**יום ראשון, חיים, 21.08.2022**

**mm- c language הזזת ביטים**

<< - הזזת ביט ימינה (מחלק ב-2).

>> - הזזת ביט שמאלה (מכפיל ב-2).

כשמסתכלים על מספר בינארי איך נדע אם הוא שלילי או חיובי?

1. נבדוק אם הוא sign/unsigned.
2. במקרה שunsigned- הספרה השמאלית ביותר קובעת סימן ושאר המספרים את ערך המספר.
3. במקרה שsign- המספר חיובי, וכל הספרות קובעות את ערך המספר.

דוגמא: 1111 1111

בsign: -1.

בunsigned: 255.

פקודה מקוצרת: num=num<<1 🡪 num<<=1

**פעולות לוגיות על ביטים**

השוואה בין ביט לביט כמתואר בסיכום שיעור 16.08 אלון.

And=&&

Or=||

Xor=^

* Xor על עצמו מאפס את המספר.

מינוח מקצועי לפעולת מיסוך ביטים= mask.

**המרה בין טיפוסים- typecast**

* קיבולת (int=4 byte, char=1 byte).
* ערך

ניתן לשים ערך של משתנה עם קיבולת קטנה במשתנה עם קיבולת גדולה יותר.

אך לא ההפך (חוץ מאשר במקרה הבא)

**תהליך casting: פעולת typecast**

לדוגמא= int bignum=3;

Char num= 20;

num=(char)bignum;

* ניתן יהיה להמיר אך אם הערך גדול מדי מקיבולת המשתנה, הערך יחתך.
* יחתך= לפי מספר הספרות הבינאריות המסוגלות להיכנס למשתנה.

למשל: במקרה שלמעלה יחתכו 3 בתים סך הכל 24 ביטים שהם 24 ספרות בינאריות.

יש שני סוגי המרות

1. המרה מפורשת: explicit. ---> כאשר נותנים פקודה להמיר באופן מפורש num=(char)b
2. המרה מרומזת: implicit. ---> כאשר יש המרה מאחורי הקלעים כלומר השמנו משתנה מסוג אחד למשתנה מסוג שני, הקומפיילר כבר עושה בעצמו את ההמרה.