הדפסת מספרים ממשיים ב++C

כידוע, מספרים ממשיים עלולים להיות אינסופיים, ואנחנו לא רוצים להדפיס מספרים אינסופיים. לכן כשמדפיסים מספרים ממשיים, המחשב מעגל אותם בהתאם ל**רמת הדיוק** שבחרנו. המשמעות של "רמת הדיוק" (precision) היא מספר הספרות המשמעותיות שמודפסות – מספר הספרות בין הספרה השמאלית ביותר שאינה 0 לספרה הימנית ביותר שאינה 0. לדוגמה:

- במספר 7600 יש שתי ספרות משמעותיות – 7, 6.
- גם במספר 0.0076 יש שתי ספרות משמעותיות – 7, 6.
- במספר 7600.0076 יש שמונה ספרות משמעותיות.

ברירת-המחדל של רמת-הדיוק בהדפסה ל-cout היא **6**. אם מדפיסים מספר עם יותר מ-6 ספרות משמעותיות – נראה רק 6. למשל:

- `cout << 1234.5678;`

מדפיס **1234.57**: במספר המקורי יש 8 ספרות משמעותיות, המחשב חותך 2 ומעגל (במקרה זה מעגל למעלה)

- `cout << 12345678.;`

מדפיס **1.23457e+07**: שוב המחשב חותך שתי ספרות ומעגל, אבל במקום לכתוב 12345700, כדי לחסוך באפסים, הוא כותב רק את 6 הספרות המשמעותיות, ומוסיף את האקספוננט e+07 (שמשמעותו: "כפול 10 בחזקת 7").

אפשר לשנות את רמת-הדיוק ע"י הפקודה setprecision שנמצאת בכותרת <iomanip> - ראו בתיקיה 8. למשל, אם נשנה את רמת הדיוק ל-4, נקבל **1235** ו **1.235e+07**; אבל אם נשנה ל-10, נקבל רק 8 ספרות משמעותיות – **1234.5678** ו **12345678**.

אם רמת-דיוק היא גבוהה מאד, אנחנו עלולים לקבל תוצאות לא צפויות. למשל, ברמת דיוק 100 מקבלים:
1234.5678000000000338332029059529304504394531 25
מדוע? - כי יש אינסוף מספרים ורק מספר סופי של ייצוגים של מספרים בסיביות. לכן, לרוב המספרים אין ייצוג בסיביות, והמחשב מעגל אותם לייצוג הקרוב ביותר. למשל, למספר 1234.5678 אין ייצוג בסיביות, והמספר הקרוב ביותר שיש לו ייצוג בסיביות (לפחות על המחשב שלי) הוא המספר הנ"ל.
כשרמת הדיוק היא נמוכה, אנחנו לא שמים לב לזה, כי בתצוגה המספר מתעגל ל 1234.5678. אבל כשרמת הדיוק גבוהה, אנחנו רואים את המספר כפי שהוא באמת.

מקורות

- 1) Floating-Point Determinism ,
- 2) What Every Computer Scientist Should Know About Floating-Point Arithmetic ,
- 3) Is floating point math broken? .
- https://en.cppreference.com/w/cpp/io/ios_base/precision

סיכם: אראל סגל-הלוי