
    return s.substring(n,nextSpaceLocation);
  }
}
```

6. "מספר מושלם" (perfect number) הוא מספר שסכום המחלקים שלו, חוץ מעצמו, שווה למספר.
למשל, 6 הוא מספר מושלם כי 6=1+2+3 .

6-א. (20 נקודות): כיתבו מתודה perfectNumberPrint(x) שמקבלת מספר int גדול מאפס כפרמטר. אם המספר מושלם – למשל 6 - המתודה כותבת את השורה הבאה:

6 = 1 + 2 + 3

אם המספר אינו מושלם, המתודה מסיימת בלי לעשות כלום.

כיתבו את המתודה כאן:

```
// If x is a perfect number, prints x and its divisors
public static void perfectNumber(int x) {
  int sum = 0;
  String s = x + " = 1";
  for (int j = 1; j < x; j++)
    if (x % j == 0) {
      sum = sum + j;
      // Builds the output string on the fly  // 5 points
      if (j>1) s = s + " + " + j;
    }
  // If x is perfect, prints the output string
  if (sum == x)
    System.out.println(s);
}
```

6-ב. (10 נקודות): כיתבו מתודת main כדלקמן: המתודה קולטת מספר חיובי n מהמשתמש ומדפיסה את כל המספרים המושלמים מ 1 עד n. למשל, אם המשתמש יספק את המספר 1000, התכנית תכתוב:

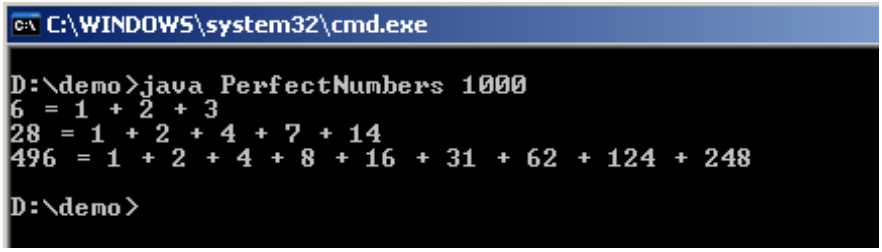
6 = 1 + 2 + 3
28 = 1 + 2 + 4 + 7 + 14
496 = 1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248

למה? כי אלה שלושת המספרים המושלמים היחידים בין 0 ל 1000.

שימו לב: גם אם לא הצלחתם לממש את המתודה perfectNumberPrint, אתם עדיין יכולים להשתמש בה כאילו היא קיימת ופועלת כהלכה.

כיתבו את מתודת main כאן (כתבנו עבורכם את השורות הראשונות של המחלקה והמתודה).

```
import java.util.Scanner;

public class PerfectNumbers {

  public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    System.out.print("Enter a number:");
    int n = scan.nextInt();
    // write your code here:

-------------------------------------------------------------------------

// To make the solution more interesting, we demonstrate the use of a
// command line argument. This way, to invoke the program, say, on 1000,
// one writes "java PerfectNumbers 1000". See the screen shot below.
// Needless to say, this was not required in the exam.

public class PerfectNumbers {

  // Prints all the perfect numbers up to args[0]
  public static void main(String[] args) {
    // (int) args[0] won't work since args[0] is a String,
    // so we need to use a more powerful type convertor.
    int n = Integer.parseInt(args[0]);
    for (int i = 1; i < n; i++) {
      perfectNumber(i);
    }
  }

  // perfectNumber method comes here (see answer to 6-alef).
}
```



עיינו במחלקה Clock שמופיעה באחד מדפי העזר שמלווים את המבחן. הניחו שבמחלקה Clock אין אף מתודה אחרת פרט לאלה שמופיעות בדף העזר.

<u>ענו על 8 מתוך שאלות 7-16</u> (4 נקודות כל שאלה)

7. למחלקה Clock אין מתודת main. האם זאת טעות? הסבר במשפט אחד.

<u>תשובה:</u> לא, זאת אינה טעות. בהחלט יכולה להיות מחלקה ללא מתודת main. מתודת main צריכה להופיע באחת מהמחלקות שיוצרות את ה application.

שים לב שהשדות hours ,minutes, seconds מוגדרים private.

8. מהי משמעות ההגדרה הזאת?

<u>תשובה:</u> המשמעות היא שהם לא נגישים מחוץ למחלקה.

9. למה כדאי להגדיר אותם כך?

<u>תשובה:</u> כדי להגן על האובייקט ולא לאפשר ל clients לקרוא או לשנות את השדות שלו ללא פיקוח של המחלקה.

10. איך אופן ההגדרה הזה מתקשר למושג encapsulation?

<u>תשובה:</u> ההגדרה הזאת מיישמת את עיקרון ה encaspulation. היא גורמת לכך שהאובייקט מוגן (encapsulated) באופן שלא ניתן לגשת אליו ישירות מחוץ למחלקה.

התבונן בקטע הקוד הבא:

```
Clock c1 = new Clock(23,59,58);
c1.advanceSecond();
Clock c2 = c1;
c2.advanceSecond();
System.out.println(c1);  // q5
for (int i = 0; i < 614; i++) c1.advanceSecond();
System.out.println(c2);  // q6
c2 = new Clock(c1.getHours(), 15, c1.getSeconds());
System.out.println(c2);  // q7
```

11. איזה פלט תדפיס הפקודה שמופיעה לידה ההערה q5?

<u>תשובה:</u> 00:00:00

12. איזה פלט תדפיס הפקודה שמופיעה לידה ההערה q6?

<u>תשובה:</u> 00:10:14

13. איזה פלט תדפיס הפקודה שמופיעה לידה ההערה q7?

<u>תשובה:</u> 00:15:14

14. קומפיילר Java מייצר שפת מכונה. נכון / לא נכון? הסבירו במשפט אחד.

<u>תשובה:</u> לא נכון. קומפיילר Java מייצר קוד בשפת ביניים, שנקראת Bytecode.

15. הערך של משתנה static לא ניתן לשינוי ע"י מתודה מסוג private. נכון / לא נכון? הסבירו במשפט אחד.

תשובה: לא נכון. משתנה static נגיש לכל המתודות במחלקה, כולל אלה שהוגדרו private.

16. מתודה מסוג void לא יכולה לקרוא למתודה מסוג int. נכון / לא נכון? הסבירו במשפט אחד.

תשובה: לא נכון. אין שום מניעה למתודה מסוג void לקרוא למתודה מסוג int.

ענו על אחת משתי השאלות 17-18

17. (10 נקודות) אנו מעוניינים להוסיף למחלקה Clock מתודה חדשה שמקדמת את השעון n שעות, כאשר n הוא פרמטר מסוג int שמכיל מספר שלם לא שלילי. המספר מועבר למתודה ע"י קוד שנמצא מחוץ למחלקה. כיתבו את המתודה הזאת. הניחו שבמחלקה Clock אין אף מתודה אחרת פרט לאלה שמופיעות בדף העזר.

```
// Advance the clock by n hours
public void hoursElapsed(int n) {
    hours = (hours + n) % 24;
}
```

18. (10 נקודות) אנו מעוניינים להוסיף למחלקה Clock את היכולת להציג בכל רגע נתון כמה שעונים יוצרו עד כה ע"י המחלקה. ציינו בדיוק נמרץ איזה קוד צריך להוסיף למחלקה כדי לממש את הדרישה הזאת. הניחו שבמחלקה Clock אין אף מתודה אחרת פרט לאלה שמופיעות בדף העזר.

תשובה:

צריך להוסיף משתנה סטטי ששמו, נניח numberOfClocks לתחילת ה class, לפני שמתחילים להגדיר את המתודות, ולאפס אותו.

צריך להוסיף את השורה ++numberOfClocks לקונסטרקטור.

צריך להוסיף את המתודה הבאה למחלקה:

```
public int getNumberOfClocks() {
    return numberOfClocks;
}
```

הערה: ניתן לתת תשובה יותר מפורטת, אבל בחרנו לפרט כאן את התשובה המינימלית הנכונה.

מספר הנקודות במבחן הוא 102. הציון הסופי ינורמל לבסיס 100 ויעוגל למעלה. כלומר, אם סך הנקודות שקיבלת במבחן הוא x, הציון הסופי יהיה `Math.ceil(x*100.0/102.0)`

VIC Documentation

In the following documentation, the word "register" refers to Vic's data register, also known as D.

Machine Code	Assembly	Operation	Description
1xx	ADD	D= D + M[xx]	Add the value of memory[xx] value to the register
2xx	SUB	D =D - M[xx]	Subtract the value of memory[xx] from the register
3xx	LOAD	D = M[xx]	Load the value of memory[xx] to the register
4xx	STORE	M[xx] = D	Store the register's value in memory[xx]
5xx	GOTO	Execute M[xx]	Go to execute the instruction stored in memory[xx]
6xx	GOTOZ	IF D = 0 execute M[xx]	If the register's value equals zero, Go to execute the instruction stored in memory[xx]
7xx	GOTOP	IF D > 0 execute M[xx]	If the register's value is greater than zero, Go to execute the instruction stored in memory[xx]
800	READ	D = input	Read the next input value into the register
900	WRITE	output = D	Write the register's value to the output
000	STOP	STOP	Stop the program's execution

Clock Class

```
/**
* represents a clock that keeps time in the format
* hours:minutes:seconds where 0<=hours<24, 0<=minutes<60,
* <=seconds<60.
*/

public class Clock {

  // Current time represented by the clock
  private int hours, minutes, seconds;

  // Creates a Clock instance and sets the time to given values
  public Clock(int hours, int minutes, int seconds) {
      this.hours = hours;
      this.minutes = minutes;
      this.seconds = seconds;
  }

  public void advanceSecond() {
      if (seconds < 59) {
          seconds++;
          return;
       } else
            seconds = 0;
      if (minutes < 59)
           minutes++;
       else {
           minutes = 0;
            hours = hours < 23 ? hours+1 : 0;
      }
  }

  // Accessors
  public int getHours() {
    return hours;
  }

  public int getMinutes() {
    return minutes;
  }

  public int getSeconds() {
    return seconds;
  }

  // A textual representation of the Clock object
  public String toString() {
    return (hours + "\t" + minutes + "\t" + seconds);
  }
}
```

String API

שימו לב: אנו מפרטים כאן חלק מה- API של מחלקת String. כדי לענות על שאלות המבחן ניתן להשתמש רק בחלק מהמתודות המתועדות כאן.

Method Summary	
char	**charAt**(int index) Returns the char value at the specified index.
boolean	**endsWith**(String suffix) Tests if this string ends with the specified suffix.
boolean	**equals**(String aString) Compares this string to the other specified string.
int	**indexOf**(int ch) Returns the index within this string of the first occurrence of the specified character.
int	**indexOf**(int ch, int fromIndex) Returns the index within this string of the first occurrence of the specified character, starting the search at the specified index.
int	**indexOf**(String str) Returns the index within this string of the first occurrence of the specified substring.
boolean	**isEmpty**() Returns true if, and only if, length() is 0.
int	**length**() Returns the length of this string.
String	**substring**(int beginIndex) Returns a new string that is a substring of this string.
String	**substring**(int beginIndex, int endIndex) Returns a new string that is a substring of this string.
String	**trim**() Returns a copy of the string, with leading and trailing whitespace omitted.