

עבודת בית 1 – מעשי

נושא: רשימה מקושרת

תאריך פרסום: 10.11.2022

תאריך הגשה: 24.11.2022 עד שעה 23.55

הוראות מקדימות

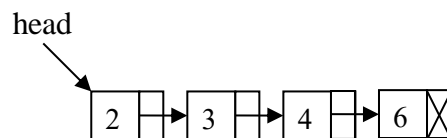
- ניתן להגיש עבודה בזוגות או ביחידים.

- איחור במועד ההגשה יגרור הורדה של ציון, 5 נק' לכל יום איחור או חלק ממנו.
בכל מקרה לא יהיה ניתן להגיש מעבר ל-3 ימי איחור ממועד ההגשה המקורי.
במקרים חריגים בלבד יש לפנות למרצה כדי לקבל אישור על הגשה באיחור.
- שאלות לגבי העבודה יש לשאול בפורום באתר הקורס ("מודל") או בשעות קבלה של המתרגלת האחראית
בלבד. אין לשלוח שאלות במייל המתרגלים או המרצה.
- יש לכתוב תכנית המכילה שלושה קבצים (קובץ header – קובץ עם סיומת h. בו תהיה הגדרה של מבנים ופונקציות בהם תשתמשו, קובץ המכיל כל גופי פונקציות וקובץ ראשי – main).
- שלושת הקבצים יש לכווץ לקובץ zip אחד שאותו תגישו לתיבת הגשה.
- **העבודה תיבדק באופן אוטומטי לפי הפלט אשר התוכניות שלכם תדפיס למסך. לכן, יש להקפיד על ההוראות ולבצע אותן במדויק. כל הדפסה אשר אינה עונה באופן מדויק לדרישות המופיעות בעבודה (כולל שורות, רווחים, סימני פיסוק, או כל תו אחר מיותר, חסר או מופיע בסדר שונה מהנדרש), תגרור להורדה בציון.**
- לנוחות, סיפקנו לכן תבנית של קבצי התכנית בתיקיה Pattern. תשתמשו בקבצי קוד הנתונים כבסיס לפתרון אליו תוסיפו את גוף פונקציות. אפשר להוסיף פונקציות עזר נוספות, אך אין לשנות חתימות של פונקציות שעליכם לממש.
- **להזכירכם: יש לשמור על הגינות אקדמית!**

עליהם לכתוב פונקציות הבאות המטפלות ברשימה מקושרת חד כיוונית (כל פונקציות מקבלות מצביה לראש רשימה מקושרת):

משימה 1

כתבו פונקציה בשם `int SumOddIndex(node* head)` המחשבת סכום איברים עם אינדקסים אי-זוגיים.
לדוגמה: לרשימה מקושרת



הסכום שווה ל-6 $(2+4=6)$.

משימה 2

כתבו פונקציה `int MinNode(node* head)` המקבלת רשימה מקושרת על ראש הרשימה המחפשת ומחזירה ערך הקטן ביותר ברשימה.

משימה 3

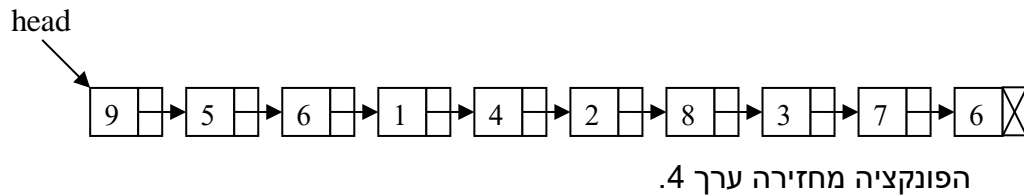
כתבו פונקציה להפיכת רשימה `void ReverseList(node**)`

לדוגמה: אם יש רשימה מקושרת הבאה: 2, 3, 6, 4 אחר הפיכה: 4, 6, 3, 2
שימו לב: לא יוצרים רשימה חדשה, השינויים על הרשימה המקורית.

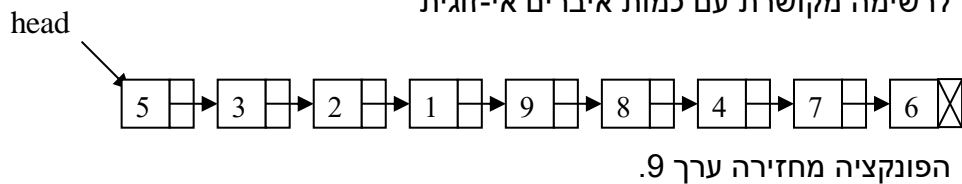
משימה 4

כתוב פונקציה `int MiddleList(node* head)` המקבלת רשימה מקושרת על יד מצביע לראש הרשימה. הפונקציה המחפשת ומחזירה איבר ה-"אמצעי" ברשימה. אם רשימה מקושרת מכילה כמות איברים אי-זוגית, הפונקציה מחזירה איבר אמצעי. אם כמות איברים זוגית, הפונקציה מחזירה איבר השמאלי מבין שניים שבאמצע.

לדוגמה: לרשימה מקושרת עם כמות איברים זוגית

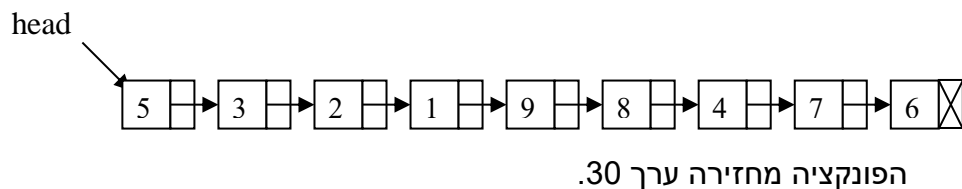


לרשימה מקושרת עם כמות איברים אי-זוגית



משימה 5

כתוב פונקציה `int MultHeadTail(node* head)` המקבלת רשימה מקושרת על יד מצביע לראש הרשימה. הפונקציה מחשבת ומחזירה תוצאה של כפל בין איבר הראשון ואיבר האחרון של רשימה. לדוגמה לרשימה מקושרת



משימה 6

כתבו פונקציה `void ifSorted(node* head)` הבודקת האם רשימה ממוינת בסדר עולה.

- אם הרשימה ממוינת, הפונקציה תדפיס הודעה "Sorted", אחרת היא תדפיס "Not sorted".

צבירה נעימה!