תוכנית דוגמא add1.asm

תוכנית הדוגמא add1.asm הינה תוכנית הקוראת מספר שלם אי שלילי מהמקלדת, מקדמת אותו באחד ומדפיסה את המספר לאחר הקידום באחד על המסך. תפקידה העיקרי של התוכנית הזו להמחיש שקלט המתקבל מהמקלדת הוא תמיד טקסטואלי ואם רוצים לבצע עליו חישובים ולהציג את התוצאה יש לבצע התמרות, התוכנית גם מראה איך באופן עקרוני הדבר נעשה.

התוכנית עושה לעצמה מספר הנחות: היא מניחה שהמספר הוא בן ארבע ספרות עשרוניות בדיוק ומטילה על המשתמש להקדים אפסים במקרה של מספר קטן יותר. לדוגמא, אם המשתמש רוצה לדעת את ערך העוקב של 569 הוא צריך להקליד 0569 ועל המסך יופיע 0570. בנוסף, התוכנית אינה בודקת תקינות קלט. היא אינה מודאת שכל התוים שהיא מקבלת הם אכן ספרות.

התוים נקראים ע"י פסיקת התוכנה INT 21h אופציה AH=1 ארבע פעמים. הפסיקה הזו חוזרת מיד עם לחיצת כל מקש (עם קוד האסקיי של המקש ב-AL) ולפיכך אין צורך בלחיצת Enter על מנת לקבל את התשובה. אשר לפלט, התוכנית מכינה את ResultStr על מנת שתוכל לשלוח אותו למסך בבת אחת ע"י INT 21h אופציה AH=9. למעשה התוכנית תכתוב לכל המשבצות במחרוזת שמאותחלות ב-X.

המשתנה Var1 משמש לשימור סכומי הביניים משום ש-AX חייב לשמש אותנו לקריאת המקש מהמקלדת. מאחר ואין אפשרות להוסיף רק את AL ל-Var1, אנחנו מוסיפים לו את AX לא לפני שמאפסים את AH.

התוכנית ממחישה, בין השאר, את הקשיים של לכתוב תוכנית כזו באסמבלי. אפילו בעזרת INT 21h אנחנו יכולים לקרוא רק טקסט מהמקלדת, ולשלוח רק טקסט למסך. אם אנחנו צריכים את נתון הקלט בצורה בינארית אנחנו צריכים להמיר אותו. אם אנחנו רוצים לשלוח נתון שחושב באופן בינארי אנחנו צרכים להמיר אותו חזרה.
ההמרה מטקסט לבינארי נעשה ע"י כך שכל סיפרה שמקבלת מומרת לבינארי ע"י הפחתה של קוד האסקיי של '0' (הערך הוא 48 למי שסקרן). מעבר לכך, על כל סיפרה שנקראית (למעט הראשון) מכפילים את המספר הבינאי שאנחנו יוצרים ב-10 ורק לאחר מכן מוסיפים את הערך הבינארי של הסיפרה החדשה. לדוגמא אם הקלט הוא "275" אנחנו

נחשב את הערכים הבינאריים 275 ,27 ,2.

ההמרה <u>מבינארי לטקסט</u> נעשה ע"י חלוקות חוזרות של המספר הבינארי ב-10 עד שהוא מתאפס כאשר לשארית החלוקה מוסיפים את הקוד אסקי של '0' וכך מתקבלים הספרות בסדר עולה של משמעות. לדוגמא אם הערך הבינארי הוא 275 אנחנו נחשב את הטקסט בשלבים "275" ,"75" ,"5".

כאשר אנחנו מקבלים ספרות מהמקלדת אנחנו מקבלים אותם בסדר יורד של משמעות וכאשר אנחנו מתמירים את הערך המקודם באחד ל-Ascii ע"י חלוקות חוזרות של 10 אנחנו מקבלים את הספרות בסדר עולה של משמעות, ובגלל זה אנחנו ממלאים את שני החלקים של ResultStr בסדרים הפוכים: משמאל לימין עבור המספר הנקלט ומימין לשמאל לגבי מספר התוצאה.

מאחר ולא ניתן לבצע את הפקודות MUL ו-DIV עם אופרנד קבוע, התוכנית מגדירה משתנה Ten מוגדר כ-word ע"י המאפיין DW המאותחל ב-10. התוכנית אינה משנה את ערך המשתנה הזה, כמובן. את אפקט ההכפלה וחילוק ב-10 מקבלים ע"י הפקודות

```
MUL Ten
```

ו-

```
DIV Ten
```

בהתאמה.

```
; add1.asm
;
     .MODEL SMALL
     .STACK 100h
     .DATA
Var1        DW 0
PromptStr DB 'Enter 4 digit number:',13,10,'$'
ResultStr DB 13,10,'XXXX + 1: XXXX',13,10,'$'
Ten         DW 10
                              ;
      .CODE
      MOV AX,@DATA    ; DS can be written to only through a register
      MOV DS,AX       ; Set DS to point to data segment
      MOV AH,9        ; Set print option for int 21h
      MOV DX,OFFSET PromptStr  ;  Set  DS:DX to point to PromptString
      INT 21h                  ;  Print DisplayString
;
      MOV Var1,0
;           Read first digit
      MOV AH,1
      INT 21h
      MOV ResultStr[2],AL
      SUB AL,'0'
      MOV AH,0
      MUL Ten
      MOV Var1,AX
      MOV AX,0
;           Read second digit
      MOV AH,1
      INT 21h
      MOV ResultStr[3],AL
      SUB AL,'0'
      MOV AH,0
      ADD AX,Var1
      MUL Ten
      MOV Var1,AX
      MOV AX,0

;            Read third digit
      MOV AH,1
      INT 21h
      MOV ResultStr[4],AL
      SUB AL,'0'
      MOV AH,0
      ADD AX,Var1
      MUL Ten
      MOV Var1,AX
      MOV AX,0
;           Read forth digit
      MOV AH,1
      INT 21h
      MOV ResultStr[5],AL
      SUB AL,'0'
      MOV AH,0
      ADD AX,Var1
```

```
;
;   Increment by one

      INC AX
;
;    Convert back to ascii
;
      MOV DX,0
      DIV Ten
      ADD DL,'0'
      MOV ResultStr[15],DL
;
      MOV DX,0
      DIV Ten
      ADD DL,'0'
      MOV ResultStr[14],DL
;
      MOV DX,0
      DIV Ten
      ADD DL,'0'
      MOV ResultStr[13],DL
;
      MOV DX,0
      DIV Ten
      ADD DL,'0'
      MOV ResultStr[12],DL
;
      MOV AH,9
      MOV DX,OFFSET ResultStr
      INT 21h
;
      MOV AH,4Ch        ; Set terminate option for int 21h
      INT 21h         ; Return to DOS (terminate program)
      END
```

```
E:\>add1.exe
Enter 4 digit number:
0099
0099 + 1: 0100

E:\>
```