נספח עבודה

הקדמה:

וו int ,char ,boolean כזכור לכל משתנה מטיפוס פרימיטיבי יש הקצאה של n ביטים בזיכרון. עבור עבור , 10 , 10 ו int וש הקצאה של 1, 16, 18 ו – 64 ביטים בזיכרון בהתאמה. נתמקד במשתנה מטיפוס double – עם הקצאה של n=32 ביטיים שונים הנבדלים על ידי עם הקצאה של n=32 ביטיים, לדוגמא ניתן לייצג את הטווח n=32 המכיל מספרים שלמים אי-שליליים מבסיס רצף הביטים, לדוגמא ניתן לייצג את הטווח n=32 ביטיים שלמים אי-שליליים מבסיס רצף הביטים.

ברירת המחדל של int בשפת JAVA היא שרצפי הביטים השונים מייצגים מספרים שלמים הן ברירת המחדל של (Two's complement) עליה שליליים והן אי-שליליים. שיטת ייצוג זו נקראת שיטת המשלים ל-2 (most significant bit–MSB) מציין את נפרט בהמשך. בשיטה זו הביט במיקום השמאלי ביותר (most significant bit–MSB) מציין את הסימון, כלומר אם המספר בבסיס 10 הוא אי-שלילי או שלילי. כאשר הביט השמאלי ביותר הוא המספר הוא שלילי, וכאשר הוא 0 המספר הוא מספר אי-שלילי. לפי האיור הבא:

<u>(in</u>	ייצוג בבסיס 10			
01111111	11111111	11111111	11111111	2147483647
01111111	11111111	11111111	11111110	2147483646
01111111	11111111	11111111	11111101	2147483645
0000000	00000000	00000000	00000011	3
0000000	00000000	0000000	0000010	2
0000000	0000000	0000000	0000001	1
0000000	00000000	00000000	0000000	0
1 1111111	11111111	11111111	11111111	-1
1 1111111	11111111	11111111	11111110	-2
1 1111111	11111111	11111111	11111101	-3
1 1111111	11111111	11111111	11111100	-4
1 0000000	00000000	00000000	00000010	-2147483646
1 0000000	00000000	00000000	0000001	-2147483647
10000000	00000000	00000000	00000000	-2147483648

אמנם אנו רגילים להשתמש ב- n עם 32 n עבור ייצוג מספרים שלמים בבסיס 10, אך באופן כללי n אמנם אנו רגילים להשתמש ב- n ואף n בסיס n בשיטת המשלים ל-2 גם עבור n אוף n ואף n ביתן לייצג מספרים שלמים בבסיס 10

כאשר אנו נעזרים ב n כלשהו לייצג מספרים בבסיס 10 בשיטת המשלים ל -2 נוכל להגדיר את טווח n כאשר אנו נעזרים ב $-2^{n-1},2^{n-1}=1$.

למעשה עם שימוש ב- n שונים ניתן לייצג את אותו מספר בבסיס 10 רק עם רשימת ביטים באורך n בולגשה שהוא נמצא בטווח של ה – n בו הוא מוגדר. לדוגמא:

n = 32

$$n = 4$$
, $[-2^3, 2^3 - 1]$
 $n = 8$, $[-2^7, 2^7 - 1]$
 $n = 32$, $[-2^{31}, 2^{31} - 1]$

$$(5)_{10} = (00000000|00000000|00000000|00000101)_2 = (00000101)_2 = (0101)_2$$

$$(0)_{10} = (00000000|00000000|00000000|00000000)_2 = (00000000)_2 = (0000)_2$$

$$(1000033)_{10} = (00000000|00001111|01000010|01100001)_2 = \frac{(01100001)_2}{(0001)_2} = \frac{(0001)_2}{(0001)_2}$$

$$(-1000033)_{10} = (11111111|11110000|10111101|10011111)_2 = \frac{(10011111)_2}{(1100)_2} = \frac{(1100)_2}{(1100)_2} = \frac{(1100)_2}{$$

מכאן ניתן לראות שלכל מספר בבסיס 10 יש **ייצוג בינארי מינימאלי** בביטים. כלומר את 5 ניתן לייצג עם לראות שלכל מספר בבסיס 12. אם היינו מנסים לייצג את 5 עם n=3 בלבד בשיטת המשלים ל-2. אם היינו מנסים לייצג את 5 עם n=3 בינארי (101) שהייצוג העשרוני שלו הוא 3- בשיטת המשלים ל

ייצוג בינארי מינימאלי:

n=4

<u>n=8</u>

לאחר שראינו ביטוי של ייצוג בינארי מינימאלי מהדוגמא הקודמת נפנה כעת להגדיר את המושג. n – באופן פורמאלי, **ייצוג בינארי מינימאלי** הוא הייצוג של מספר בבסיס 10 בייצוג בינארי כך שה (מספר הביטים) איתו הוא מיוצג הינו הקטן ביותר.

הוא הקטן n=4 פרה 5 ניתן לייצג על ידי ייצוגים בינארים שונים $^{\rm n=4,8,32}$ כך ש n=4 הוא הקטן לדוגמא את הספרה $^{\rm r}$

$$(5)_{10} = (00000000|00000000|0000000|00000101)_2 = (00000101)_2 = (0101)_2$$

בדוגמא הקודמת הזכרנו שלא יתכן n=3. הסיבה לכך היא שב- n=3 אנו נאלצים להשמיט את הביט n=3. בדוגמא הקודמת הזכרנו שלא יתכן n=3. הסיבה לנו על סימן השלם בשיטת המשלים ל-2. עבור מספרים אי שלילים ו 1 עבור שליליים.

$$(5)_{10} = (00000000|00000000|0000000|00000101)_2 = (0101)_2 \neq (101)_2$$

נתבונן כעת בייצוגים הבינארים המינימאליים של המספר 1 המספר 0 והמספר 1- בבסיס 10:

$$(1)_{10} = ({\color{red}0}1)_2 \neq ({\color{red}1})_2$$

 $(0)_{10} = (0)_2$

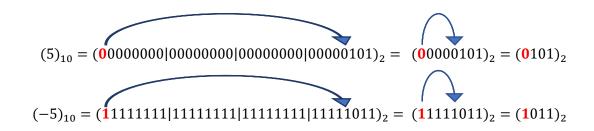
$$(-1)_{10} = (\mathbf{1}1)_2 \neq (\mathbf{1})_2$$

ניתן לראות שגם בדוגמא זו כאשר אנו מנסים לייצג את 1- בבסיס בינארי מינימאלי כך ש n=1, אנו למעשה מאבדים את הייצוג הבינארי שלו בשיטת המשלים ל-2. וכאשר אנו מנסים לייצג את 1 עם למעשה מאבדים את הייצוג הבינארי שלו בשיטת המשלים ל-1, אינו חוקי. n=1 אנחנו מייצגים ספרה ללא סימון. לכן נגדיר שהייצוג הבינארי 1,2 אינו חוקי.

היוצא דופן הוא 0. מכיוון שהוא אינו מצריך סימן, ניתן להשתמש בו כייצוג בינארי המינימאלי של 0 בבסיס 10.

ייצוג נוסף שאינו חוקי הוא מספרים בינארים עם MSB שהוא 1 וכל שאר הביטים 0. לדוגמא 1000, 1000, 10 שאינו חוקי הוא מספרים בינארים עם MSB שהוא 1 וכל שאר הביטים 0. לדוגמא ביותר 1000 חייצוגים את המספר הקטן ביותר בייצוג המשלים ל-2 עבור n קבוע, לדוגמא עבור n ווווווי 1000 מייצג את s-. אך 1000 יכול להתקבל גם מחיבור של 01 עם 0111 (7+1) שמחזיר 1000 (8-) ונקבל פתרון לא נכון. לכן נגדיר שמספרים בינאריים מסוג זה אינם חוקיים עבור ייצוג בינארי מינימאלי. ובמקום עלינו להוסיף את הMSB המתאים על מנת לייצג את המספר בבסיס 10 המתאים. לדוגמא 01000 עבור 8 ו- 11000 עבור 8-.

פרקטית, הדרך למצוא את הייצוג הבינארי המינימלי היא באמצאות העברת ה- MSB למקום שעדיין מקיים אותו ייצוג בבסיס 10.



את אורך הייצוג בינארי מינימאלי עבור מספר שלם i שונה מ-0 ניתן לחשב על ידי הנוסחא:

$$log_2(|i|) + 2$$

את התוצאה שנקבל נעגל כלפי מטה על מנת לקבל את האורך הרצוי, לדוגמא:

$$log_2(|5|) + 2 = 4.321 \rightarrow 4$$

בטבלה הבאה ניתן לראות דוגמא לשימוש בייצוג בינארי בשיטת המשלים ל-2 לצד הייצוג הבינארי המינימאלי, עבור n=4.

		I
המספר	4 ייצוג ב	ייצוג בינארי
	ביטים	מינימאלי
7	0111	0111
6	0110	0110
5	0101	0101
4	0100	0100
3	0011	011
2	0010	010
1	0001	01
0	0000	0
-1	1111	11
-2	1110	110
-3	1101	101
-4	1100	1100
-5	1011	1011
-6	1010	1010
-7	1001	1001
-8	1000	11000

<u>שיטת המשלים ל – 2</u>

שיטת המשלים ל־2 היא שיטה לייצוג מספרים עם סימן בבסיס בינארי . בשיטה זאת הסיבית הגבוהה ביותר (MSB - Most Significant Bit) מייצגת את הסימן של המספר (חיובי או שלילי) ושאר הספרות מייצגות את ערך המספר (בצורה שונה מייצוג רגיל אם הוא שלילי). שיטה זו מקובלת בתחום המחשבים כשיטה שימושית לייצוג בינארי של מספרים שעשויים להיות שליליים או חיוביים. השימוש במשלימים במחשבים נועד לפשט את פעולת החיסור, וכן לביצוע פעולות לוגיות.

השיטה מתבצעת באופן הבא, לדוגמא אם ברשותי 10 ערכים בינאריים כלשהם השיטה מתבצעת באופן הבא. x_1 , x_2 , ... באופן טבעי נבחר לייצגם עם ערכים בבסיס 10 באופן הבא:

$$x_1 \to 0$$
, $x_2 \to 1$, ... $x_8 \to 7$, $x_9 \to 8$, $x_{10} \to 9$

אך במידה ואנו מעוניינים להשתמש בייצוגים שליליים נאלץ להקצות חצי עבור ערכים שלילים באותו האופן:

$x_1 \rightarrow 0$	$x_2 \rightarrow 1$	$x_3 \rightarrow 2$
$x_4 \rightarrow 3$	$x_5 \rightarrow 4$	$x_6 \rightarrow -5$
$x_7 \rightarrow -4$	$x_8 \rightarrow -3$	$x_9 \rightarrow -2$
$x_{10} \rightarrow -1$		

הדרך לעשות זאת עם מספרים בינאריים היא על ידי שימוש ב- MSB, כי חצי מהמספרים הבינאריים הם עם MSB וחצי עם MSB.

על מנת למצוא את הייצוג הבינארי השלילי של מספר כלשהו ניתן לבצע את הצעדים הבאים:

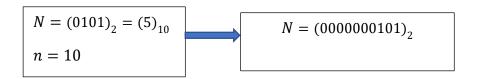
- . הפיכת כל הביטים
- ביצוע פעולת חיבור עם 1.

.-5 לדוגמא: נעבר מ- 5 ל-

באופן יותר פורמאלי, בהיתנן מספר חיובי N לפי בסיס 2, הכולל ח ספרות של ביטים, המשלים ל-2 של N מוגדר כ-:

$$f(N,n) = \begin{cases} 2^n - N & \text{if } N \neq 0 \\ 0 & \text{if } N = 0 \end{cases}$$

:לדוגמא



$$f((000000101)_{2}, 10) = (2^{10})_{10} - (0000000101)_{2}$$

$$= (10000000000 - 0000000101)_{2}$$

$$= (1111111011)_{2}$$

$$= (-5)_{10}$$

אריתמטיקה

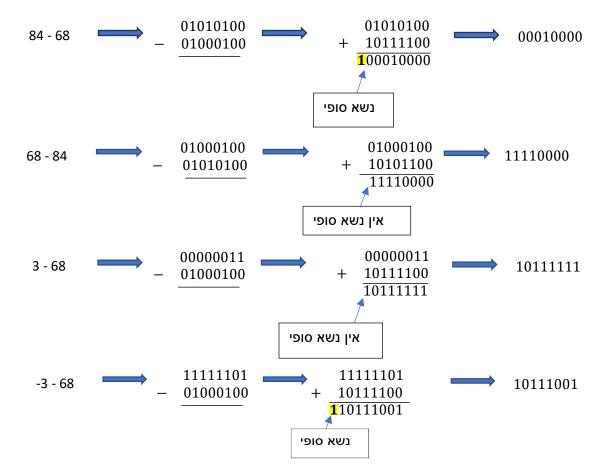
פעולות אריתמטיות של מספרים לפי בסיס 2 נעשות לפי אותם הכללים הנהוגים בפעולות המספרים בבסיס 10. כאשר מחשבים לפי בסיס 2, יש להקפיד ולהשתמש אך ורק ב-2 הספרות המותרות לשימוש, כלומר 1 ו – 0.

להלן דוגמאות לחיבור שני מספרים בינאריים מינימאלים:

פעולת החיסור בשיטת המשלים ל-2 של מספרים בינאריים מינימאלים היא מעט שונה מן השיטות המוכרות. אפשר לחסר שני מספרים חיוביים (M-N) לפי בסיס 2 כדלהלן:

- ו N אינו שווה רפדו N אינו שווה רפדו M אבעו פעולת חיבור בין M למשלים ל-2 של N אם מספר הביטים של M את הביטוי הבינארי עם מספר הביטים המועט יותר.
 - לאחר חישוב השלב הראשון:
 - אם יש נשא סופי התעלמו ממנו, והחזירו את התוצאה.
 - אם אין נשא סופי החזירו את התוצאה.

יש לציין שהתוצאה אינה תמיד מתקבלת כיצוג בינארי מינימלי.

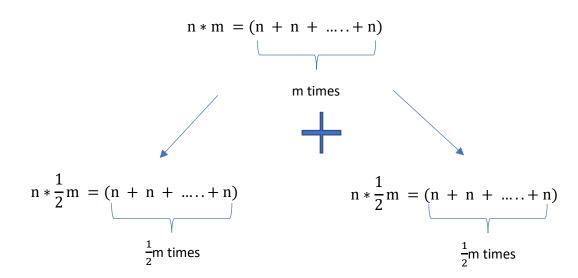


בפעולת כפל וחילוק ניתן להשתמש בכפל/ חילוק ארוך כפי שנלמד בבתי הספר היסודיים אך אם נתבונן בפעולת הכפל, באופן מופשט ניתן לפתור בעיית כפל על ידי חיבור חוזר:

$$n*m = (n + n + \dots + n)$$

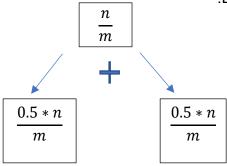
$$m \text{ times}$$

ובאופן דומה ניתן לפתור בעיית חילוק. אך הפיתרונות אינם יעילים, אבל בהחלט ניתן לייעל את הפתרון המופשט על ידי כלים שנלמדו בקורס. כלומר במקום שנבצע פעולות חיבור אחד אחרי השני בלולאה, ניתן לפצל את פעולת החיבור ולבצעם במקביל.



ובעבור חילוק $\frac{n}{m}$ נוכל להגדיל את המכנה או להקטין את המונה.

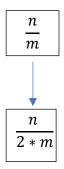
סכימת בסיס שבו הביטוי $\frac{n}{m}$ שווה ל 1 או קטן מ-1 ולבסוף סכימת (1 הקטנת המונה n בחצי עד מקרה בסיס שבו הביטוי מקרי בסיס מתאימים:



שימו לב שאם בחרתם לממש שיטה זו בעזרת השיטה (divideBy2() הנתונה לכם, תוצאת החילוק מחזירה תמיד מספר שלם. לכן עליכם למצוא דרך להתחשב בשאריות לאורך החלוקות.

שימו לב שבמימוש (*divide()* בעבודה אתם נדרשים להחזיר מספר שלם.

מי 2 עד מקרה שבו הביטוי $\frac{n}{m}$ קטן מ-1. ולאחר מכן ביצוע פעולות (2 אריטמתיות מתאימות עם מספר הפעמים שכפלנו את m כדי להחזיר תוצאה נכונה.



(נסו זאת בעצמכם עם מספרים בבסיס 10)