# פעולות אריתמטיות לוגיות

**פעולות לוגיות**

פתחו את הקובץ base.asm **ושימרו אותו בשם אחר.**

## קלט ופלט

בתרגיל הבא מספר שאלות שעבורן נצטרך לקלוט ולהדפיס מספרים. לפני שנתחיל לעבוד עם הפקודת הלוגיות צרו את הפרוצדורות הבאות:

* readUserInput
* printString
* printNumber
* newline

|  |  |
| --- | --- |
| ; new line  proc newLine  pusha  mov dl, 0ah  mov ah, 2h  int 21h  popa  ret  endp newLine | ;proc to get input to al  proc readUserInput  ; get user first input number to al  mov ah, 1h  int 21h  sub al, '0'  ret  endp readUserInput |
| ;proc to print the a number stored in al  proc printNum  pusha  mov ah, 0  div [ten]  add ax, '00'  mov dx, ax  mov ah, 2h  int 21h  mov dl, dh  int 21h  popa  ret  endp printNum | ; print message  proc printString  pusha  mov ah, 9h  int 21h  popa  ret  endp printString |

## פקודת and

בעזרת פקודת and נבדוק האם מספר הוא זוגי או אי זוגי

כדי לדעת אם מספר זוגי עלינו לבדקו את הביט הימני ביותר שלו.

מסכה של and:

1 and 1 ⇨ 1

1 and 0 ⇨ 0

0 and 0 ⇨ 0

**הדרכה:**

1. הגדירו שלושה משתנים של מחרוזות

msg db 'Please enter a number: $'

הגדיר עוד שתי 2 מחרוזות odd ו even.   
אם המספר זוגי נדפיס למסך even ואם הוא אי זוגי נדפיס למסך odd

הדפיסו את המחרוזת של בקשה להקליד למסך תו אחד בעזרת הפרוצדורה **printString**  
אל תשכחו להעביר את תחילת המחרוזת לרגיסטר dx ⇦ mov dx, offset msg

1. בעזרת הפרוצדורה readUserInput קלטו מספר כלשהו לרגיסטר al.

לדוגמא:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

al =5 ⇨

1. חשבו מהי המסכה המתאימה ביותר לבדוק האם המספר מתחלק ב – 2.   
   הפעילו את ההוראה and על רגיסטר al ובדקו את התוצאה.  
   אם המספר שווה ל – 0 קפצו לתווית והדפיסו את המחרוזת even

אם המספר אינו שווה לאפס קפצו תווית אחרת והדפיס ולמסך את המחרוזת את odd

1. כתבו תכנית חדשה הבודקת האם מספר (8 ביט) מתחלק ב- 4?

כתבו תכנית הקולטת מספר,   
ובודקת אם הוא מתחלק ב – 4?

**מהי המסכה שצריך לצורך הבדיקה?**

## פקודת or

הגדירו משתנה בשם my\_bits בגודל בית וקראו מספר לתוכו.

"הדליקו" את ערכו של הביט השלישי (משמאל) במשתנה, כלומר ודאו שערכו 1.  
בדקו את התוצאה ב tùrbo debugger

## פקודת xor

1. קראו ערך כלשהו ל al ואפסו אותו באמצעות פעולת ביטים. הדפיסו את התוצאה.
2. אפסו את רגיסטר bx בעזרת פקודת xor ובעזרת פקודת השמה של 0 לרגיסטר, ובדקו איזו פקודה תופסת פחות מקום במקטע הקוד.

## שימוש בפקודות הזזה (shift) לכפל וחילוק

* הכניסו ל-al את הערך 3. בעזרת פקודות הזזה, כיפלו את al ב-4. הדפיסו את התוצאה
* הכניסו ל-al את הערך 136d. בעזרת פקודות הזזה, חלקו את al ב-8. הדפיסו את התוצאה
* הכניסו ל-al את הערך 4 (דצימלי). בעזרת פקודות הזזה וחיבור, כיפלו את al ב-20. הדפיסו את התוצאה

**הדרכה:** התייחסו ל-20 בתור סכום של 16 ו-4. השתמשו ברגיסטרים נוספים כדי לשמור חישובי ביניים.

