ייצוג מספרי נקודה צפה (floating point)

הייצוג הנפוץ ביותר, אותו ראיתם בהרצאות, מוגדר על פי IEEE 754 Standard.

מספרי floating point הינם בעלי דיוק מוגבל. הם מיוצגים בבסיס בינארי , כסכום של חזקות חיוביות ושליליות של 2. כמובן שלא כל מספר ניתן לייצג באמצעות סכום של חזקות של 2 ועבור מספרים כאלה נחזיק קירוב למספר ולא את הערך עצמו. לדוגמא, ראינו בכיתה את המספר 0.1 שלא ניתן לייצג בעזרת סכום של חזקות של 2 ולכן מיוצג כקירוב.

נניח כעת שאנו עובדים עם מחשב 64 ביט (גודל מילה הוא 64 ביט).

מספר floating point יוחזק באופן הבא:

$$sign \cdot 2^{exponent-1023} \cdot (1 + fraction)$$

- $sign$ ייוצג ע"י ביט בודד (הביט השמאלי ביותר) : ערך 1 יתאים למספר שלילי ואז הביטוי יראה כך:

$(-1) \cdot 2^{exponent-1023} \cdot (1 + fraction)$, וערך 0 יתאים למספר חיובי ואז הביטוי יראה כך: $2^{exponent-1023} \cdot (1 + fraction)$.

- exponent הינו מספר שלם אי שלילי שייוצג ע"י 11 הביטים הבאים. הערכים האפשריים של exponent הם 0 עד $2^{11} - 1 = 2047$. זה אומר שהביטוי $2^{exponent-1023}$ יקבל ערך מינימלי 2^{-1023} עבור exponent =0, וערך מקסימלי 2^{1024} עבור exponent=2047.
- fraction הינו סכום של חזקות שליליות של 2, שמיוצג ע"י 52 הביטים הבאים.
לדוגמא:
- אם 52 הביטים הם :
1000000000000000000000000000000000000000000000000000
אז הסכום הינו $1 * 2^{-1} + \sum_{i=2}^{52}(0 * 2^{-i}) = 0.5$
במקרה כזה ערכו של הביטוי $(1 + fraction)$ יהיה 1.5.
- כדי שערכו של $(1 + fraction)$ יהיה $1.9843751 + \sum_{i=1}^{6}\frac{1}{2^i}$ = צריך ש fraction ייצג את 0.984375 (כלומר את הסכום $\sum_{i=1}^{6}\frac{1}{2^i}$). במקרה כזה 52 הביטים של fraction יראו כך:
1111110000000000000000000000000000000000000000000000

שימו לב שהוגדרו מספר מקרים מיוחדים שחורגים מהתבנית שהוסברה לעיל, ביניהם:
- אם exponent מורכב כולו מ 0-ים ו - faction מורכב כולו 0-ים אז בהתאם ל sign נייצג את 0.0 ואת 0.0- .
- אם exponent מורכב כולו מ 1-ים ו- fraction מורכב כולו מ 0-ים אז בהתאם ל sign נייצג את "פלוס-אינסוף" ו:"מינוס-אינסוף". למען האמת, המספר המקסימלי שאינו "מקרה מיוחד" יכיל exponent שמתאים לערך $2^{11} - 2 = 2046$, כלומר יהיה גדול מ 2^{1023} וקטן מ 2^{1024}.

נשים לב שלכל מספר שאינו חורג מהתבנית שהוסברה לעיל, למעט 0, יש ייצוג יחיד .

נשתמש בפונקציה *display_float* (הקוד מההרצאה, אינכם צריכים להבין קוד זה) ונראה כמה דוגמאות:

נייצג את *0. 5* ע"י הביטוי הבא: $2^{-1} * (1 + 0)$

>>> display_float(0.5)

'0 01111111110 0000000000000000000000000000000000000000000000000000'

sign הינו 0 ולכן מתאים למספר חיובי

exponent מתאים ל 1022 ולכן מתקיים ש: $2^{1022-1023} = 2^{-1}$

fraction יכיל 0 בכל הביטים ולכן הסכום $\sum_{i=1}^{52} \frac{1}{2^i}$ יתאים ל 0 כנדרש.

נייצג את *0.75* ע"י הביטוי הבא: $2^{-1} * (1 + 0.5)$

>>> display_float(0.75)

'0 01111111110 1000000000000000000000000000000000000000000000000000'

sign הינו 0 ולכן מתאים למספר חיובי

exponent מתאים ל 1022 ולכן מתקיים ש: $2^{1022-1023} = 2^{-1}$

fraction יכיל 1 בביט הראשון ולכן הסכום $\sum_{i=1}^{52} \frac{1}{2^i}$ יתאים ל *0.5* כנדרש.

>>> display_float(-0.75)

'1 01111111110 1000000000000000000000000000000000000000000000000000'

כאן sign הינו 1 ולכן מתאים למספר שלילי.

נייצג את 5.5 ע"י הביטוי הבא: $2^2 * \left(1 + (2^{-2} + 2^{-3})\right) = 2^2 * (1 + (0.25 + 0.125))$

>>> display_float(5.5)

'0 10000000001 0110000000000000000000000000000000000000000000000000'

>>> int("10000000001", 2)

1025

sign הינו 0 ולכן מתאים למספר חיובי

exponent מתאים ל 1025 ולכן מתקיים ש: $2^{1025-1023} = 2^2$

fraction יכיל 1 בביטים השני והשלישי ולכן הסכום $\sum_{i=1}^{52} \frac{1}{2^i}$ יתאים ל *0.25+0.125* כנדרש.