קורס אנליזה נומרית – עבודת בית מס' 1

שאלה 1

א.

נמיר את 0.1 לערכו בבסיס בינארי:

המספר	המספר לאחר הכפלה ב- 2	הערך השלם לאחר ההכפלה
0.1	0.2	0
0.2	0.4	0
0.4	0.8	0
0.8	1.6	1
0.6	1.2	1
0.2	0.4	0

ניתן לראות שהתהליך מחזורי כאשר מגיעים ל- 0.2 ובפרט אינסופי.

לכן מתקיים:

$$\mathbf{0.1_{10} = 0.0\overline{0011}_2}$$

לאור זאת ומהגדרת $0.\tilde{1}$ בשאלה מתקיים:

$$\mathbf{0.\tilde{1}_{10} = 0.000110011001100110011001100_2}$$

ב.

$$\Delta \mathbf{0.\tilde{1}} = \left|0.1_{10} - 0.\tilde{1}_{10}\right| = 0.00000000000000000000000\overline{0011}_2 = \mathbf{0.1_{10} * 2^{-20}}$$

$$\boldsymbol{\delta} \mathbf{0.\tilde{1}} = \frac{\Delta 0.\tilde{1}}{|0.1|} = \frac{0.1 * 2^{-20}}{0.1} = \mathbf{2^{-20}}$$

ג.

$\boldsymbol{d}$	$\frac{\mathbf{2^{1-d}}}{\mathbf{2}} = \mathbf{2^{-d}}$
0	1
1	0.5
2	0.25
3	0.125
…	…
20	$9.53674316 \times 10^{-7}$
21	$4.76837158 \times 10^{-7}$

מהנתונים בטבלה ניתן לראות שמתקיים:

$$\delta 0.\tilde{1} \leq 2^{-20}$$

וכן,

$$\delta 0.\tilde{1} > 2^{-21}$$

כלומר, ערך d המקסימלי המקיים את התנאי הינו 20.

מהגדרת ספרות משמעותיות **מתקיים שהמספר $0.\tilde{1}$ מקרב את 0.1 ל- 20 ספרות בינאריות משמעותיות**.

ד.

מההגדרות בשאלה נובע:

$$n_1 = 8 * 60 * 60 * 10 = 288{,}000$$
$$n_2 = n_1 + 20 = 288{,}020$$
$$\tilde{t_1} = 0.\tilde{1} * n_1 = 0.\tilde{1} * 288{,}000$$
$$\tilde{t_2} = 0.\tilde{1} * n_2 = 0.\tilde{1} * 288{,}020$$
$$\Delta\tilde{t} = \tilde{t_2} - \tilde{t_1} = 0.\tilde{1} * 288{,}020 - 0.\tilde{1} * 288{,}000 = 0.\tilde{1} * 20$$

מערך $\Delta\tilde{t}$ נובע:

$$\Delta t = 0.1 * 20$$

לאור כל זאת, השגיאות במדידות הזמן הן:

$$\Delta(\Delta\tilde{\boldsymbol{t}}) = |\Delta t - \Delta\tilde{t}| = 0.1 * 20 - 0.\tilde{1} * 20 = 20 * \left(0.1 - 0.\tilde{1}\right) = 20 * \Delta 0.\tilde{1}$$
$$= 20 * 0.1 * 2^{-20} = \mathbf{2^{-19}}$$
$$\boldsymbol{\delta}(\Delta\tilde{\boldsymbol{t}}) = \frac{\Delta(\Delta\tilde{t})}{|\Delta t|} = \frac{2^{-19}}{0.1 * 20} = \mathbf{2^{-20}}$$

ה.

מההגדרות בשאלה נובע:

$$n_1 = 100 * 60 * 60 * 10 = 3{,}600{,}000$$
$$n_2 = n_1 + 20 = 3{,}600{,}020$$
$$\tilde{t_1} = 0.\tilde{1} * n_1 = 0.\tilde{1} * 3{,}600{,}000$$
$$\tilde{t_2} = 0.\tilde{1} * n_2 = 0.\tilde{1} * 3{,}600{,}020$$
$$\Delta\tilde{t} = \tilde{t_2} - \tilde{t_1} = 0.\tilde{1} * 3{,}600{,}020 - 0.\tilde{1} * 3{,}600{,}000 = 0.\tilde{1} * 20$$

מערך $\Delta\tilde{t}$ נובע:

$$\Delta t = 0.1 * 20$$

נשים לב כי הערכים שקיבלנו עבור Δt ו- $\Delta\tilde{t}$ זהים לערכים שהתקבלו בסעיף ד'. הסיבה לכך היא <u>שאין תלות במספר השעות לאחר האתחול</u>, ובפרט ערכי השגיאות בסעיף זה זהה לערכי השגיאות שהתקבלו בסעיף ד'.

ו.

$$n_1 = 8 * 60 * 60 * 10 = 288{,}000$$
$$n_2 = n_1 + 20 = 288{,}020$$
$$\tilde{t_1} = 0.\tilde{1} * n_1 = 0.\tilde{1} * 288{,}000$$
$$\tilde{t_2} = 0.1 * n_2 = 0.1 * 288{,}020$$
$$\Delta\tilde{t} = \tilde{t_2} - \tilde{t_1} = 0.1 * 288{,}020 - 0.\tilde{1} * 288{,}000$$
$$= 0.1 * (288{,}000 + 20) - 0.\tilde{1} * 288{,}000$$
$$= 2 + 288{,}000 * \left(0.1 - 0.\tilde{1}\right) = 2 + \left(288{,}000 * \Delta 0.\tilde{1}\right)$$
$$= 2 + (288{,}000 * 0.1 * 2^{-20}) = 2 + 28{,}800 * 2^{-20}$$

מערך $\Delta\tilde{t}$ נובע:

$$\Delta t = 0.1 * 288{,}020 - 0.1 * 288{,}000 = 0.1 * 20 = 2$$

לאור כל זאת, השגיאות במדידות הזמן הן:

$$\Delta(\Delta\tilde{\boldsymbol{t}}) = |\Delta t - \Delta\tilde{t}| = |2 - (2 + 28{,}800 * 2^{-20})| = \mathbf{28{,}800 * 2^{-20}}$$

$$\boldsymbol{\delta}(\Delta\tilde{\boldsymbol{t}}) = \frac{\Delta(\Delta\tilde{t})}{|\Delta t|} = \frac{28{,}800 * 2^{-20}}{2} = \mathbf{28{,}800 * 2^{-21}}$$

ז.

$$n_1 = 100 * 60 * 60 * 10 = 3{,}600{,}000$$
$$n_2 = n_1 + 20 = 3{,}600{,}020$$
$$\widetilde{t_1} = 0.\tilde{1} * n_1 = 0.\tilde{1} * 3{,}600{,}000$$
$$\widetilde{t_2} = 0.1 * n_2 = 0.1 * 3{,}600{,}020$$

$$\Delta\tilde{t} = \widetilde{t_2} - \widetilde{t_1} = 0.1 * 3{,}600{,}020 - 0.\tilde{1} * 3{,}600{,}000$$
$$= 0.1 * (3{,}600{,}000 + 20) - 0.\tilde{1} * 3{,}600{,}000$$
$$= 2 + 3{,}600{,}000 * \left(0.1 - 0.\tilde{1}\right) = 2 + (3{,}600{,}000 * \Delta 0.\tilde{1}) =$$
$$= 2 + (3{,}600{,}000 * 0.1 * 2^{-20}) = 2 + 360{,}000 * 2^{-20}$$

מערך $\Delta\tilde{t}$ נובע:

$$\Delta t = 0.1 * 3{,}600{,}020 - 0.1 * 3{,}600{,}000 = 0.1 * 20 = 2$$

לאור כל זאת, השגיאות במדידות הזמן הן:

$$\Delta(\Delta\tilde{\boldsymbol{t}}) = |\Delta t - \Delta\tilde{t}| = |2 - (2 + 360{,}000 * 2^{-20})| = \mathbf{360{,}000 * 2^{-20}}$$

$$\boldsymbol{\delta}(\Delta\tilde{\boldsymbol{t}}) = \frac{\Delta(\Delta\tilde{t})}{|\Delta t|} = \frac{360{,}000 * 2^{-20}}{2} = \mathbf{360{,}000 * 2^{-21}}$$

נשים לב כי בשונה מהמקרה הקודם, כעת השגיאות <u>תלויות</u> במספר השעות לאחריהן נקרא שעון המערכת מאז אתחולה ולכן התוצאות בסעיפים ו' ז' שונות.

שאלה 2

חשב את המספר הגדול ממש מ- 0 והקטן ביותר הניתן לייצוג במקרים הבאים:

א. <u>ייצוג single</u>

a. *ייצוג נורמלי*

<u>אקספוננט</u>: בייצוג נורמלי הביטים של האקספוננט חייבים להכיל לפחות 0 אחד ולפחות 1 אחד. ובפרט, הערך המינימלי האפשרי מיוצג ע"י הביטים 00000001, שערכו לאחר החסרת $bias = 127$ הינו $\mathbf{-126}$.

<u>מנטיסה</u>: בייצוג נורמלי מופיע לפני הנקודה העשרונית ביט שערכו 1 שאינו מוצג בייצוג. ובפרט, הערך המינימלי האפשרי מיוצג ע"י 23 ביטים של 0, וערכה הכולל הינו $\mathbf{1}$.

ובסה"כ המספר הינו:

$$1 * 2^{-126} = \mathbf{2^{-126}}$$

b. *ייצוג תת-נורמלי*

<u>אקספוננט</u>: קבוע בייצוג זה ושווה $\mathbf{-126}$.

מנטיסה: על מנת לקבל מספר שערכו גדול ממש מ- 0, הביטים של המנטיסה חייבים להכיל לפחות 1 אחד. ובפרט, הערך המינימלי האפשרי מיוצג ע"י 22 ביטים של 0, כאשר הביט ה- 23 והאחרון הינו 1 וערכה הכולל הינו $\mathbf{2^{-23}}$.
ובסה"כ המספר הינו:

$$2^{-23} * 2^{-126} = \mathbf{2^{-149}}$$

ב. ייצוג double

a. *ייצוג נורמלי*

אקספוננט: בדומה להסבר בסעיף aא, ערכו לאחר החסרת $bias = 1023$ הינו $\mathbf{-1022}$.

מנטיסה: בדומה להסבר בסעיף aא, ערכה הינו $\mathbf{1}$.
ובסה"כ המספר הינו:

$$1 * 2^{-1022} = \mathbf{2^{-1022}}$$

b. *ייצוג תת-נורמלי*

אקספוננט: קבוע בייצוג זה ושווה $\mathbf{-1022}$.

מנטיסה: בדומה להסבר בסעיף aא, ערכה הכולל הינו $\mathbf{2^{-52}}$.
ובסה"כ המספר הינו:

$$2^{-52} * 2^{-1022} = \mathbf{2^{-1074}}$$

שאלה 3

נפרט את השפעת השיטה המוצגת בשאלה על הבעיות הבאות:

i. איבוד משמעות: למדנו בכיתה שבעיית איבוד משמעות עלולה להיווצר כתוצאה מחיסור שני מספרים מאוד קרובים בערכם, ובפרט חיבור של מספר חיובי ומספר שלילי שמאוד קרובים בערכם המוחלט. בצורת החישוב המקורית, לא קיימות פעולות חיסור מפורשות וכן מהעובדה שלכל x מתקיים $e^x \geq 0$ גם לא קיימת סכנה לחיבור של מספר חיובי ומספר שלילי, אזי לא נראית בה סיבה להיווצרות בעיית איבוד משמעות. בניגוד לכך, בשיטה המוצגת בשאלה, קיימות פעולות חיסור מפורשות שעבור מספרי קלט שעבורם ערכי הפונקציה e^x קרובים עלולות לגרום לבעיית איבוד משמעות.
לאור האמור לעיל, השיטה המוצגת בשאלה אינה עשויה לעזור בפתרון הבעיה הנ"ל ואף עבור קלטים מסוימים עשויה לגרום לה.

ii. רוב המספרים הממשיים אינם ניתנים לייצוג במערכת ייצוג בינארית: מגבלת הייצוג נובעת ממגבלת הזיכרון במחשב, שכן קיימים מספרים ממשיים שהייצוג שלהם בשיטת הנקודה הצפה הינו אינסופי ולכן ניתנים לייצוג בצורה מקורבת בלבד (למשל בשאלה 1 סעיף א המספר 0.1). **השיטה המוצגת בשאלה אינה נותנת מענה לשיטת הייצוג או למגבלת החומרה ולכן אינה עשויה לעזור בפתרון הבעיה הנ"ל.**

iii. סכימה של מספרים קטנים ביחד עם מספרים גדולים עלולה לגרום לשגיאות מהותיות:
בביטויים הללו קיימת פעולת סכימה במכנה בלבד ולכן ננתח אותה – פעולה זו סוכמת K איברים שתוצאתם היא הפעלה של הפונקציה e^x. הפונקציה e^x היא מונוטונית עולה, קצב עלייתה איטי מאוד עבור x קטנים (שואפת ל- 0), ובניגוד לכך קצב עלייתה מהיר מאוד (אקספוננציאלי) בחזקות חיוביות. **לאור השוני בסדרי הגודל בהתנהגות הפונקציה עבור ערכים גדולים וקטנים, השיטה המוצגת בשאלה עלולה לעזור כאשר ערכי הפונקציה בסדרת הקלט בעלי הפרשים משמעותיים, שכן לאחר פעולת החיסור המתבצעת בשיטה החדשה נקבל מספרים קטנים הקרובים יחסית בערכם.**

iv. <u>underflow</u>: בעיה זו נוצרת כאשר תוצאת חישוב מובילה למספר קטן מאוד בערך מוחלט שלא ניתן לייצוג במערכת. פונקציית האקספוננט מונוטונית עולה, ולכן ערך המכנה בביטוי המקורי גדול מערך המכנה בביטוי החדש. בנוסף, מתכונת השברים, שאומרת שככל שהמכנה גדול יותר השבר כולו קטן יותר, נובע שערך הביטוי המקורי קטן מערך הביטוי החדש. **ערך הביטוי החדש גדול יותר ובפרט רחוק יותר מהאפס בערך מוחלט ולכן שיטה זו עשויה לעזור בפתרון הבעיה הנ"ל.**

שאלה 4

א.

$$Z = e^{\alpha(X-Y)}$$
$$\nabla Z = \left(\alpha e^{\alpha(X-Y)}, -\alpha e^{\alpha(X-Y)}\right) = (\alpha Z, -\alpha Z)$$
$$\Delta \boldsymbol{Z} \approx \nabla Z|*|(\Delta X, \Delta Y) = |\alpha Z|\Delta X + |\alpha Z|\Delta Y = |\boldsymbol{\alpha}|\boldsymbol{Z}(\Delta \boldsymbol{X} + \Delta \boldsymbol{Y})$$

- הסימן $|*|$ מייצג את הסימן שהוגדר בכיתה המסמל את הפעולה בה קואורדינטות הגרדיאנט מופיעות בערך מוחלט.
- השוויון האחרון נובע מכך שהפונקציה Z חיובית לכל X, Y.

$$\delta Z = \frac{\Delta Z}{|Z|} \approx \frac{|\alpha| Z(\Delta X + \Delta Y)}{Z} = |\alpha|(\Delta X + \Delta Y)$$

ב. נציב $\Delta X = \Delta Y = 2$

$$\delta Z \approx |\alpha|(\Delta X + \Delta Y) = 4|\alpha|$$

מתקיים

$$4|\alpha| < 0.05$$

עבור

$$\boldsymbol{-0.0125 < \alpha < 0.0125}$$

ג. הייצוג $Z = \frac{e^{\alpha X}}{e^{\alpha Y}}$ בעל משמעות מתמטית שקולה לייצוג בסעיף א'. <u>בביצוע פעולות אריתמטיות לחישוב השגיאה היחסית</u>, בדומה לאלה שביצענו בסעיף א', <u>נקבל שהשגיאה היחסית של שני הייצוגים שווה.</u> אך יחד עם זאת, קיימים מספר הבדלים בין הייצוגים:

א. בייצוג $Z = \frac{e^{\alpha X}}{e^{\alpha Y}}$ מתבצעת פעולת חילוק במקום פעולת חיסור. למדנו בכיתה כי חיסור מספרים הקרובים בערכם עלול לגרום לעלייה חדה בשגיאה היחסית של הפלט - תופעת איבוד משמעות. לכן, <u>עבור X, Y קרובים בערכם, בייצוג בסעיף א' עלולה להיווצר תופעת איבוד משמעות בעת חישוב השגיאה היחסית</u>.

ב. בייצוג $Z = \frac{e^{\alpha X}}{e^{\alpha Y}}$ מתבצעות שתי הפעלות של פונקציית האקספוננט ובייצוג המקורי הפעלה אחת. <u>הבדל זה בא לידי ביטוי ביעילות זמן החישוב וכן עלול לגרום לעלייה בשגיאה היחסית</u> - שכן תוצאת פונקציית האקספוננט הינה מספר ממשי שאינו בהכרח בר ייצוג.

לאור האמור לעיל, לשתי השיטות יתרונות וחסרונות בעלי השפעה על השגיאה היחסית, ולאן לא קיימת שיטה העדיפה באופן חד משמעי על השנייה.

שאלה 5

```python
from math import floor


def adder(man_a, exp_a, man_b, exp_b):
    sum_man = man_a + man_b * (10 ** (exp_b - exp_a))
    sum_exp = exp_a
    while sum_man > 1:  # normalization
        sum_man /= 10
        sum_exp += 1
    sum_man = floor(sum_man * 1000) / 1000  # cutting of 3 digits after the decimal point
    return sum_man, sum_exp


def tar7_ass1(n):
    step_man = 0.1
    step_exp = -2
    res_man = 0.1
    res_exp = -2

    for i in range(2, n):
        res_man, res_exp = adder(res_man, res_exp, step_man, step_exp)

    real_res = 0.001 * n
    approximate_res = res_man * (10 ** res_exp)
    err = abs(real_res - approximate_res)
    return err
```

א.

לאחר 80 איטרציות השגיאה המתקבלת היא : **0.0010999999999999899**

לאחר 8000 איטרציות השגיאה המתקבלת היא : **7**

שגיאת הצובר גדלה ככל שמספר האיטרציות גדל, בעקבות ה"צבירה" של השגיאה אשר קורית כיוון שהמספר אותו אנו סוכמים, 0.001, אינו בר ייצוג (דורש ייצוג אינסופי של ביטים).

ב.

ההפרש בין השגיאה אחרי 80 איטרציות לבין השגיאה אחרי 82 איטרציות הינו : **0**

ההפרש בין השגיאה אחרי 8000 איטרציות לבין השגיאה אחרי 8002 איטרציות הינו : **0.002000000000000668**

שני ההפרשים הנ"ל כה שונים זה מזה משום ששגיאת הצובר, המצטברת בכל איטרציה כפי שהסברנו בסעיף א, אינה גדלה בקצב ליניארי – אלא בסדר גודל גדול מאוד.