

תרגיל מס' 3 – אופרטורים

הוראות הגשה

יש לשלוח את כל קבצי המקור (cpp, h)

- בתרגיל זה, כל הקלטים מהמקלדת והפלט למסך. אין להדפיס כללהוראות למשתמש, בקשות ושאלות, כגון "הכנס מספר" וכו'.
- יש לקרוא את התרגיל היטב לפני שמתחילים לעבוד, ולוודא שהבנתם את כל הפרטים.
- סוף התרגיל ישנן הערות חשובות הנוגעות להבנת ולכתיבת התרגיל, יש לקרוא אותן היטב לפני תחילת העבודה.
- ניתן להגיש בזוגות
- בכל פניה למתרגלים, נא לציין את שם הקורס ושם התלמיד!

רקע

בתרגיל זה עליכם לממש פלינדרום.

הגדרה:

פלינדרום הוא מילה, מספר, משפט או כל רצף סמלים אחר שניתן לקרוא משני הכיוונים, משמאל לימין ומימין לשמאל ללא שינוי בתוצאה. מקור המילה הוא יוני פֶּלִין - שוב, הפוך (ו) דרומוס – כיוון.

השם "נתן" למשל הוא שם פלינדרומי, וכן המילים זוז, סוס. המילים הפלינדרומיות הארוכות ביותר בשפה העברית שיש להן משמעות הן "ולכשתשכלו", "מכיתותיכם". המילה הפלינדרומית הארוכה ביותר בלועזית הנכתבת בעברית היא טריומוריוט (שלטון של שלושה אנשים ברומא העתיקה).
לפלינדרומים אין ערך לשוני, והם נועדו בעיקר לשעשועי לשון, ומהווים נושא לחידונים וחידודי לשון.

לצורך התכנית עליכם להגדיר את המחלקות הבאות (אלו עיקרי הדרישות, עליכם להוסיף לפי ראות עיניכם לפי בנייתכם):

(כמובן שלכל המחלקות צריך להוסיף Constructor, Destructor, Copy constructor, אופרטור השמה ומתודות עזר לפי הצורך).

מחלקת Palindrome

כתבו מחלקה בשם Palindrome המייצגת את מחרוזות הפלינדרום. המחלקה מכילה כ- data member את האיברים הבאים (לפחות):

○ מצביע למחרוזת תווים (בגודל כלשהו) שמהווים חצי מהפלינדרום. המחרוזת תכיל אך ורק את התווים 'a'-'z', 'A'-'Z' ו-'0'-'9'.

לדוגמא אם הפלינדרום הוא "abcba" אזי במחרוזת הנ"ל תישמר המחרוזת "abc".

○ size – משתנה שמחזיק את גודל הפלינדרום האמיתי (ולא רק את מה ששמור במחרוזת) לדוגמא אם הפלינדרום הוא "abcba" גודל הפלינדרום הוא 5.

שימו לב כי הפלינדרום עם המחרוזת "abc" בגודל 5 ("abcba") אינו זהה לפלינדרום עם המחרוזת "abc" בגודל 6 ("abccba")!

במחלקה יש להגדיר את המתודות הבאות:

פונקציות קונסטרוקטור ודסטרוקטור מתאימות (כולל Copy Constructor).

אופרטור += כאופרטור שירשור:

האופרטור מקבל כארגומנט אובייקט של Palindrome ומשנה את האובייקט שהפעיל את המתודה להיות השירשור של שתי המחרוזות מהאובייקטים שהתקבלו. לדוגמא, נניח שני אובייקטים x, y. אובייקט x מייצג את הפלינדרום "9339" ואובייקט y מייצג את הפלינדרום "abba". הפעלת $x+=y$ משנה את x כך שמעכשו הוא מייצג את הפלינדרום: "93abba93". אובייקט y לא משתנה. כמו כן שימו לב כי הפרמטר הזה מושפע מסדר האובייקטים, למשל $y+=x$ ייתן תוצאה שונה, כי y יכול את "ab9339ba".

שימו לב כי אם מספר התווים בפלינדרום הראשון הוא אי-זוגי ניאליץ להגדיל את הפלינדרום על-ידי הכפלת התו האמצעי, דוגמא:

$x = "abCba"$, $y = "232"$

לאחר ביצוע $x+=y$, נקבל $x = "abC232Cba"$

במידה ואחד הפלינדרומים ריק, אזי השירשור יהיה רק העתק (עמוק!) של הפלינדרום שני.

אופרטור += כאופרטור הוספה:

האופרטור מקבל כארגומנט תו (char) ומוסיף אותו במקום הראשון (והאחרון) של הפלינדרום. לדוגמא, אם האובייקט ייצג את הפלינדרום "abba" והוספנו לו את התו 'c' הפלינדרום החדש הוא: "cabbac".

אופרטור -= אופרטור החסרה:

האופרטור מקבל כארגומנט אובייקט של Palindrome. במידה והארגומנט הוא תת פלינדרום של האובייקט שהפעיל את המתודה – תת הפלינדרום הנ"ל יחוסר מהפלינדרום המקורי. לדוגמא אם x מייצג את הפלינדרום "abbba" ו-y מייצג את הפלינדרום "bbb" אזי y הוא תת פלינדרום של x ולכן לאחר ביצוע $x=y$ ערכו של x יהיה "aa".
דוגמאות נוספות:

- x מייצג את הפלינדרום "abdba" ו-y את "bb", לאחר ביצוע $x=y$, x ייצג את הפלינדרום "ada".
- x מייצג את הפלינדרום "abcba" ו-y את "abba", לאחר ביצוע $x=y$, x ייצג את הפלינדרום "c".

אם הפלינדרום המתקבל (אופרנד ימני) אינו תת-פלינדרום של האופרנד השמאלי (שמפעיל את המתודה) – ערכו של האופרנד השמאלי לא ישתנה.

אופרטור + כאופרטור שירשור:

מבצע שירשור בין שני אובייקטים (כפי שהוסבר באופרטור +=) רק בלי לשנות אף אחד מהאובייקטים. תוצאת השירשור מוחזרת כאובייקט חדש מהמתודה.

אופרטור ++ postfix:

האופרטור ++ אופרטור קידום. הפעלת האופרטור על האובייקט גורמת לקידום התו הראשון בפלינדרום של האובייקט ב-1 (ולכן גם של האחרונה). לדוגמא, אם התו הראשון בפלינדרום היה 'a' אז הפעלת האופרטור גורמת להפיכת התו הראשון להיות 'b'. אם ערך התו לפני הקידום היא 'z' (או 'Z', או '9') אזי לאחר הקידום ערך התו יהיה 'a' (או 'A', או '0') בהתאמה.

במידה והפלינדרום ריק (אורכו אפס) לא יקרה דבר.

יש להקפיד ולממש את האופרטור ה-Postfix!

האופרטור == אופרטור השוואה:

מחזיר true אם 2 שני האובייקטים שווים. אובייקטים שווים הם אובייקטים שהמחרוזות שלהן שוות והגודל שלהם שווה. יש להתעלם מהבדלים של אותיות גדולות/קטנות (לדוגמא, המחרוזות "fL3Lf" ו-"fL3lF" שוות). ז"א שפלינדרומים שווים אם הם מייצגים את אותו פלינדרום תוך התעלמות מאותיות גדולות / קטנות.

האופרטור ! אופרטור שלילה:

הופך את כל התווים שהם אותיות במחרוזת ל-lower case (אותיות קטנות).

האופרטור [] אופרטור גישה:

מקבל מספר שלם i ומחזיר את התו ה-i בפלינדרום. אם i גדול יותר מאורך הפלינדרום או לא חוקי, יש להחזיר את התו '!'.
אופרטור () אופרטור המרה:

עושה casting מ-Palindrome ל-const char* לפי הפורמט המוסבר באופרטור פלט (בהמשך).

האופרטור < אופרטור השוואה:

מחזיר true אם הפלינדרום הראשון נמצא לפני הפלינדרום השני בסדר לקסיקוגרפי, ו-false אחרת (תוך התעלמות מאותיות קטנות / גדולות) ('a' < 'B', 'a' < 'b', 'A' < 'b').

האופרטור = אופרטור השמה:

ביצוע פעולת השמה בין 2 אובייקטים. יש לבצע העתקה עמוקה של המחרוזת תוך שימוש בזיכרון דינמי!

פונקציות גלובליות שקשורות למחלקה ויש לממש:

האופרטור << אופרטור פלט:

להדפסת המחרוזת. האופרטור מדפיס את הפלינדרום המיוצג ע"י האובייקט. ז"א אם הפלינדרום מכיל את המחרוזת "aScv" ואת הגודל 8 למסך תודפס המחרוזת "aScvvcSa".

האופרטור >> אופרטור קלט:

לקליטת מחרוזת לאובייקט. האופרטור קורא מחרוזת תווים מה-standard input. אם המחרוזת היא פלינדרום חוקי – היא דואגת שהאובייקט ייצג את אותו פלינדרום. אם המחרוזת היא לא פלינדרום חוקי האובייקט יקבל את הערך של מחרוזת ריקה וגודלו יהיה אפס.

שימו לב כי אות גדולה אינה זהה לאות קטנה ולכן:

פלינדרומים חוקיים:

"aBBa", "ssDfRgRfDss", "qw3E3wq", "23RR32"

לא חוקיים:

"9sd", "Abba", "abBa", "dfgFd", "qweEwq"

הערה: במימוש האופרטורים כפונקציות גלובליות עדיף לממשם כפונקציות friend.

איתחול ב-main:

בתוכנית זאת נעבוד עם שלושה אובייקטים של פלינדרום. נכנה אותם "פלינדרום1", "פלינדרום2" ו-"פלינדרום3".

יש לפעול לפי התפריט המתואר בסעיף הבא.

התפריט:

פעולה 1: יצירת פלינדרום

הסבר: יצירת פלינדרום חדש בעזרת הקונסטרוקטורים. בהפעלת אופציה זאת המערכת תצפה לשני קלטים נוספים.

קלטים:

(1) מספר שלם בין 1-3. מס' הפלינדרום אותו אנו רוצים ליצור.

(2) מחרוזת בשביל ליצור ממנה את הפלינדרום.

הערות

במידה ובקלט1 מתקבל מספר לא תקין יש להדפיס error ולחזור לתפריט.

במידה והפלינדרום כבר נוצר בעבר אין ליצור אובייקט חדש (ולא להדפיס כלום למסד).

פעולה 2: עידכון פלינדרום קיים

הסבר: עידכון של פלינדרום קיים בעזרת אופרטור קלט. המערכת תצפה ל-2 קלטים.

קלטים:

(1) מספר שלם בין 1-3. מס' הפלינדרום אותו אנו רוצים לעדכן.

(2) מחרוזת בשביל ליצור ממנה את הפלינדרום.

במידה ובקלט1 מתקבל מספר לא תקין או במידה והפלינדרום עוד לא נוצר יש להדפיס error ולחזור לתפריט.

פעולה 3: שירשור פלינדרומים

הסבר: שירשור שני פלינדרומים קיימים. באופציה זאת המערכת תצפה ל-3 קלטים.

קלטים:

(1) מספר שלם בין 1-3. מס' הפלינדרום שישמש כאופרנד השמאלי.

(2) מספר שלם בין 1-3. מס' הפלינדרום שישמש כאופרנד הימני.

(3) מספר שלם בין 1-3. מס' הפלינדרום שאליו תיכנס התוצאה.

הערות

שימו לב כי יש פה אפשרות להפעיל מספר אופרטורים. ז"א אם הערך של קלט1 ו-קלט3 זהים יש להפעיל את אופרטור +=, אחרת יש להפעיל את אופרטורים + ו= (או CCTOR) עם האובייקטים המתאימים.

אם הפלינדרומים שמקבלים בקלט 1 או 2 לא אותחלו עדיין, או שהמספרים שהתקבלו בקלטים 1,2 או 3 אינם חוקיים – יש להדפיס error ולחזור לתפריט.

פעולה 4: חיסור פלינדרומים

הסבר: חיסור שני פלינדרומים קיימים. באופציה זאת המערכת תצפה ל-2 קלטים.

קלטים:

(1) מספר שלם בין 1-3. מס' הפלינדרום שישמש כאופרנד השמאלי (ולכן גם הפלט).

(2) מספר שלם בין 1-3. מס' הפלינדרום שישמש כאופרנד הימני.

הערות

שימו לב כי יש להפעיל פה את אופרטור =-.

אם הפלינדרומים שמקבלים בקלט 1 או 2 לא אותחלו עדיין או שהמספרים שהתקבלו בקלטים 1 או 2 אינם חוקיים – יש להדפיס error ולחזור לתפריט.

פעולה 5: הוספה לפלינדרום

הסבר: הוספת תו לפלינדרום. המערכת תצפה לשני קלטים.

קלטים:

(1) מספר שלם בין 1-3. מס' הפלינדרום שישמש כאופרנד השמאלי (אליו נוסיף את התו).

(2) תו כלשהו שהוא או בין 'a'-'z' או בין 'A'-'Z' או בין '0'-'9'.

הערות

אם הפלינדרום שמקבלים בקלט 1 לא אותחל עדיין או שקלט 1 אינו מספר חוקי – יש להדפיס error ולחזור לתפריט.

התו תמיד יהיה תקין. אין חובה לבדוק אותו.

פעולה 6: השוואת פלינדרומים

הסבר: יש להשוות שני פלינדרומים. המערכת תצפה לשני קלטים. אם הפלינדרום הראשון (אופרנד שמאלי) קטן מהשני (אופרנד ימני) יש להדפיס למסך "1-", אם הם שווים יש להדפיס "0", אחרת יש להדפיס "1".

קלטים:

(1) מספר שלם בין 1-3. מס' הפלינדרום שישמש כאופרנד השמאלי.

(2) מספר שלם בין 1-3. מס' הפלינדרום שישמש כאופרנד ימני.

הערות

אם הפלינדרומים שמקבלים בקלט 1 או בקלט 2 לא אותחלו עדיין או שהמספרים המתקבלים אינם חוקיים – יש להדפיס error ולחזור לתפריט.

שימו לב כי ייתכן שתצטרכו להשתמש פה במספר אופרטורים.

פעולה 7: קידום פלינדרום

הסבר: יש לקדם את התו הראשון בפלינדרום ב-1. המערכת תצפה לקלט יחיד.

קלטים:

מספר שלם בין 1-3. מס' הפלינדרום שיש לקדם.

הערות

אם הפלינדרום המתקבל כקלט לא אותחל עדיין או שהמספר שהתקבל בקלט 1 אינו חוקי – יש להדפיס error ולחזור לתפריט.

פעולה 8: גישה לפלינדרום

הסבר: יש לגשת לפלינדרום המבוקש באינדקס המבוקש ולהדפיס את התו שמתקבל למסך. המערכת תצפה לשני קלטים.

קלטים:

(1) מספר שלם בין 1-3. מס' הפלינדרום שיש לגשת אליו.

(2) מספר שלם. האינדקס שבו אנחנו רוצים לגשת לפלינדרום.

הערות

אם הפלינדרום המתקבל כקלט לא אותחל עדיין או שהמספר המתקבל בקלט 1 אינו חוקי – יש להדפיס error ולחזור לתפריט.
שימו לב שאת הבדיקה האם האינדקס חוקי מבצעת המתודה של מחלקת פלינדרום!

פעולה 9: הדפסת פלינדרום

הסבר: יש להדפיס את הפלינדרום המבוקש למסך.

קלטים:

מספר שלם בין 1-3. מס' הפלינדרום שיש להדפיס.

הערות

אם הפלינדרום המתקבל כקלט לא אותחל עדיין או שהמספר המתקבל בקלט 1 אינו חוקי – יש להדפיס error ולחזור לתפריט.

פעולה 10: יציאה

הסבר: יש לצאת מהתוכנית בצורה מסודרת.

קלטים:

אין קלטים.

הערות:

- את ההצהרות וההגדרות של הפונקציות יש לחלק לקבצים כאשר לכל מחלקה יש קובץ H וקובץ cpp משלה. הפונקציה הראשית (main) ופונקציות חופשיות (במידה וקיימות כאלו) יכתבו בקובץ main.cpp.
- התוכנית תצפה לקליטת אפשרות כל עוד לא נבחרה אפשרות היציאה.
- בחירה באפשרות שאינה מופיעה בתפריט תגרום להדפסת הודעה error וחזרה לתפריט.

- יש להשתמש במספור האפשרויות כפי שמופיע בתרגיל!
- יש לבדוק תקינות קלט, גם כאשר אתם לא נדרשים לכך במפורש (למעט אם נאמר אחרת).
- **כל הדפסה תיעשה בשורה נפרדת.** אנא הקפידו על הניסוחים המדויקים של הקלט/הפלט שהראינו כאן (כולל אותיות גדולות/קטנות, שגיאות כתיב וכד').
- הניחו שהקלטים המספריים הם מספרים שלמים (אנו נכניס רק מספרים שלמים).
- בכל אופציה שאמורה לקבל מספר קלטים **תמיד** יתקבלו כל הקלטים. גם אם יש שגיאה בקלטים קודמים של אותה אופציה, יש לקלוט את הקלטים הנוספים ולהתעלם מהם.
- יש להקפיד על תכנות נכון :

- אין להשתמש במשתנים גלובליים!
 - בכל מקום שאפשר עדיף להגדיר `const`, `define` ו-`enum`, אין זה נכון להשתמש בקבועים מספריים.
 - יש להקפיד על `private` / `public` ועל **כימוס המידע (למעט ב-set ו-get לא דרושים)**.
 - **יש להקפיד על reuse בקוד.**
 - יש לרשום הערות בשפע!
 - בכל אחת מהמחלקות שהגדרנו אפשר (ורצוי אם צריך) להגדיר מתודות ומשתני עזר שיעזרו לכם בביצוע החישוב.
 - יש לנסות ולייעל את הקוד והתוכנית ככל שניתן.
 - יש להקפיד לשלוח / לקבל `by reference` בכל מקום אפשרי.
 - יש לאפשר שירשורים של אופרטורים בכל מקום שניתן לשרשר את האופרטורים הפרימיטיביים!
 - **יש להקפיד על כל כללי התכנות הנכון כפי שנלמדו בכיתה ומופיעים באתר.**
- לאחר כתיבת התוכנית מומלץ לעבור שוב על התרגיל ולוודא שמלאתם את כל ההוראות בתרגיל - בכל הסעיפים ובכל ההערות.

• בהצלחה ☺