

COBOL
Common Business Oriented Language

ההשפה בעלת אופי מסחרי, ותוכננה לשימושים עסקיים. החוקים של המחשב בשפה זו עושים אותה שימושית לבעלי מסחריות: היא אינה מותאמת במיוחד לשימושים בעלי אופי מדעי.

.DIVISIONS חטיבות ל-4 מחולקת קובל תוכנת

חטיבת הדמיון - IDENTIFICATION חטיבת הסביבה - ENVIRONMENT חטיבת הנתונים - DATA
חטיבת הבצע - PROCEDURE

לדוגמה, חטיבת הנתונים, המתארת את כל שטחי האחסון בהם משתמש התוכנית, מתחילה לשני קטעים:

כל קטע ניתן לחלקה לPARAGRAPHS.

כל יתר המשפטים, הכתובים בתכנית COBOL נחשבים STATEMENTS.

* בזיר-הרים כ- 5 קילומטרים צפונית לזרם נסען דרומה.

מחרוזות A-טוריים | סט 2112 | סט 2182

16 - B RUACH

המלה AUTHOR שם המחבר), שהוא שם סעיף בתכנית, חיבת להתחילה במרוחך A. לאחריה ניתן לשים כל מידע שהוא במרקוזות B.

ניצן לשים את האות A של המלה AUTHOR בטור 8 או במרוחה A. המלה עצמה, עם זאת, ממשיכת אל תוך התוכום של מרוחה . המידע המוכנס לאחר המלה, AUTHOR חייב להתחיל במרוחה B, או אחרי טור 11. בדוגמה שלנו,

חוגי מרכז שלדים:

IDENTIFICATION DIVISION

ENTRANCE EXAMINATION DIVISION

የኢትዮ ሥር : AUTHOR

- INSTALLATION

- DATE-WRITTEN

- DATE-COMPILED

SECURITY - ביטחון.

. הערות - REMARKS

ENVIRONMENT DIVISION

- CONFIGURATION SECTION .1

- CONFIGURATION SECTION 1 INPUT OUTPUT SECTION 2

- INPUT-OUTPUT SECTION .2

SELECT גזירה **AS** (פירוש היחס) **IS** (שם קבוע) **ORGANIZATION**

DATA DIVISION

הטיית הנתונים בתכנית COBOL הינה החטיבה, המתארת ומגדירה את שדות הנתונים בשטח האחסון. יש להגדיר כל שטח אחסון, הדריש לעיבוד הנתונים כל תכנית, הקרויה נתונים נתוני פלט, זקופה לקטע קבצים, כדי לתאר את הקלט והפלט. קטע שטח-האחסון לעובדה הוא הקטע, בו מגדירים שדות, שאינם חלק ממשתי אחסון הקלט או הפלט.

FILE SECTION .1

קטע הקבצים הוא הקטע המתאים את כל קובצי הקלט והפלט בהם משתמשת התוכנית.
קובצים אלה כבר הוגדרו בחטיבת הסביבה, בפקודות SELECT, המקצתות שם לכל קובץ אשר הוגדר, ומשיקות אליו ייחידת גלגול-פלט.

FILE SECTION .1
WORKING-STORAGE SECTION .2
REPORT SECTION .3

-FD- תיאור קובץ

SELECT הקבצים מתאר כל הוראת תיאור הקובץ FD . כל FD יתאר קובץ, שהוגדר באמצעות הוראת בחתיבת הסביבה. לדוגמה FD file1

- L 1. כלים לכתיבה של טקסטים תיאורי הרשותה

A 2. יש לרשום את FD במרקוז A

B 3. שאר הרישומים ירשמו במרקוז B

E 4. נקודה נרשמת ז' פלאז גאנטונג וואויאן סל א' FD

L 5. מותר לרשום פסיק בין משפט אחד לשני .

G 6. אם נרשם פסיק הוא חייב להיות לפני מרוז אחד לפחות .

J 7. כל משפט יכול להרשם בשורה נפרדת או לחילופין כרכץ בשורה אחת, אולם ברור כי קריית הנתונים בהירה יותר אם כל משפט נכתב בשורה נפרדת

LABEL RECORDS ARE OMITTED STANDARD

LABEL RECORDS ARE STANDARD

1. הרשומה הראשונה בקובץ אינה רשומה נתונים, אלא **תוויות רשומה התחלתית תקנית**, בת 80 תוויות. באופן דומה, הרשומה האחורה הינה **תוויות רשומה סופית**. (מיועדים רק לקבצי סרט מגנטוי, וגישה ישירה).
 2. בקובצי קלט - על המחשב לבדוק **תוויות אלה**, ובקובצי פלט ' על המחשב **לייצרן**.
עבור קבצי הדפסה יש **לכתוב את המשפט**:

מופיע RECORD CONTAINS רשות.

(אורך הרשומה) מצביעים על אורכה של כל רשותה.
לדוגמא:

RECORD CONTAINS (15) CHARACTERS

מופיע BLOCK CONTAINS הגוש מכיל / רשות

BLOCK CONTAINS RECORDS (קבוע מספרי) RECORDS

המשפט DATA RECORD(S) רשות הנתונים/ רשות (ויליאם ססן ג'ון)

DATA RECORD(S) {IS ARE} שם רשומה {שמות רשומות}).

שם	דוח חובב/רשות	פריט המשפט	משפט
הנפקה צפוי דין לאירועי גנובות	אזרע	{STANDARD OMITTED }	LABEL RECORD ARE
הנפקה צפוי לאירועי גנובות	סמלר	CHARACTERS (קבוע מספרי)	RECORD CONTAINS
הנפקה צפוי לאירועי גנובות	סמלר	RECORDS (קבוע מספרי)	BLOCK CONTAINS
הנפקה צפוי לאירועי גנובות	ולאייר	ARE (שם רשומה) IS (שםות רשומות)	DATA (RECORD(S

VALU OF FILE-ID IS "FILE.DAT"

תרגיל

כתב FD עבור קבצי הרשומות המצורפים.

הגדרת רשומות וshedot

PICTURE משפט

- **מצין אלפבית** - A (האות הראשונה) ב (האות השנייה) C (האות השלישי)
 - **מצין אלףונומרי** - X (המספר הראשון) Y (המספר השני) Z (המספר השלישי)
 - **מצין נומרי (ספרתי)** - 1 (המספר הראשון) 2 (המספר השני) 3 (המספר השלישי)

02 AMT PICTURE IS 99999.~ 9(5) 11

02 ITEM1 PICTURE IS AAAAA.

01 REC-1.

02 ITEM1 PICTURE AAAAA

02 AMT PICTURE 99999

02 CDE PICTURE XX

04 NAME PICTURE A(10).

04 NAME PIC A(10).

PIC A (25)

בדי לאיין שדרה, שאיין בו גמינו בעל משמעות. בימנו ב-COBOL להשתמש במלגה השמורה FILTER.

מיוחד, שדה - FILLER, בעל משפט PICTURE מותאים, מצוי שטח בראשומה, אשר אינו משמש בו כרגע. לא ניתן לגשת

אל שדה FILLER

01 TIME-CARD.

02 EMPLOYEE-NAME PIC A(25).

02 FILLER PIC X(5).

02 HOURS-WORKED PIC 9(5).

02 FILLER PIC X(45).

02 AMT PIC 999V99

תרגיל

תארו את מבנה רשומות הקובץ לרשותות המצורפות.

לפניהם. * בזאת כוונתנו, נא-01 יי' נס-01 רוז הילברג, גזענאל גיטל וו, אפריל 1944 מ-
רלען, (-) פירט וווען וויאן זא-01 נס-01 רוז הילברג, גזענאל גיטל וו, אפריל 1944 מ-

לכל קובץ, המצוין במשפט DATA RECORDS, יש לשינר רמה 01, ולתאר את הרמות הרכפפות לו. יש לתאר את כל הרשומות בקובץ (בקטע FD) מסוים, לפני שעוברים לתיאור הקובץ (ה-FD) הבא.

קטע האחסון לעובודה WORKING-STORAGE SECTION

כל שדה הדרש ליעבוד, שאינו חלק מהקלט או מהפלט, ניתן להגדירה בקטע שטח האחסון לעובודה...למשל, אם יש לתאר לשדה ערך התחלתי 0, ולהגדיל את ערכו ב-1 רק כאשר תקרא הרשומה האחונה, ניתן להגדיר שדה זה ב-WORKING-STORAGE SECTION. גם שטחים, הדרושים לאחסון תוכנות ביןיהם של עיבוד הנתונים, יוגדרו בקטע זה.

כללים לשימוש בקטע שטח האחסון לעובודה

1. קטע שטח האחסון לעובודה יבוא לאחר כל המשפטים, המופיעים בקטע הקבצים.
2. יש לקודד WORKING-STORAGE SECTION בשורה נפרדת, במרוח A, ולסימן בנקודה.
3. הגדרה ראשונית של פריט מקובץ, אשר יתחלק לשטחי-אחסון נוספים. לדוגמה:

```
.WORKING-STORAGE SECTION
. 01 STORED-AREA
```

4. הגדרת כל השדות הדרושים בתוך הגדרת 01 לעיל :

```
.WORKING-STORAGE SECTION
. 01 STORED-AREA
. 02 END-OF-FILE-SWITCH
. 02 GROSS-AMT
```

5. יש לתת שמות לפריטים המקבצים והבסיסיים, לפי כללי מתן-שמות.

6. לכל פריט בסיסי בלתי תלוי - יש לכתוב משפט : PICTURE

```
.WORKING-STORAGE SECTION
. 01 STORED-AREA
. 02 END-OF-FILE-SWITCH PIC 9
. 02 GROSS-AMT PIC 999V99
```

7. כל פריט בסיסי בלתי תלוי יכול לקבל ערך תחלתי, אם דרוש צהה :

```
.WORKING-STORAGE SECTION
. 01 STORED-AREA
. 02 EOF-SWITCH PIC 9 VALUE ZERO
. 02 GROSS-AMT PIC 999V99 VALUE ZERO
```

אם אמ רצים להשתמש בשדה EOF-SWITCH, לסמן מתי בוצעה קריית הרשומה الأخيرة, אנו יכולים להשתמש בו בחטיבת הביצוע :

READ FILE AT END MOVE 1 TO EOF-SWITCH.

כל המשפטים בקטע זה יכולים להכיל הצהרות - VALUE, אך אלה אינן הצהרות חובה, אלא רק על פי צרכי של המתכונת.

דוגמאות:

WORKING-STORAGE SECTION.

01 WORK-AREAS.

02 TOTAL PIC 9(5) VALUE ZERO.

02 CONST PIC X(4) VALUE 'CODE'.

02 SAVE-AREA PIC X(3) VALUE SPACES.

01 ADDRESS-1.

02 NUMB PIC 9(4) .

```

02 STREET PIC X(20) .
02 CITY PIC X(25) .
02 STATE PIC X(3) .
01 DATEX.
 02 MONTH PIC 9(2) VALUE 06.
02 YEAR PIC 9(4) VALUE 2003.

```

ג. שימוש בפריטים ברמה-77 בשטח העבודה .

פריט-רמה 77 בקטע שטחי העבודה ישמש להגדרת שדה עצמאי. הוא יכול להכיל משפט-VALUE, ויש לו קודדו במרקווין A.

שיטה 1

```

WORKING-STORAGE SECTION.
01 WORK-AREAS VALUE ZERO.
 02 EOF PIC 9.
 02 TOTAL PIC 999.
 02 CTR PIC 99.

```

שיטה 2

```

WORKING-STORAGE SECTION.
 77 EOF PIC 9 VALUE 0.
 77 TOTAL PIC 999 VALUE 0.
 77 CTR PIC 99 VALUE 0.

```

פריטי-רמה-77 הדוגמה שלעיל מראה, כי ניתן לא להשתמש כלל ברמה-77 בתכנית. על ידי הגדרת פריט מקובץ ברמה 01, שנקרא כאן WORK-AREAS, ניתן להגדיר את כל הפריטים העצמאיים כחלק של פריט מקובץ זה.

חטיבת הביצוע PROCEDURE DIVISION

חטיבת הביצוע מכילה את כל ההוראות, אשר יבוצעו על ידי המחשב, וזאת על פי הלוגיקה, אשר נקבעה בתוכנית בעזרת הוראות אלה מטרתת של חטיבת הביצוע:

1. לגשת לקובץ קלט ופלט.
2. לקרוא ולכתוב מידע.
3. לבצע פעולות הצבה והעברה פשוטות.
4. לבצע פעולות סיום תכנית מסוימות.

כל ההוראות בחטיבת הביצוע מקודדות במרקווין A. כל שמות הסעיפים נכתבים במרקווין B.

הוראת OPEN

לפני שניתן לקרוא או לכתוב קובץ קלט או פלט, יש לפתוח (OPEN) את הקובץ. הוראת OPEN מורה למחשב לגשת (ACCESS) אל הקובץ.

name OPEN [INPUT (file name)] , [OUTPUT (file)]

FD תאר קבצים ושיר אליו ייחידות קלט/פלט ע"י משפט SELECT
הוראת - OPEN :

1. מאפיינת את הקבצים FILE-1 FILE-2 FILE-3 וFILE-1 FILE-2 FILE-3 וFILE-1.
2. מאפשרת גישה אל הקבצים לצורך עיבוד.

OPEN INPUT FILE-1,OUTPUT FILE-2.

לעתים קרובות, כתבים תוכניות, המשמשות במספר קבצי קלט או פלט. תוכנית עדכון (UPDATE), למשל, עשויה למaza (MERGE) שני קבצי קלט בשם FILE-1 FILE-2 ו FILE-3 אחד יחד בשם FILE-3, וכן להדפיס רשימת שגיאות בשם PRINT-FILE.

לדוגמ:

OPEN INPUT FILE-1,FILE-2, OUTPUT FILE-3, PRINT-FILE.**הוראת READ**

לאחר שפתחנו קובץ קלט, ניתן לבצע בו הוראת READ.

READ (שם-קובץ) AT END (שם-קובץ).

שם הקובץ, המופיע בהוראת READ, כבר מופיע בשלושה מקומות קודמים בתכנית:

1. select & file המצביע על שם הקובץ ויחידת הקלט/פלט.
2. FD VP המתאר את הקובץ.
3. open, המאפשרת גישה לקובץ.

תפקידה העיקרי של הוראת READ הינו להעביר רשומה אחת של נתונים אל האחסון; כלומר, בכל פעם שמתבצעת הוראת READ, נקראת רשומה אחת.

החלק AT END בהוראת READ בודק, אם יש נתונים נוספים. ביצורו עם הוראת READ, מורה למחשב מה עליו לעשות, במקרה שאין נתונים קלט נוספים. בדרך כלל, תיכתב הוראת READ כך:

READ FILE-1 AT END MOVE 1 TO EOF (שם-קובץ)

דוגמא

READ FILE-1 AT END MOVE 1 TO EOF.

בכל קובץ קיים בקובץ סימן לגבי סוף הקובץ EOF. סימן זה הוא שגורם לביצוע התנאי AT END.

יש לציין תנאי AT END בכל הוראות READ. כאמור, יש לציין למחשב, מה צריך לעשות כאשר נתונים הקיימים.

```
FD FILE-1
LABEL RECORDS ARE OMITTED,
RECORD CONTAINS 20 CHARACTERS,
BLOCK CONTAINS 10 RECORDS,
DATA RECORD IS REC.

01 REC.
02 NAME PIC X(15).
02 AMT-OF-TRANS PIC 9(5).
```

נניח כי מתבצעת הוראה:

.READ FILE-1 AT END MOVE 1 TO EOF

15 התווים הראשונים של הנתונים מהסרט יושמו באחסון השדה NAME. AMT-OF-TRANS. 5 התווים הבאים ימוקמו בשדה DATA DIVISION הוא בעל חשיבות רבה. הנתונים מוכנסים אל השדות לפי הסדר, בו מופיעים שמותיהם. פקודת READ תבצע גם פעולות בדיקה. אורך הגוש על הסרט יבדק, אם הוא מכיל 20 נתונים, או 10 רשומות, ברגע בו תבוצע קריאה של גוש. כל אי התאמה תגרום לשגיאת קלט/פלט מהסוג "אורך רשומה בלתי מתאים", אשר תגרום לטעינה.

הוראת PERFORM ... UNTIL

PERFORM (תנאי) UNTIL (שם-קטע) (שם-קובץ).

הוראה זו מעבירה את הבדיקה לקטוע, שבשמו היא נוקבת.

קטוע זה מבוצע על כל עמוד קטען, עד שפינט תקען, נתקען או נסתיים
כאשר מתקיים תנאי זה, יבוצע הפקודת קטען צנויין, דהיינו גורם ג

דוגמא

PERFORM CALC-RTN UNTIL EOF = 1 .

הוראת PERFORM מורה כי יש לבצע את כל ההוראות בקטוע CALC-RTN, עד אשר אין יותר רשומות קלט לעיבוד. אך תחזור הבדיקה להוראה שאחרי הוראת PERFORM .

סדר הצעדים:

```

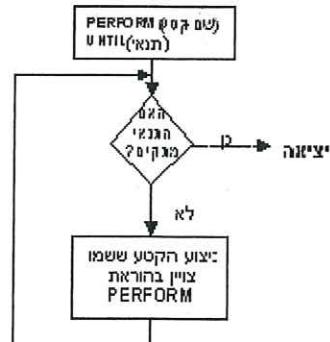
OPEN
READ
PERFORM
CLOSE
STOP
CALC-RTN.
{תהליך הביצוע
.READ FILE-1 AT END MOVE 1 TO EOF

```

התווחה (RANGE) של הוראת PERFORM כולל את כל ההוראות בקטוע שצויין, עד שפינט עמוד קטען.

או עד רצף של פינטים

בהוראת UNTIL מעורבת בדיקות-תנאי על ידי המחשב. למשל, חיבר למתכונים תנאי מסוימים.



בשימוש בהוראות: **UNTIL EOF = 1.** (שם סעיף) **UNTIL EOF = 1.** READ (שם סעיף) יש תמיד למקם את הוראת READ AT END. (אחרת יבוצע עיבוד על הפעולה الأخيرة פעמיים).

```

OPEN
READ
PERFORM (שם סעיף) UNTIL EOF = 1
...
(שם סעיף)
...
READ ... AT END MOVE 1 TO EOF.

```

הוראות CLOSE, STOP, RUN

פעולות סיום-התוכנית END OF JOB. פעולות אלה כוללות בדרך כלל שחרור של הקבצים בהם טיפולנו, והפסקת העיבוד. קיימות שתי הוראות, המהוות חלק הכרחי בכל שגרת סיום-התוכנית:

1. close - יש לסגור את כל הקבצים,

2. stop - יש להוראות למחשב להפסיק את מהלך התוכנית.

הוראת CLOSE**CLOSE (שם(ות) הקובץ(ים))**

לדוגמא:

CLOSE FILE-1, FILE-2, FILE-3.

.IN:

CLOSE FILE-1.

CLOSE FILE-2.

.CLOSE FILE-3

הוראת STOP RUN

הוראות STOP RUN מורה למחשב לסיים את עבודתו.

OPEN INPUT FILE-1, OUTPUT FILE-2.
 READ FILE-1 AT END MOVE 1 TO EOF.
 PERFORM CALC-RTN UNTIL EOF=1.
 CLOSE FILE-1,FILE-2.
.STOP RUN

ההוראות ב-CALC-RTN יכללו:

1. הוראות העיבוד לרשומה.
2. הוראת READ לקרוא הרשומה הבאה.
3. תנאי AT END, המורה למחשב להציב 1 ב-EOF.

הוראת MOVE פשוטה**MOVE (שם-נתון-2 TO (שם-נתון-1)**

כל שדה בזיכרון יכול להיות מועבר לשדה אחר באמצעות הוראת MOVE

CALC-RTN.
 MOVE NAME-IN TO NAME-OUT.
 MOVE AMT-OF-CREDIT-IN TO AMT-OF-CREDIT-OUT.
 .MOVE AMT-OF-DEBIT-IN TO AMT-OF-DEBIT-OUT

כדי להשלים את RN, יש לכתוב (WRITE) את הרשומה שניצטבנה בשטח הפלט, אז לקרוא את הרשומה הבאה מקובץ הקלט.

הוראת הכתיבה WRITE לוקחת את הנתונים, שניצטברו בשטח הפלט, ומעבירה אותם אל יחידת הקלט/פלט, אשר צוינה בחטיבת הסביבה.

WRITE (שם-רשומה)

שם הרשומה מופיע בرمאה 01, והוא מחולק, בדרך כלל, לשדות. הרשומה מתארת את תבנית FORMAT הפלט. במקרה WRITE - אנו מורים למחשב לכתוב את הנתונים לפי התבנית שצויינה.

FD FILE-1 LABEL RECORDS ARE OMITTED,
 RECORD CONTAINS 25 CHARACTERS,
 DATA RECORD IS REC-1.
 01 REC-1.
 02 NAME-OUT PIC X(15).
 02 AMT-OF-DEBIT-OUT PIC 9(5).
 02 AMT-OF-CREDIT-OUT PIC 9(5).

כדי לכתוב מידע המאוחסן בחטיבת הנתונים יש לכתוב WRITE REC-1 WRITE REC-1 ולא-1 WRITE FILE-1.

כasher כתבים או יוצרים מידע, יש להשתמש קטע (טבון)
 בעוד כאשר אנו קוראים מקובץ, יש להשתמש קטע (טבון)

הפקודה WRITE REC-1 מعتبرת נתונים אל יחידת הפלט, בהתאם לתבנית, שפורטה בתיאור הרשומה 1. אם קיימת יותר מבटנית אחת של רשומות בקובץ, יש לציין את זו, שבהם הם מעוניינים בהוראת WRITE.

לדוגמא,

FD PRINT-FILE LABEL RECORDS ARE OMITTED,
RECORD CONTAINS 133 CHARACTERS,
DATA RECORD ARE HDR, DETAIL-REC.

01 HDR.
...

01 DETAIL-REC.
...

לפי התבנית הרצiosa לנו, יש לכתוב Right DETAIL-REC Left HDR

PROCEDURE DIVISION.

OPEN INPUT FILE-1, OUTPUT FILE-2.
READ FILE-1 AT END MOVE 1 TO EOF.
PERFORM CALC-RTN UNTIL EOF=1.
CLOSE FILE-1,FILE-2.
STOP RUN.
CALC-RTN.
MOVE NAME-IN TO NAME-OUT.
MOVE AMT-OF-CREDIT-IN TO AMT-OF-CREDIT-OUT.
MOVE AMT-OF-DEBIT-IN TO AMT-OF-DEBIT-OUT.
WRITE REC-1
.READ FILE-1 AT END MOVE 1 TO EOF

הוראת הצבה (MOVE) רגילה

1. הוראת MOVE מספרית השדה השולח והשדה המתקבל, שניים מספריים.

כללים:

- החלק השלים מועבר מימין לשמאל תווים מסדר גובה, שאינם מתמלאים, יוחלפו באפסים.
- החלק העשרוני מועבר ממשמאליימן תווים מסדר נמוך, שלא התמלאו, יוחלפו באפסים.

2. הוראת MOVE, שאינה מספרית

השדה המתקבל אינו מספרי.

כללים:

מעובר ממשמאליימן
תווים מסדר נמוך, שלא התמלאו, יוחלפו בתווים רזוחות.

משפטית - VALUE.

משפט ה-E-VALUE יכול Kbיע או Kbיע מייחד, אשר יציב בשדה. הוא צריך להיות מאותו סוג, שהוגדר במשפט ה-PICTURE.
אם משפט ה-PICTURE הגדר שדה מספרי, הערך צריך להיות Kbיע מספרי, או אפס(ים) (ZEROS(S)).

הוראות אריתמטיות**א. הוראת חיבור**

ניתן לכתוב הוראת חיבור פשוטה, באמצעות שתי התבניות הבאות:

ADD { שם-נתון-1 } TO { שם-נתון-2 } { Kbיע }	ADD { שם-נתון-1 } { שם-נתון-2 } GIVING { שם-נתון-3 } { Kbיע-1 } { Kbיע-2 }	דוגמאות		
ADD TAX TO DEDUCTIONS. ADD 15.8 TO TAX. ADD 1.20, AMOUNT GIVING TOTAL. ADD AMT1, AMT2 GIVING NET				
		מבנה 1 - צורה מורחבת		
	ADD { שם-נתון-N } TO ... [שם-נתון-2] , { שם-נתון-1 } { Kbיע-2 } { Kbיע-1 }			
	ADD { שם-נתון-N } TO (שם-נתון-1+1), (שם-נתון-2), [שם-נתון-1] { Kbיע-2 } { Kbיע-1 }	מבנה 2 - צורה מורחבת		
		כללים לפועלות החיבור		
		<ol style="list-style-type: none"> 1. כל המשתנים והקבועים חייבים להיות נומריים. 2. מקבל התוצאה (שלאחר המלה TO או GIVING) חייב להיות משתנה, אסור שהוא קבוע. 3. בשימוש בתבנית מספר 1 (עם המלה TO) כל העריכים, כולל הערך של המשתנה התוצאה, מסוכמים יחדיו, ומואחסנים במשתנה התוצאה. 4. בשימוש בתבנית מספר 2 (עם GIVING), הערך של המשתנה התוצאה אינם מסוכם, ומשתנה זה משמש רק לאחסון והציגת התוצאה. 5. אסור, שבאותה הוראת חיבור יופיעו גם המלה TO, וגם המלה GIVING יחדיו. 		
		דוגמאות:		
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">.ADD AMT1, AMT2, AMT3 GIVING AMT4</td> </tr> <tr> <td style="padding: 5px; text-align: center;"> AMT1 2AMT 3AMT AMT4 2 4 6 15 לפני ביצוע ההוראה 2 4 6 15 אחרי ביצוע ההוראה </td> </tr> </table>			.ADD AMT1, AMT2, AMT3 GIVING AMT4	AMT1 2AMT 3AMT AMT4 2 4 6 15 לפני ביצוע ההוראה 2 4 6 15 אחרי ביצוע ההוראה
.ADD AMT1, AMT2, AMT3 GIVING AMT4				
AMT1 2AMT 3AMT AMT4 2 4 6 15 לפני ביצוע ההוראה 2 4 6 15 אחרי ביצוע ההוראה				

התוכן המספרי של שלושת המשתנים הראשונים אינם משתנה, אבל התוצאה, שנמצאת בשדה AMT4. (כמו בדוגמה א').

מכילה רק את ערכי המשתנים אלו. הערך המקורי של AMT4 אינו מסוכם, ואין לו כל השפעה על התוצאה.

דוגמא ב

ADD AMT1, AMT2, AMT3 TO AMT4.

AMT1	2AMT	3AMT	AMT4
------	------	------	------

2 4 6 15

2 4 6

לפני ביצוע ההוראה

אחרי ביצוע ההוראה

התוכן המספרי של שלושת המשתנים הראשונים אינם משתנה לאחר ביצוע הפעולה. התוצאה כוללת את הערכים של כל מקום שתבנית ההוראה ADD יכולה להסתכם בצורה הבאה:

הוראת חישוב

SUBTRACT {שם-נתון-ה} FROM ... [שם-נתון-2] , {שם-נתון-1}

{קבוע-2} {קבוע-1}

SUBTRACT AMT1, AMT2, AMT3 FROM TOTAL1, TOTAL2, TOTAL3.

או:

SUBTRACT {שם-נתון-ה} GIVING (שם-נתון-1) ... [שם-נתון-1]

{קבוע-1} {קבוע-1}

דוגמאות

SUBTRACT 25 FROM FLDA.

SUBTRACT FLDA, FLDB, 99 FROM FLDC.

SUBTRACT TAX FROM GROSS GIVING NET.

.SUBTRACT TAX, FICA, INSUR-PREM FROM GROSS GIVING NET

כללים לפעולת החישוב

1. כל המשתנים והקבועים חייבים להיות ממיריים.
2. שדה התוצאה (שאחרי המלה FROM בתבנית שימוש 1, או אחרי המלה GIVING בתבנית שימוש 2) חייב להיות משתנה, אסור שייהי קבוע.
3. בשימוש בתבנית מספר 1, ערכו כל המשתנים והקבועים שלפני המלה FROM מוסכמים יחד, והסיכון מופחת מהמשתנה שאחרי המלה FROM התוצאה (ההפרש) מאוחסנת במשתנה שאחרי המלה FROM . כל יתר המשתנים נשארים בערכיהם המקוריים.
4. בשימוש בתבנית מספר 2, הפעולה שמתבצעת זהה לפעולה, אשר הוסבירה בכלל מספר 3, אבל התוצאה מאוחסנת במשתנה שאחרי המלה GIVING, שערך המקורי אין שום השפעה על הפעולה. במקרה זה, גם במשתנה או בקבוע, שאחרי המלה GIVING, נשאר הערך המקורי.

דוגמאות:

דוגמא א

SUBTRACT 15.4, TAX, TOTAL FROM AMT.

TAX	TOTAL	AMT
-----	-------	-----

30 10 100

30 10 44.6

לפני ביצוע ההוראה

אחרי ביצוע ההוראה

דוגמא ב

SUBTRACT 15.4, TAX, TOTAL FROM AMT GIVING NET.

TAX	TOTAL	AMT	NET
-----	-------	-----	-----

30 10 100 87

לפני ביצוע ההוראה

30	10	100	אחרי ביצוע ההוראה
----	----	-----	-------------------

הוראות כפל וחילוק

MULTIPLY {שם-נתון-2} BY {שם-נתון-1} {קבוע-1} MULTIPLY {שם-נתון-3} GIVING {שם-נתון-2} BY {שם-נתון-1} {קבוע-2} {קבוע-1}	דוגמאות
MULTIPLY 100 BY QTY-ON-HAND. MULTIPLY QTY BY PRICE. MULTIPLY TAX BY GROSS GIVING DEDUCTIONS. MULTIPLY 150 BY DEPENDENTS GIVING STD-DED.	

הוראת החילוק הפешטה :

DIVIDE {שם-נתון-2} INTO {שם-נתון-1} {קבוע-1} DIVIDE {שם-נתון-3} GIVING {שם-נתון-2} BY {שם-נתון-1} {קבוע-2} {קבוע-1}	דוגמאות
DIVIDE 12 INTO ANN-SAL. DIVIDE N INTO SUM. DIVIDE 12 INTO ANN-SAL GIVING MONTHLY-SAL. DIVIDE N INTO SUM GIVING AVERAGE. DIVIDE ANN-SAL BY 12 GIVING MO-SAL.	

אנו רואים אפוא, שלכל ההוראות האריתמטיות ישן שתי תבניות כתיבה בסיסיות. במקרים רבים, שבסתניהם, המשתתפים בהוראה אריתמטית, אינם חשוב להמשך התכנית, ואין צורך לשמר אותו תוך כדי ביצוע התכנית, נשתמש בתבנית כתיבה מס' 1. לעומת זאת, במקרים שבהם, תוך כדי ביצוע ההוראה האריתמטית, חשוב לשמר על ערך מקורי, של משתנים, המשתתפים בהוראה מסוימת זו, הרי נשמש בתבנית כתיבה מס' 2 (GIVING).

בהוראות כפל או חילוק, שלא כמו בהוראות חיבור וחיסור, איןנו יכולים לבצע פעולה על הרבה קבועים ומשתנים ייחודיים. בכל הוראה, אך ורק שני קבועים או משתנים יוכפלו או יחולקו, כך שאם נctrar לדוגמה להכפיל את המשתנים והקבוע :

PRICE * 20 * QTY

1. MULTIPLY 20 BY PRICE
2. MULTIPLY PRICE BY QTY

הפקודות צרכות להיכתב זו אחר זו, וההתוצאה הסופית תהיה המכפלה של כל 3 המשתנים והקבוע, מאוחסנת במשתנה QTY. בהוראת הכפל נשמש תמיד במלחה השמורה BY, ואילו בהוראת החילוק אנו יכולים להשתמש במלחה השמורה INTO, וגם במלחה השמורה GIVING.

DIVIDE 3 INTO 6 GIVING AMT,
AMT, הרי התוצאה של 6/3, שהיא 2, תואחסן ב-AMT.
DIVIDE 6 BY 3 GIVING AMT.

דוגמא 1

הסבת מעלות צלסיוס למלגות פרנהיט מתבצעת על פי הנוסחה

$$F = C + 32(9/5)$$
 בשפת COBOL נכתבת:

MULTIPLY 9 BY C.
DIVIDE 5 INTO C.
ADD 32, C GIVING F

MULTIPLY 1.8 BY C.
ADD 32, C GIVING F.

כיוון ש- $9/5 = 1.8$, נוכל גם לכתוב כך:

דוגמא 2

חשב את ממוצע שלושת שדות הנתונים הבאים: HRS-WEEK2,HRS-WEEK3 ,HRS-WEEK1 . הצב את תשובה
בשדה הנתונים AVERAGE וdag לך שעריכיהם של שדות הנתונים המשמשים בסיס לחישוב לא ישתנו:
ADD HRS-WEEK1, HRS-WEEK2, HRS-WEEK3 GIVING AVERAGE.
DIVIDE 3 INTO AVERAGE.

דוגמה 3

מצאו את ערכו של שדה הנתונים C על פי הנוסחה הבאה:

$$C = A^2 + B^2$$

שוב, נמצא מຕך הינה ששדות הנתונים A, B, ו-C מוגדרים בחטיבת הנתונים.

פתרון:

MULTIPLY A BY A.

MULTIPLY B BY B.

.ADD A, B GIVING C

זכור: המכפלת שדה הנתונים A בעצמו (A * A) או רישום חזקה של A (A^2) אלה פעולות זהות מבחינה ביצוען.

(שארית) REMINDER

כאשר מתבצעת פעולה חילוק, התוצאה מאוחסנת במשתנה התוצאה, על פי הגדרתו ואפיונו; ולעתים קרובות מקבלים منها, וגם שרירות בסוף הפעולה. לדוגמה:

DIVIDE 130 BY 40 GIVING ITEM-A.

התוצאה: 03 מאוחסן ב-ITEM-A . שרירות: 10

DIVIDE {שם-נתון-3} BY {שם-נתון-2} {שם-נתון-1} (שם-נתון-3) GIVING {שם-נתון-2} {שם-נתון-1} (שם-נתון-2) INTO {קבוע-2} {קבוע-1} (שם-נתון-4) REMINDER

דוגמה

WORKING-STORAGE SECTION.

WORK-AREAS.

REMAIN-1 PIC 99.

...

PROCEDURE DIVISION.

...

.DIVIDE 130 BY 40 GIVING ITEM-A REMAINDER REMAIN-1

יש לציין, שהשימוש בצורה זו (עם REMINDER) אינו הכרחי כלל, ואין משפיע בשום צורה, על התוצאה של פעולה החילוק המקורי.

לסייעם להוראות האריתמטיות לצורת פעולה, נסתכל על הדוגמאות דלהלן, המציגות את ערכי המשתנים, לאחר ביצוע הפעולה:

D	C	B	A	פעולה אריתמטית
		A+B	A	ADD A TO B
A+B+C+D	C	B	A	ADD A, B, C TO D
A+B+C	C	B	A	ADD A, B, C GIVING D
		B-A	A	SUBTRACT A FROM B
$\frac{C}{(A+B)}$	B	A		SUBTRACT A, B FROM C
C-(A+B)	C	B	A	SUBTRACT A, B FROM C GIVING D
		A*B	A	MULTIPLY A BY B

A*B	B	A	MULTIPLY A BY B GIVING C
B/A	A		DIVIDE A INTO B
B/A	B	A	DIVIDE A INTO B GIVING C
A/B	B	A	

עיגול תוצאות חישוב (תקרה)

ADD AMT1, AMT2 GIVING TOTAL ROUNDED.

ADD AMT1, AMT2 GIVING TOTAL1 ROUNDED, TOTAL2 ROUNDED.

הוראת החישוב COMPUTE STATEMENT

ברוב המערכות המסחריות, המטפלות בנפח גדול של נתונים, ומעט מאוד בחישובים אРИתמטיים, ניתן להסתפק באربע ההוראות, שלמדנו עד כה. אולם, כאשר צריכים לכתוב תוכנית, שבה קיים שימוש רב יותר בחישובים ובפעולות אРИתמטיות, בהוראת COMPUTE, ניתן להשתמש בסמלים הבאים, עבור הפעולות האРИתמטיות:

+	חיבור
-	חיסור
*	כפל
/	חילוק
**	העלאה בחזקקה

נתבונן בשימוש בהוראה זו בדוגמאות דלהלן:

(1) COMPUTE TAX = .05 * AMT

(2) COMPUTE A = B * C / D

(3) COMPUTE NET = AMT - .05 * AMT

COMPUTE TOTAL = AMT1 + AMT2 - AMT3.

כל הסימנים האРИתמטיים, המופיעים בהוראת החישוב, חייבים להתנהג לפי חוקי המרוחך; כלומר - לכל סימן יש מרוחך אחד לפני ואחד אחריו (כולל סימן השווון). אם נרצה לכתוב $A = B+C+D$, הרי ב-COBOL יופיעו מרוחכים כדלקמן:

2 COMPUTE A = B + C + D +

במקרים הבאים נעדר להשתמש בהוראה MOVE, הפונה יחסית, לצורך ייעילות התוכנית (ביצוע מהיר יותר):

1. הוראה אРИתמטית, לדוגמא:

COMPUTE SALARY = HRS * RATE

2. קבוע, לדוגמא:

COMPUTE TAX = .05

3. שמות נתונים, לדוגמא:

COMPUTE FLDA = FLDB.

כמו באربע ההוראות הקודמות, ניתן להשתמש באפשרות העיגול, לפי אותן חוקים שלמדנו עד כה. הוראה זו, עם אפשרות

העיגול, נכתבת כך:

COMPUTE [ROUNDING] = [] (שם-נתון) (שם-נתון)

קבוע
ביטוי חשבוני

דוגמא

(a) COMPUTE A = B + C + D

(b) COMPUTE A ROUNDED = B + C + D

תרגילים

1. כתבי תוכנית המקבלת קלט רדיוס של מעגל ומדפיסה את הקפו ואת שטחו (השתמשי ב-4 ההוראות בסיסיות).

2. כתבי תוכנית המקבלת מספר ומדפיסה את השורש (חזקקה של 0.5) – עגלי את התוצאה, הרבע והמכפלה של

המספר במספר העוקב אחריו (השתמשי בהוראה COMPUTE).

display = cout
accept = cin
compute = pow

perform

חוקי הנקודה ב COBOL

1. בסיום כל שם של DIVISION.
2. בסיום כל שם של SECTION.
3. כל שם של PARAGRAPH מסתיים בנקודה.
4. כל הגדירה ב DATA מסתיימת בנקודה.
5. נקודה בסיום הפקודה الأخيرة של ה PARAGRAPH.

ערכים מותניים – CONDITION NAMES - רמה 88

רמה 88 - היא שם של תנאי שיש לה תמיד ערך TRUE או FALSE. יש להגדיר את המשתנה עליו רוצים לשאול ולשים לו משתנים עם ערכים ברמה 88 עליהם נשאל בתוכנית. בתוכנית נשאל על המשתנה ברמה 88 ישרות. אין לשאול על מצב המשתנה אם לא נתנו לו שם ברמה 88.

דוגמא:

```
01 SW-SEARCH.
 88 SW-FOUND      VALUE 1.
 88 SW-NOT-FOUND VALUE 0.
```

ניתן גם להגדיר טווח:

דוגמא 1

```
01 AGE PIC 9(3).
 88 AGE-KATIN VALUE 0 THRU 18.
 88 AGE-BOGER VALUE 19 THRU 65.
 88 AGE-GIMLAI VALUE 66 THRU 120.
 88 AGE-TAKIN VALUE 0 THRU 120.
```

דוגמא 2

```
88 AGE-AGOL VALUE 10 20 30 40 50.
```

אין אפשרות לעשות MOVE למספר משומש שהוא כמו תנאי.

VALUE TO SET (שם-מצב – רמה 88) – מכון את המצב הזה להיותאמת ע"י שמעביר את ה VALUE

למשתנה הראשי ה GROUP ITEM

צומת

נקודה בתוכנית שבה ניתן לבחור דרך אחת מתוך N דרכים, לבצע מה שנאמר ולהמשיך בצורה מאוחדת.

```
EVALUATE TRUE
when num > 0
  DISPLAY STR-POS
when num < 0
  display STR-NEG
when other
  CONTINUE
END-EVALUATE.
```

ה WHEN מתייחס אחד לשני ביחס של OR עד שאחד מהתנאים מתקיים ואז עבר ל END. אם לא נמצא עבור WHEN OTHER ווגמר, אם לא כתוב WHEN OTHER עבר הלאה בתוכנית.

קבצים

ב-פיזי קובץ פיזי (Physical file) הוא קובץ של נתונים אטום (atomic), כלומר לא ניתן לחלק אותו לשני חלקים.

סוגי קבצים

קובול מכיר שני סוגים:

1. **קובץ פיזי** – יוצרים נתונים על גבי ה Hard disk. דוגמאות – סדרת רשומות מודפסות, קובץ סדרתי – סדרה של שורות היושבות אחת אחרי השניה ולא ניתן לחלק רשומה לפי בחירה אלא רק לפי סדר ההגעה.
2. **קובץ סדרתי** – מועבד בצורה ברורה עם עובדה רשומה אחת אחרי השניה. קובץ סדרתי מועבד בצורה ברורה עם עובדה רשומה אחת אחרי השניה.
3. **קובץ אקראי** – לכל רשומה יש מפתח מיוחד אחד ערכיו אפשרויות העיבוד הם גם לבחור רשומה לפי מפתח ואז הפעולה תבוצע רק על הרשימה הנדרשת אך יש אפשרות לעובד גם בצורה סדרתית השינוי יהיה רק בעיבוד ולא בקובץ עצמו. –RELATIVE – העיקרון הטבלי ע"ג אחסון חיצוני המערכת יודעת למצאו רשומות לפני היחס. INDEXED – קבצים שיוצרים למערכת אך הוא יכול כי יש הפרדה בין הטבלאות של הנתונים לבין הטבלאות של האינדקסים.

פעולות**.REWRITE, WRITE, READ, CLOSE, OPEN**

כל פעולה סידור שמתבצעת המערכת זורקת בסופו סטטוס שאומר מה מצבו של הקובץ אחרי הפעולה, ולכן היא מחייבת לבירר האם עבר בשולם ואז יש סיכוי שהמערכת לא תפעיל אותו אלא תנתן לנו אפשרות לעזoor את הניפויה. כאשר מגדרים קובץ למערכת יש להגדיר לאיזה משתנה לזרוק את הסטטוס. המשתנה יהיה מסוג XX PIC. אנו מצפים לקבל סטטוס "00" אחריו כל פעולה. לתוכנית קובל יש משתנה סטטוס אחד ברגע נתון יכולה להתבצע רק פעולה אחת ולכן סטטוס שומר את כלם.

טיפול בתקלות - Section**הבדלים בין תוכנית ורילה לתוכנית עם קבצים**

1. ENVIRONMENT DIVISION (ENVIRONMET DIVISION) – מגדיר תקודה אינטראקטיבית, דוחים, פונקציות ועוד.
2. DATA DIVISION – מגדיר משתנים, קבועים, פונקציות ועוד.
3. PROCEDURE DIVISION – מגדיר פונקציות, מערכות, מדדים ועוד.

כללים

1. לען פונקציית open() – פונקציית open()
2. לען פונקציית close() – פונקציית close()
3. לען פונקציית read() – פונקציית read()
4. לען פונקציית write() – פונקציית write()

תרגילים

1. כתבי תוכנית הקולטת נתונים עובדים לתוכן קובץ עובדים: שם פרטי, שם משפחה, ת"ז, תעריף לשעה, מס שעוט יומי ומס שעוט חדש.
2. תוכנית שנייה המחשבת את משכורתו של העובד, תעריף לשעה * מס' שעות. ומוציאת דוח משכורות.
3. בתוכהן קובל אוסף נתונים של לקוחות ולקוחות על כל לקוח

מערכות- טבלאות

מבנה נתונים ברענון של ארון מגירות לכל המגירות יש אותו שם גודל אופי וחס אך כל אחד אם אינדקס אחר.
האינדקס מתחילה 1, אין לפנות למגירה אפס וכן אין לפנות למגירות שמעבר להגדלה. לכל תא יש כניסה ורכיב.

הגדרת טבלה :

01 TV-A.

03 A PIC 9(3) OCCURS 100.

אם נפנה עכשו ל A בלבד זה יהיה שגיאת קומפילציה ושגיאת ריצה, חיבים לפנות ל A עם אינדקס שצריך להיות בין 1 ל 100.

איפוס לטבלה נומרית MOVE ALL ZERO. -

01 TV01-MISPARIM.

03 TV01-MIS PIC 9(3) OCCURS 100.

הפנייה ל MIS היא נומרית ולא MISPARIM היא אלף נומרית.

טבלאות דו ממדיות

טבלה של כל השדות בה הם דו ממדים לכל שדה יש שני רכיבים, יש חלוקה ומתחם עוד חלוקה.

01 TV-00.

03 TV01-OVDIM.

05 TV01-OVED OCCURS 100.

07 TV01-SHEM PIC X(20).

07 TV01-CHODESH OCCURS 12.

09 TV01-SHAOT PIC 9(3).

09 TV01- MASKORET PIC 9(5).

מספר האינדקסים לרכיב כמספר פעמים שהmillion OCCURS הזרת על עצמה.

האינדקס הראשון הוא ה OCCURS הראשון.

ADD 500 TO TV01-MASKORET(17 3) - עובד 17 בחודש 3.

1. יש לכתוב תוכנית לנוהל מערכת אוטובוסים, בגודל 5.

פרט אוטובוס:

מס' קוו, תחנת יציאה, תחנה סופית ותדיות לשעה.

תחנה מוגדרת:

1-תחנה מרכזית

2. מסוף

3. בנין אגד

יש לקלוט את הנתונים ואח"כ להדפים.

מצאי את הקוו בעל התדיות הגבוהה ביותר.

מצאי את הקווים שיוצאים מתחנה מרכזית.

ומצאי את הקווים שmaguis למסוף.

שדות עריכה

שדות אלו הן מעין תבנית שאנו שולחת אליהם שדה גלמי ותפקידו ליפוט אותו כך שהלקוח יוכל להבין את הערך המספרי שלו בלי לדעת איזה סוג של שדה זה היה.

שדות עריכה מוגדרים בפלט או בשורות הדפסה.

כדי ליצור עריכה חביב להיות משתנה פרט依 Item .

מגדירים משתנה שקוראים לו שם ובPIC שלו מגדירים את המבנה ואז בתוכנית עושים לו MOVE .
בשדות עריכה כתובים את ה 9 במספר מלא ולא קיצור .

שדה העריכה הזאת מקבל את התוכן ע"י פקודת MOVE בזמן זה הוא מתנהג כשדה נומרי אבל מהרגע שהתוכן נכנס לשדה העריכה זה כבר לא נחשב לנומרי לפחות אלא רק לתצוגה .

הפקודות השימושיות זה DISPLAY , MOVE .

עריכה להדפסה:

1. **הוספת תוים / O B** - במקום שבו כתבת את התו שם הוא יופיע חוץ מ B שבמוקומו יופיע רווח BLANK .

2. **נקודה עשרונית** – במקומות שנכתבו אותה שם היא תציג , נקודה עשרונית יש רק אחת בזמן ה MOVE הנקודה העשרונית מתוישבת על ה 7 ומה שנשים משמאלי יכנסו משמאלי וכן בימין . מותר לשלב הוספת 2 תוים כמו 99,999.99 .

3. **עריכה קבועה ע"י - + \$** - הסימן יהיה במקומות קבוע ערך עזרת הסימן \$ - יכול לבוא רק משמאלי של השדה ,

עריכה בעזרת הסימן + מתוישת לסימן של השדה התוכן שיוצג הוא הסימן לא משנה איזה 99+ .
עריכה בעזרת הסימן – שימוש במינוס אומר הצג סימן אם הוא שלילי והציג רוחם אם הסימן חיובי 99- המשמעות היא כאן רק לחיבורים שלא נראה מה יש בבית הריבועי .

עריכה קבועה ניתן לבחור סימן בצד ימין או משמאלי חוץ מ \$ רק משמאלי .

4. **דיכוי אפסים מוביילים Z *** - שימוש בתו Z - כל Z יכול להיות מוחלף בספרה לחילופין יכול להיות רוחם אם הוא אף מוביל שימוש בתו * - כל אפס מוביל משאיר אחריו *

5. **עריכה צפה - + \$** - לעריכה צפה יש אפשרות לדכא אפסים מוביילים חוץ מהסימן השמאלי ביזור שלעלום לא יופיע להיות ספרה או רווח אם זה אפס מוביל או הסימן \$ בשדות עריכה צפה מס' הספרות שהם מוכנים להציג זה אחד פחות מסופר הספרות .

モותר לשלב סימני ערכה כל עוד הם אינם מטפלים באותה בעיה .

כל תווים ההווספה בטור השדות המדכאים אפסים מוביילים ספרה ממידה ואין מושאלם ספרה מוביל הסימן לא יוצג אלא יוצג במקומו בתו שסבירו .
דוגמאות:

9,ZZ,ZZ- הפסיק הופך להיות כמו ה Z .

***, ** pic *** - הפסיק הופך להוית * .

+++, ++ PIC - הפסיק יוצג רק אם יש סימן אלפיים ולא عشرות אלפיים כי הוא סימן פה ולא קיים בטור ספרה .

---- PIC - מקבל מקסימום 3 ספרות ספרה האחרונה מוכנה להיפוך לרוחם אם המספר חיובי אך לא ספרה נסفة .

1. כתבי תבנית להדפסת רשומה בגודל 80 תוים :

5-1 מספר המוצר (5) X

8-6 מספר המחסן (3) X

11-9 כמהות יכולה להיות שלילית במקרה של זכות . 0000000000

13-12 קוד מוצר X X

18-14 מחיר יחידה (6) 0/00000

38-19 תיאור המוצר (6) X

41-39 אходו הנחה 0000000000

46-42 מספר לקוח (6) X

80-47 לא בשימוש (3) X FILVER AL PIC

REFERENCE MODIFICATION - שינוי התייחסות

ניתן לשנות את צורת התייחסות למחוזת, ע"י הוספת סוגרים לאחר המוחזרות עם שתי ערכים ונקודותים ביניהם.
שני המספרים בסוגרים מיצגים תת מחוזת בתוך מחוזת.
הראשון מאייזה מספרתו והשני אומר אורך תת מחוזת - (אורך: מאייזה תו)
בסוגרים יכולם להיות: 1. מספר 2. משתנה נומי שיש בו מספר שלם.

Move B(2:3) to A(4:3)

משתנה	סוג	ערך המקורי	ערך לאחר ביצוע
A	X(8)	"abc <u>defgh</u> "	a <u>b</u> c x <u>y</u> z <u>g</u> h
B	X(4)	"w <u>xyz</u> "	w <u>x</u> y <u>z</u>

טבלה 12.2 – הסמלים בהם אפשר להשתמש במשפט PICTURE

המשמעות	הסמל
שדה אלפאנומרי	X
שדה נומי	9
שדה אלפיבטי	A
נקודה עשרונית מוכללת; לשימוש בשדות מספריים בלבד	V
סימן השדה; לשימוש בשדות מספריים בלבד	S
מקום עשורוני משוער; לשימוש בשדות מספריים בלבד	P
טו להשמת אפס	Z
נקודה עשרונית	.
סימן חיובי	+
סימן שלילי	-
סמל הדולר	\$
פסיק לקטוע סדרות של מספרים	,
סמל ה"זכות" (CREDIT) בחשבונאות	CR
סמל ה"חובה" (DEBIT) בחשבונאות	DB
סמל להגנת המהאות	*
טורהפרדה בתוך שדה – להכנסתטו רווח	B
טו להכנסת 0	0

שדה שולח		שדה הדפסה	
PICTURE	התובן	PICTURE	תוצאת העריכה
9(6)	123456	\$ZZZ.ZZZ.99	\$123,456.00
9999V99	001234	\$Z.ZZZ.99	\$ 12.34
9(5)V99	0000123	\$**.***.99	\$****1.23
S9(6)	012345 (שלילי)	+Z(6)	- 12345
S9(6)	123456 (חיובי)	-Z(6)	123456
S9999V99	123456 (חיובי)	+Z(4).99	+ 1234.56
S999	123 (שלילי)	ZZZ-	123-
9(6)	123456	99BBBB9999	12 3456
S99	05 (שלילי)	\$ZZ.99DB	\$ 5.00DB
999	123	999000	123000
S99V99	1234 (שלילי)	\$ZZ.99CR	\$12.34CR

מלא את החסר בעמודות המותאמות:

שדה השולח		שדה הדפסה	
PICTURE	תובן	PICTURE	תובן
1. 99V99	0467	ZZ.99	\$.47
2. 9(4)V999	0086754	\$Z.ZZZ.99	\$ 46.25
3. 999	467	\$ZZ.99	\$ 467.00
4. S99V99	0098 (שלילי)	\$ZZ.99+	\$ 98.00+
5. S99V99	0089 (חיובי)	\$ZZ.99-	\$ 89.00-
6. S999	005 (חיוב)	\$ZZZ.99CR	\$ 5.00
7. S999	005 (חיוב)	\$ZZZ.99DB	\$ 5.00 DB
8. S99V99	0006 (חיוב)	\$ZZ.99CR	\$.06
9. S99V99	0005 (חיוב)	\$**.99-	\$ * .06
10. 9(4)	1357	\$**.99	\$ * .06
11. XXXX	CRDB	\$*.***.99	\$.1354.00
12. 9(4)	0170	XXBBXX	C R DB
13. 999V99	13579	99BB99	0 A 23
14. 999V99	00009	\$\$\$\$.99	\$ 135.23
15. 999V99	00000	\$\$\$\$.99	\$.09
16. S9(5)	00567 (שלילי)	+++++	\$.00
17. S99V99	0034 (חיוב)	\$\$\$\$.99-	\$.00
18. S99	00 (חיוב)	++	- \$0
19. S99	00 (חיוב)	--	\$.30
20. 9999V99	000988	\$2.99	\$ 9.88
21. 9999V99	000988	\$22.99	\$ 9.88
22. 99V99	8738	-\$22.99	+\$87.38