האוניברסיטה העברית בירושלים

בית הספר להנדסה ולמדעי המחשב ע"ש רחל וסלים בנין

סדנאות תכנות בשפת C ו־C++ – קיץ (קורס 67320)
C – תרגיל 3

תאריך ההגשה של התרגיל והבוחן התיאורטי: יום שלישי, ה־20 באוגוסט, 2019 – עד השעה 23:55;

הגשה מאוחרת (בהפחתת 10 נקודות): יום רביעי, ה־21 באוגוסט, 2019 – עד השעה 23:55.

נושאי התרגיל: מצביעים לפונקציות, ספריות (יצירת ספריה סטטית ודינמית).

1 רקע

בתרגיל זה נחבר, ולאחר מכן גם נעשה שימוש, בספריה המאפשרת לחשב שני ערכים חשובים ביותר בחשבון האינפיניטסימלי – אינטגרל ונגזרת.

2 הגדרות

תחילה, לשם הנוחות, נזכיר להלן מספר הגדרות ומשפטים שהוצגו והוכחו במהלך הקורסים חשבון אינפיטסימלי (1) וחשבון אינפיניטסימלי (2) (או חשבון אינפיניטסימלי (2) לתלמידי מדעי המחשב).

2.1 הנגזרת של פונקציה ממשית

כזכור, הנגזרת של פונקציה ממשית מתארת את קצב ההשתנות של הפונקציה. בפן הגיאומטרי, הנגזרת של פונקציה בנקודה שווה לשיפוע המשיק באותה נקודה.

הגדרה: תהי f פונקציה המוגדרת בסביבה מלאה של $x_0 \in \mathbb{R}$. נאמר ש־f גזירה בנקודה x_0 אם ורק אם קיים הגבול במובן הצר:

$$\lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0}$$

במקרה הזה נקרא לגבול שלעיל "הנגזרת של f בנקודה x_0" ונסמנו $f'(x)$.
הגדרה שקולה (שהוכחה באינפי 1): תהי f פונקציה המוגדרת בסביבה מלאה של $x_0 \in \mathbb{R}$.
נאמר ש־f גזירה בנקודה x_0 אם ורק אם קיים הגבול במובן הצר:

$$\lim_{h \to 0} \frac{f(x_0+h)-f(x_0)}{h}$$

במקרה זה יתקיים:

$$f'(x) = \lim_{h \to 0} \frac{f(x_0+h)-f(x_0)}{h}$$

הגבול הנ"ל מייצג את שיפוע הישר המחבר את הנקודה $(x_0,\, f(x_0))$ שעל גרף הפונקציה עם הנקודה הקרובה לה $(x_0+h,\, f(x_0+h))$ – ומכאן שהנגזרת מתארת את שיפוע המשיק בנקודה x_0.

2.2 האינטגרל המסויים לפי ריימן (Riemann)

כידוע, האינטגרל המסויים של פונקציה תחת הקטע מסויים מייצג ערך השווה לשטח הכלוא בין ציר ה־X לבין גרף הפונקציה – ובין קצות הקטע. ערך זה מסומן $\int_a^b f(x)dx$.
אחת הדרכים המוקדמות יותר מבחינה היסטורית להערכת אינטגרלים הייתה באמצעות **סכומי ריימן**. סכום רימן הוא סוג מסוים של קירוב האינטגרל של פונקציה על ידי סכימה. הערכת האינטגרל בגישה זו מבוססת על **סכומי רימן**, שיטה הקרויה על שם המתמטיקאי שהגה אותה במאה התשע־עשרה – ברנהרד ריימן. למרות שהשימוש העיקרי שלה הוא לשיערוך אינטגרלים, ניתן לעשות שימוש בסכומי ריימן גם לשיערוך כימורים וערכים נוספים.
הגדרה – סכומי ריימן: תהי $f : [a,\, b] \to \mathbb{R}$ פונקציה תחת הקטע $[a,\, b] \subset \mathbb{R}$ ותהי $P = \{a = x_0,\, \ldots,\, x_n = b\}$ חלוקה של הקטע $[a,\, b]$ המקיימת $a = x_0 < x_1 < \cdots < x_n = b$.
נבחר נקודות עבור החלוקה P כך שנקבל סדרה $\xi = (\xi_k)_{k=1}^n$ כך ש־$\xi \in [x_{k-1},\, x_k]$. **סכום ריימן** (Riemann sum) של f ביחס לחלוקה P ובחירת הנקודות ξ הוא:

$$\sigma(f,\, P,\, \xi) = \sum_{i=1}^{n} f(\xi_i) \cdot \Delta x_i$$

כאשר $\Delta x_i = x_i - x_{i-1}$.
במובן הגיאומטרי, ניתן לפרש את הגדרה זו כניסיון לשערך את השטח מתחת לגרף הפונקציה f על ידי יצירת מלבנים שבסיסם $[x_{i-1},\, x_i]$ וגובהם $f(\xi_i)$. במקרה זה, סכום ריימן, שסומן $\sigma(f,\, P,\, \xi)$, הוא סכום המלבנים.
הגדרה – האינטגרל המסויים לפי ריימן: תהי $f : [a,\, b] \to \mathbb{R}$ פונקציה תחת הקטע $[a,\, b] \subset \mathbb{R}$. נאמר ש־$f$ **אינטגרבילית לפי ריימן** אם קיים מספר τ כך שלכל $\varepsilon > 0$ קיים $\delta > 0$ כך שלכל חלוקה P של הקטע $[a,\, b]$ עם $\lambda(P) < \delta$ ולכל בחירה של ξ עבור P מתקיים:

$$\sigma(f,\, P,\, \xi) - \tau| < \varepsilon$$

במקרה זה, τ נקרא **אינטגרל ריימן** של f בקטע $[a,\, b]$.

3 ספריית "libinfi"

בחלק הראשון של תרגיל זה נכתוב את הספריה "libinfi", שמטרתה לבצע חישוב נומרי של ערך האינטגרל המסויים והנגזרת של פונקציה כלשהי.

בספריה יהיו אך ורק שני קבצים - קובץ ההצהרות, infi.h. שאותו הגדרנו עוברכם ואינכם רשאים לערוך (**<u>ואסור להגישו</u>**); קובץ המימוש, infi.c. שם תממשו את הפונקציות שהוגדרו בקובץ ה־h (תוכלו להגדיר, כמובן, פונקציות עזר כראות עיניכם, אך שאלו לא יהיו זמינות מחוץ לספריה). **<u>את קובץ זה עליכם להגיש</u>**.

3.1 הגדרת הקובץ "infi.h"

בספריה "libinfi" יהיה קובץ ההצהרות שאותו הגדרנו עוברכם - infi.h. קובץ זה יכיל את ההצהרות לפונקציות הבאות:

- <u>אינטגרציה נומרית:</u> הגדירו וממשו פונקציה בשם "integration" המקבלת את הפרמטרים הבאים:

  1. מצביע לפונקציה עליה מתבצעת האינטגרציה ($f:\mathbb{R}\to\mathbb{R}$).
  2. נקודת ההתחלה של האינטגרציה (כלומר הערך $a\in\mathbb{R}$ בקטע קטע $[a,\,b]$).
  3. נקודת הסיום של האינטגרציה (כלומר הערך $b\in\mathbb{R}$ בקטע קטע $[a,\,b]$).
  4. מספר $n\in\mathbb{N}$ שמסמל את כמות קטעי החלוקה (ראו את ההסבר על שיערוך ערך האינטגרל באמצעות סכומי רימן בהמשך).

  הפונקציה תחזיר מספר המסמן את תוצאת האינטגרציה. חתימת הפונקציה תהא:

```
double integration(double (*)(double), double, double, int);
```

- <u>דיפרנציאל נומרי:</u> הגדירו וממשו פונקציה בשם "derivative" המקבלת את הפרמטרים הבאים:

  1. מצביע לפונקציה עליה מבוצעת הגזירה($f:\mathbb{R}\to\mathbb{R}$).
  2. הנקודה שבה מתבצעת הגזירה (כלומר $x_0\in\mathbb{R}$).
  3. ערך $h\in\mathbb{R}$, המסמל את גודל הצעד (ראו להלן את ההסבר על שיערוך הנגזרת).

  הפונקציה תחזיר את מספר המסמן את תוצאת הגזירה. חתימת הפונקציה תהא:

```
double derivative(double (*)(double), double, double);
```

הערות כלליות:

- **לא ניתן** להניח שהפרמטרים תקינים. עליכם לוודא את תקינותם **באמצעות פקודות** assert (זכרו "לייבא" את assert מ־assert.h).
- עליכם לייצג מספרים שאינם שלמים (כלומר מ־$\mathbb{R}$) באמצעות double.
- המצביע לפונקציה, שאותו יש לשלוח לשני הפונקציות שתיצרו, ייצג פונקציה שמקבלת מספר double ומחזירה מספר double.

- **<u>הערה חשובה מאוד – שימו ♡:</u>** יכול ויהיה **מעט** שוני בין פתרונכם לבין פתרון בית הספר, וכן בין הרצת פתרונכם על מחשבכם האישי ועל מחשבי בית הספר.[1] לפיכך, אנא נהגו לפי ההנחיות (המחייבות) שלהלן:

1. זכרו שהתוצאה המחייבת היא זו שמתקבלת מהרצת התרגיל במחשבי בית הספר.
2. כדי לוודא שנעשית בדיקה שוויונית המתעלמת מרכיב השגיאה, פתרונכם יבדק רק ביחס למספר ספרות ראשונות לאחר הנקודה העשרונית. כמות הספרות אליהם נתייחס בבדיקת התרגיל יגזרו מערך ה־n שישלח לפעולת האינטגרציה (ככול שהערך גדול יותר נצפה לתוצאה מדוייקת יותר) וערך ה־h שישלח לפעולת הדיפרנציאל (ככול שהערך קטן יותר נצפה לתוצאה מדוייקת יותר).
על כל פנים, זכרו כי מדובר בנושאים ידועים ומוכרים שנלקחו בחשבון בעת חיבור התרגיל והבדיקות האוטומטיות – כך שאין ממה לחשוש. תדעו כי פתרונכם נכון אם זה יגיע לתוצאה מספרית קרובה מאוד לתוצאה אליהם תגיעו מחישוב מתמטי עם המשפטים שלמדתם באינפי' 1 ו־2 ואל הערך שאליו מגיע פתרון בית הספר.

3.2 שיערוך אינטגרציה נומרית באמצעות סכומי רימן

ראינו לעיל כיצד ניתן לבצע אינטגרציה על פונקציה מסויימת באמצעות סכומי רימן.
בתרגיל זה נעשה שימוש באינטגרציה נומרית לאינטגרל מדרגה 1 (Numerical Integration, פעולה זו גם לפעמים נקראת Numerical Quadrature) בכדי לשערך את הערך הכלוא תחת פונקציה רציונלית מסויימת.
כדי לעשות זאת נשתמש בסכומי רימן ־ ובפרט בכלל נקודת האמצע (Midpoint Rule):
נגדיר P חלוקה **שווה** של הקטע $[a,\, b]$ כך ש־$\Delta x = \frac{b-a}{n}$ ונקבל:

$$P = (a,\, a + \Delta x,\, a + 2 \cdot \Delta x, \ldots, a + (n-1) \cdot \Delta x,\, b)$$

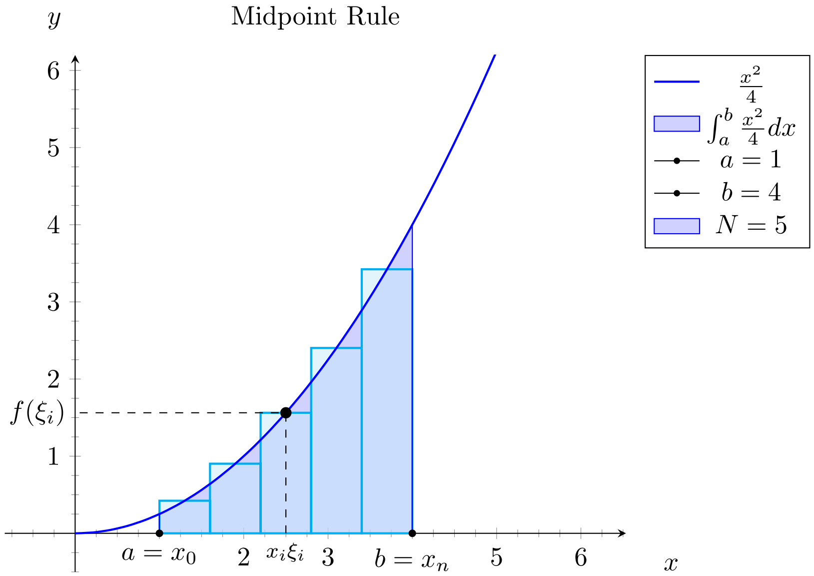
עתה עבור כל קטע, נגדיר את ξ_i כנקודת האמצע של הקטע ה־i. ברור ש־$\xi_i \in [x_{i-1},\, x_i]$.
אם כך, חישוב סכומי רימן יוביל לסכימת השטח הכלוא תחת n מלבנים כך שרוחבו של כל מלבן הוא ξ_i ואורכו $f(\xi_i)$.
אם כך, ערכו **המשוערך** של האינטגרל המסוים של f בקטע $[a,\, b]$ יהא:

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} f\left(\xi_i\right) \cdot \Delta x$$

דוגמה: תהי הפונקציה $f : [0, \infty) \to \mathbb{R},\, f(x) = \frac{x^2}{4}$. נרצה לשערך את הסכום הכלוא תחת הפונקציה $f(x)$ בקטע $[1,\, 4]$.
תהי P חלוקה של הקטע $[a,\, b]$ כך ש־Δx ו־ξ_i מוגדרות כפי שהוסבר לעיל עבור כל $1 \le i \le n$. אם כך, אם נניח ש־$a = 1,\, b = 4$ ו־$n = 5$ אזי נקבל את החלוקה הבאה:

---

[1] תופעה זו מוכרת באנליזה נומרית ונוגעת לעובדה שמדובר בשיערוך בלבד. ניתן להעריך את גודל "השגיאה" באמצעות פיתוח פולינום טיילור, **אך נושא זה אינו נכלל בתרגיל זה.** בנוסף לכך, תופעה זו עלולה להיגרם גם בעקבות תופעה שנקראת floating point error ונוגעת לרמת השגיאה של מספרי `double` (ו־float) מול אחסונם בזיכרון המחשב



חישוב סכום הרימן כשיערוך של האינטגרל יוביל, אפוא, לתוצאה:

$$\int_a^b f(x)dx = \int_1^4 \frac{x^2}{4} \approx \sum_{i=1}^n f(\xi_i) \cdot \Delta x = \Delta x \cdot \sum_{i=1}^n f(\xi_i)$$
$$= \frac{3}{5}\left(\frac{169}{400} + \frac{361}{400} + 1\frac{9}{16} + 2\frac{161}{400} + 3\frac{169}{400}\right) = 5.2275$$

קיבלנו ששיערוך האינטגרל הוא 5.2275. עתה, כדי לבחון עד כמה מדוייקת התוצאה שקיבלנו, נוכל לחשב את ערכו המדוייק של האינטגרל המסויים של f בקטע $[a, b]$ בעזרת המשפט היסודי של החדוו"א:

$$\int_a^b f(x)dx = \int_1^4 \frac{x^2}{4}dx = \left.\frac{x^3}{12}\right|_1^4 = \frac{4^3}{12} - \frac{1^3}{12} = \frac{21}{4} = 5.25$$

אם כך, השיערוך שביצענו עם סכומי רימן היה מדוייק עד לערך השני לאחר הנקודה העשרונית. אם היינו רוצים לקבל ערך מדוייק יותר, היה עלינו לבחור חלוקה רחבה יותר (כלומר ערך n גדול יותר). לסיום, כדאי לדעת שניתן לחשב חסם מלעיל לערך השגיאה, על מנת לבחור חלוקה שתספק אותנו מבחינת דיוק התוצאה – אך חישוב זה אינו חלק מהתרגיל.

3.3 שיערוך הנגזרת הנומרית

גם במקרה זה לא ננסה לחשב את הערך המדויק של הנגזרת, המתקבל כאשר $h \to 0$, אלא ננסה לשערכה. לשם כך, נזכור תחילה שהשיפוע של הקו שונה ממדרון המשיק בסכום פרופורציונלי ביחס ל־h. כלומר, כאשר h מתקרב לאפס, השיפוע של הקו הסמוך מתקרב

למורד הקו הממשי. לכן, השיטה הפשוטה ביותר לחישוב נגזרת נומרית היא להשתמש בהגדרת הנגזרת כאשר בוחרים $h > 0$ קטן מספיק. כלומר:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}, \quad h > 0$$

שיטת שיערוך זו נקראת Forward differencing והיא מחשבת את ערך הנגזרת ב־$f(x_0+h)$. עם זאת, לצורך תרגיל זה אנו נבחר אחרת, אך דומה. נגדיר את הנגזרת הנומרית כך:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}, \quad h > 0$$

שיטת שיערוך זו נקראת Centered differencing והיא בוחנת את ערך הנגזרת המשוער ביחס ל־$f(x_0+h)$ ו־$f(x_0-h)$. הסיבה שבחרנו בשיטה זו היא כי בתנאים מסוימים היא יותר מדויקת (למתעניינים – ניתן לחשב את השגיאה באמצעות פולינום טיילור).

דוגמה: תהי הפונקציה $f : [0,\infty) \to \mathbb{R},\ f(x) = \frac{x^2}{4}$. נרצה לשערך את ערך הנגזרת של f בסביבה מלאה של $x_0 = 5$. אם נבחר $h = 0.1$ נקבל:

$$f'(x) \approx \frac{\frac{(x+h)^2}{4} - \frac{(x-h)^2}{4}}{2h} = \frac{\frac{(x+0.1)^2}{4} - \frac{(x-0.1)^2}{4}}{0.2} = 2.5$$

קיבלנו, אם כן, שערך הנגזרת של f ב־$x_0 = 5$ לשיעורין הוא 2.5. עתה, כדי להיווכח האם התוצאה שקיבלנו נכונה – ואם כן, עד כמה היא מדוייקת – נגזור את f לפי הכללים "הרגילים" שנלמדו באינפי' 1:

$$f'(x) = \left(\frac{x^2}{4}\right)' = \frac{x}{2} \Rightarrow f'(5) = \frac{5}{2} = 2.5$$

והנה, התוצאה הנומרית ששיערכנו **זהה**, במקרה זה, לתוצאה המדוייקת. גם במקרה זה, כמובן, הדיוק של השיערוך היה תלוי ב־h (כפי שבאינטגרציה הוא היה תלוי ב־N).

לסיום, נציין שפיתרון בית הספר, למשל, מוביל לתוצאה $f'(5) \approx 2.4999999999999467$. הסיבה לכך, במקרה זה, אינה נעוצה בשגיאת החישוב (שמבוטאת על ידי פולינום טיילור – שהרי ראינו כי במקרה זה החישוב הוליד בדיוק 2.5), אלא בדרך בה המחשב שומר מספרי float בזיכרון. ערכי שגיאה הללו, כאמור, נקלחו בחשבון בעת חיבור הבדיקות האוטומטיות.

4 התוכנית NumericalAnalyzer

בחלק השני של התרגיל, נכתוב בקובץ `NumericalAnalyzer.c` תוכנית פשוטה **העושה שימוש בספריה שכתבתם** ומחשבת את ערך האינטגרציה וערך הנגזרת למספר פונקציות שהוגדרו מראש בהמשך התרגיל.

4.1 קלט

התוכנית תקבל את מלוא הקלטים שלה דרך כארגומנטים ב־cli.

- הארגומנט הראשון יהא מספר הפונקציה עליה נרצה לבצע אינטגרציה או נגזרת (בהתאם להקשר), מתוך רשימת הפונקציות בהן עליכם לתמוך. רשימת הפונקציות כאמור מופיעה בהמשך התרגיל.
- ארגומנטים 2 – 4 ישמשו לפעולת האינטגרציה, כדלקמן:
  - הארגומנט השני והארגומנט השלישי שתקבל התוכנית יסמנו את ערכי הקטע $[a, b]$ בהתאמה.
  - הארגומנט הרביעי יסמן את ערך החלוקה (n).

  ארגומנטים 5 ־ 6 ישמשו לפעולת הנגזרת, כדלקמן:
  - הארגומנט החמישי מסמן נקודת הגזירה, כלומר את x_0.
  - הארגומנט השישי מסמן את האפסילון, כלומר את h.

הערות:

1. **לא ניתן** להניח שהקלט תקין בשום מובן שהוא. למשל, לא ניתן להניח שהתקבלו מספר ארגומנטים תקין או שהתקבלו מספרים תקינים. אם התקבל ארגומנט שאינו תקין, עליכם להדפיס ל־`stderr` את הפלט:

   ```
   Invalid input\n
   ```

   כש־"`\n`" מסמן ירידת שורה. לאחר הדפסת הפלט, עליכם לסגור באופן מיידי את התוכנית עם קוד סיום `EXIT_FAILURE`.

2. **כל הארגומנטים** שתוארו לעיל הם חובה. אם לא נשלחו ארגומנטים כלל, או שכמות הארגומנטים שגויה, עליכם להדפיס אל ה־`stderr` את הודעת Usage הבאה:

   ```
   Usage: NumericalAnalyzer <function number> <a> <b> <N> <x0> <h>\n
   ```

   (ירידת השורה שבאמצע הודעת השגיאה **לא אמורה להופיע** בפתרונכם; עליכם לרדת שורה רק בסוף ההודעה; מנגד, התו `\n` אכן מסמן את תו ירידת שורה). לאחר מכן, עליכם לסגור באופן מיידי את התוכנית עם קוד סיום `EXIT_FAILURE`.

4.2 רשימת פונקציות

על התוכנית לתמוך בפונקציות הבאות (עבור כל f_i שבטבלה, i הוא מספר הפונקציה שיש להזין כארגומנט ו־f_i זו הגדרתה):

$f_1(x) = \frac{x^2}{4}$	$f_2(x) = \frac{-x^3+3\cdot x^2+x-4\cdot\sqrt{x}}{2\cdot x\cdot\sqrt{x}}$	$f_3(x) = sin^2x - cos^2x$
$f_4(x) = \frac{sin(x)}{1+cos(2x)}$	$f_5(x) = e^x$	$f_6(x) = \frac{sinh(2x)}{e\cdot x^2}$

תזכורת: תוכלו לגשת לפונקציות כדוגמת sin ו־cos, או לערכים כדוגמת π, בעזרת `math.h`.

4.3 פלט

על התוכנית להדפיס את ערכי האינטגרל והדיפרנציאל של הפונקציה שנבחרה ועם הערכים שנשלחו ב־cli, ולסיים את ריצתה עם EXIT_SUCCESS, כך:

1. בשורת הפלט הראשונה יודפס ערך האינטגרל לפי הפורמט הבא:

```
Integral: {...}
```

1. בשורת הפלט הראשונה יודפס ערך הדיפרנציאל לפי הפורמט הבא:

```
Derivative: {...}
```

כאשר בשני המקרים "\n" מסמן ירידת שורה ו־`{...}` מסמן את התוצאה. את המספרים יש להדפיס בפורמט "`%0.5f`".

5 הידור התוכנית והספריה באמצעות קובץ Makefile

בתרגיל זה הבדיקה האוטומטית לא תהדר באופן אוטומטי את פתרונכם, **ועליכם לספק קובץ Makefile** המגדיר כיצד להדר את התרגיל. על הקובץ לתמוך בתכונות הבאות:

- פקודת ברירת מחדל שמייצרת את הספריה (בשם "libinfi") ואת התוכנית לדוגמה (בשם "NumericalAnalyzer").
- פקודת clean (כלומר, PHONY) שמסירה כל קובץ שהפקודה הדיפולטיבית יצרה.

הנחיות ודגשים לחיבור קובץ ה־Makefile:

- הספריה "`libinfi`" תהא ספריה **סטטית.**
- "`libinfi`" תכלול **את** "`infi.h`" ו־"`infi.c`" **בלבד.**

6 פתרון בית הספר

בתרגיל זה נציע שתי דרכים לבחון את התנהגותו של פתרון בית הספר:

- תכלו לעשות שימוש בספריה "libinfi" שנכתבה על ידי בית הספר ולקמפלה כנגד תוכניות לדוגמה שתכתבו (זוהי הדרך המומלצת).
- בנתיב המכיל את פתרון בית הספר זמינות שתי תוכניות לדוגמה שעושות שימוש בספריה על מנת לחשב מספר פונקציות לדוגמה, בהתאם לקלט המשתמש.

7 דוגמה

נניח שנרצה להריץ את התוכנית עם הפרמטרים הבאים (i מסמן את מספר הפונקציה):

$$i = 1 \Rightarrow f(x) = f_1(x) = \frac{x^2}{4},\, a = 1,\, b = 4,\, n = 5,\, x_0 = 5,\, h = 0.1$$

נוכל להריץ את התוכנית כך:

```
$ ./NumericalAnalyzer 1 1 4 5 5 0.1
Integral: 5.22750
Derivative: 2.50000
```

(כאשר השורה שנפתחת ב־$ מסמנת את הפקודה שהוקלדה).

8 נהלי הגשה

- קראו בקפידה את הוראות תרגיל זה ואת ההנחיות להגשת תרגילים שבאתר הקורס.
- כתבו את כל ההודעות שבהוראות התרגיל בעצמכם. העתקת ההודעות מהקובץ עלולה להוסיף תווים מיותרים ולפגוע בבדיקה האוטומטית, המנקדת את עבודתכם.
- כאמור בנחיות הכלליות להגשת תרגילים, גם בתרגיל זה חל איסור על שימוש ב־VLA (מנגד, וודאי שאתם רשאים להשתמש בזיכרון דינמי).
- **נזכיר:** כאמור בהנחיות הכלליות להגשת תרגילים ־ הקצאת זיכרון דינמית מחייבת לאמת שמערכת ההפעלה הקצתה אותו וכן מחייבת את שחרור הזיכרון, למעט במקרים בהם ישנה שגיאה המחייבת סגירת התוכנית באופן מיידי עם קוד שגיאה (כלומר קוד יציאה השונה מ־0). תוכלו להיעזר בתוכנה valgrind כדי לחפש דליפות זיכרון בתוכנית שכתבתם.
- פתרון בית הספר זמין בנתיב (שימו לב שבתיקיה הנ"ל תוכלו למצוא גם את libinfi ולנסות להדר תוכניות לדוגמה כנגדו):

  ~proglab/www/c_ex3/NumericalAnalyzer

- עליכם ליצור קובץ tar הכולל **אך ורק** את הקבצים infi.h, infi.c, NumericalAnalyzer.c, Makefile, README. ניתן ליצור קובץ tar כדרוש על ידי הפקודה:

  ```
  $ tar -cvf ex3.tar infi.h infi.c NumericalAnalyzer.c README
  Makefile
  ```

  שימו לב: קבצי קוד המקור שתכתבו נדרשים להתקמפל כהלכה עם C99, כנדרש בהוראות להגשת תרגילים שפורסמו באתר הקורס.
- אנא וודאו כי התרגיל שלכם עובר את ה־Pre-submission Script **ללא שגיאות או אזהרות**. קובץ ה־Pre-submission Script זמין בנתיב.

  ~proglab/www/c_ex3/presubmission

בהצלחה!!