

מבוא למדעי המחשב – תשע"ו**מטלה מספר 4 – מצביעים מערכים, מחרוזות והקצאת זיכרון דינאמית**

כתבו וערכו: ד"ר דוד טנקוס וד"ר שייקה בילו

כללי

(1) במהלך התרגול בקורס תכנות בשפת C עליך לבצע מספר מטלות הגשה אותם יש להגיש במועד שיקבע המרצה. כל מטלה תוגש למרצה בתקשוב בלבד באופן מושלם במועד שנקבע ופורסם על ידו. עליך לקרוא היטב את המשימות המוגדרות במטלה ולבצע אותן במלואן באופן עצמאי.

מטרות

(1) לתרגל באופן עצמאי מספר רב ככול שניתן של פתרון בעיות מגוונות הדורשות פתרון בתכנות בשפת C כדי לצבור את מלוא הניקוד במסגרת הקורס בחלק ביצוע מטלות הגשה וכן להגיע מוכן, לבחינת הסיום.

הנחיות

- (1) עליך לקרוא היטב כל סעיף במטלה, את הדרישות לפתרון ואת התוצאה הנדרשת לאחר ביצוע הרצת התוכנית.
- (2) עליך לוודא כי במהלך כתיבת התוכנית נשמר עקרון הכתיבה המובנית.
- (3) יש לוודא כי לכל המשתנים יש שמות בעלי משמעות ולוודא כי אין הגדרות מיותרות ולוודא את חוקיות הקלט!
- (4) יש להקפיד על כך שכל המשתנים יוגדרו בסוג הטיפוס הרלוונטי להם ביותר להצלחת התוכנית.
- (5) יש לוודא שימוש נכון ב- `enum const define`.
- (6) יש להקפיד כי לכל פונקציה בתוכנית יינתן שם בעל משמעות המרמז מה היא אמורה לבצע.
- (7) יש להקפיד כי בכל מהלך התוכנית, ובמיוחד בפונקציות, במקומות מפתח, יופיעו הנחיות, הסברים והגדרות עבור המשתמש המסבירות: מה עושה קטע התוכנית, מה על המשתמש להזין, מה הוא מתבקש לבצע כדי שהתוכנית תפעל נכון, האם יש הנחיות מיוחדות להרצת התוכנית והפונקציות שבתוכה.
- (8) לצורך הנחיית המשתמש יש להשתמש ב- `printf` כולל ירידת שורה לפני כל שימוש ב- `scanf` לקליטת נתונים.
- (9) עבור כל המטלות עליכם להגיש את קובץ קוד מקור אחד (קבצי C) המכיל תכנית בשפת C. בתכנית פונקציה ראשית אחת, ופונקציות נוספות.
- (10) יש לוודא כי במידה ואתם נדרשים לכך התוכנית תטפל בחריגים כדוגמת קלט לא תקין מהמשתמש וכו'.
- (11) על התוכנית לענות בדיוק על מה שנדרש ותוך שמירה על ידידותיות למשתמש בכל מהלך ריצת התוכנית.
- (12) כל סטודנט יכין ויגיש את העבודה באופן עצמאי, אין לשתף או להעתיק את העבודה או חלקים ממנה, עבירה על הוראה זו תוביל לציון 0 בעבודה. ההגשה תתבצע דרך מערכת מודול בלבד. שום עבודה לא מתקבלת במייל!
- (13) בתחילת הקובץ יש להוסיף הערה הכוללת אם שם הסטודנט, מס' ת.ז. ושם הקמפוס.
- (14) סביבת העבודה המומלצת לעבודה היא Visual Studio C++ 2013.
- (15) בתחילת הקובץ יש לרשום בברור את פרטי המגיש:

/* Assignment: 3 Campus: Ashdod / Beer Sheva Author: Israel Israeli, ID: 01234567*/

מועד הגשה: את המטלה יש להגיש עד 31.12.2015

שיטת עבודה

במטלה זו עליך לכתוב תוכנית הכוללת ארבע פונקציות: כל אחת משלושת הפונקציות הראשונות תכלול פתרון מושלם של סעיף מהמטלה. הפונקציה הראשית בתכנית תציג למשתמש תפריט הכולל את שלושת הפונקציות האחרות בתכנית. המשתמש יבחר אחת מהפונקציות הללו, ויקליד את הפרמטרים שלה. הפונקציה הראשית תדפיס את הערך שהפונקציה הנקראת החזירה. לאחר מכן יוצג התפריט שוב ושוב עד שהמשתמש יבחר לסיים את ריצת התכנית.

סטודנטים המגישים גם את שאלת הבונוס, יגישו שני קבצי C מכווצים לקובץ אחד בפורמט RAR או ZIP. סטודנטים שלא מגישים את שאלת הבונוס, יגישו קובץ C אחד.

סעיף ראשון (33%) – מצביעים ומערכים

כתוב את הפונקציה הבאה:

int *symmetricDiff(int *A, int sizeA, int *B, int sizeB, int *pNewSize);

הפונקציה מקבלת כפרמטרים: שני מערכים ואת גדליהם, כלומר המערך A וגודלו sizeA, המערך B וגודלו sizeB, ועוד מצביע למספר שלם, pNewSize.

הפונקציה מייצרת מערך חדש שיכיל את ההפרש הסימטרי של המערכים. כלומר, את כל המספרים שמופיעים במערך A אך לא מופיעים במערך B, ואת המספרים שמופיעים במערך B אך לא מופיעים במערך A. הפונקציה מייצרת את המערך באופן דינמי (כיוון שלא ניתן לדעת מראש את גודלו) ומחזירה את כתובת המערך החדש שנוצר.

הפרמטר pNewSize הוא מצביע למשתנה מסוג מספר שלם המייצג את גודלו של המערך החדש שנוצר. כלומר הפונקציה צריכה לעדכן דרכו (דרך הכתובת שהתקבלה) את גודל המערך החדש.

לדוגמא אם הפרמטרים שהתקבלו הם:

A = {1, 3, 4, 6, 10, 13, 14, 7, 8, 9}; sizeA = 10

B = {2, 3, 6, 15, 16, 18, 20, 9, 11, 13}; sizeB = 10

אזי הפונקציה תייצר ותחזיר מצביע למערך החדש:

{1, 4, 7, 8, 10, 14, 2, 11, 15, 16, 18, 20}

בנוסף, הפונקציה תעדכן דרך המשתנה pNewSize שגודל המערך החדש הוא: 12

הפונקציה הראשית קולטת מהמשתמש את המערכים A, B וגדליהם. היא מדפיסה את מערך התוצאה.

סעיף שני (33%) – מצביעים ומחרוזות

כתוב את הפונקציה הבאה:

int isEquivalent(char*lines[], int length, char* text);

הפונקציה מקבלת כפרמטרים: מערך של מחרוזות lines, את גודל המערך הזה length, ומחרוזת נוספת text. הפונקציה מחזירה 1 אם שרשרת המחרוזות שבמערך lines (לפי סדר השורות) שווה למחרוזת text, ומחזירה 0 בכל מקרה אחר. לדוגמה, אם המערך הוא:

char * arr[] = {"to ", "from ", "fine"};

והמחרוזת text היא: "to from fine" הפונקציה תחזיר 1

אבל אם המחרוזת text היא: "to here from" הפונקציה תחזיר 0

שים לב: כיוון שלא ידוע אורך המחרוזות שיישלחו, יש להשתמש בהקצאה דינמית כדי לייצר את המחרוזת המשורשרת בדיוק על פי הכמות הנדרשת.

סעיף שלישי (33%) – העברת פרמטרים

כתוב את הפונקציה הבאה:

void split(int n, int *pSmallDigits, int *pBigDigits);

הפונקציה מקבלת מספר שלם חיובי בשם n. לא ידוע כמה ספרות יש ב-n. כמו כן, הפונקציה מקבלת 2 מצביעים לשלמים.

הפונקציה תיצור שני מספרים מהספרות של n באופן הבא:

המספר הראשון יורכב מהספרות של n שהן קטנות מ-5 (כלומר 0,1,2,3,4) לפי סדר הופעתן במספר. המספר השני יורכב משאר הספרות (כלומר 5,6,7,8,9), שוב לפי סדר הופעתן במספר.

הפונקציה הראשית בתכנית מדפיסה את שני המספרים ש-split מוצא.

לדוגמה, אם הפרמטר הוא n=380512 אז

***pSmallDigits=3012, *pBigDigits=85.**

דוגמה נוספת, אם הפרמטר הוא n=13220 אז

***pSmallDigits=13220, *pBigDigits=0.**

דוגמה נוספת, אם הפרמטר הוא n=9905 אז

***pSmallDigits=0, *pBigDigits=995.**

סעיף בונוס (20%)

כתוב תוכנית הממשת ערימה של קלפים. לכל קלף יש צבע מסויים. בשלבים שונים של התכנית ניתן להוסיף קלף לראש הערימה, או לקחת את הקלף שבראש הערימה. אין אפשרות להוסיף קלף או לקחת קלף ממקום אחר בערימה. התכנית כוללת את ההגדרה הבאה:

enum {blue, green, yellow, orange, red, purple, white, black, brown, grey, pink, MAX};

בפונקציה הראשית מוגדר מערך בשם cards המכיל את הקלפים שבערימה, והמשתנה top מכיל את האינדקס של הקלף העליון. לדוגמא, אם בתחתית הערימה יש קלף כחול, מעליו שני קלפים סגולים, ומעליהם קלף ירוק, אז:

cards[0]=0, cards[1]=5, cards[2]=5, cards[3]=1, top=3

האיברים של המערך cards בעלי אינדקס גדול מ-top לא מעניינים אותנו. מבחינתנו הם מכילים זבל. אם ערימת הקלפים ריקה אז top=-1. בתחילת התכנית המשתמש מקליד את ערכו של המשתנה size, ולאורך כל התכנית אסור שהערימה תכיל יותר מאשר size קלפים.

התכנית כוללת את הפונקציות הבאות:

- **find_code** המקבלת שם של צבע (=מחרוזת), ומחזירה את השווי שלה. לדוגמא, אם הפרמטר הוא "blue", אז הפונקציה תחזיר 0. אם הפרמטר הוא "green", אז הפונקציה תחזיר 1. אם הצבע לא מופיע בהגדרה של enum, יוחזר -1.
- **print_color** המקבלת כפרמטר מספר שלם, ומדפיסה את שם הצבע המתאים. לדוגמא, אם הפרמטר הוא 0, אז יודפס "blue". אם הפרמטר הוא 1, יודפס "green".
- **is_empty** מחזירה 1 אם ערימת הקלפים ריקה. אחרת, הפונקציה מחזירה 0.
- **is_full** מחזירה 1 אם ערימת הקלפים מלאה. (בגודל size). אחרת, מחזירה 0.
- **print** מדפיסה את צבעי הקלפים שבערימה מלמעלה למטה. כלומר, הצבע שבראש הערימה יודפס ראשון.
- **insert** מקבלת כפרמטר מספר שלם. אם הערימה לא מלאה אז הפונקציה מוסיפה את הקלף המתאים לראש הערימה ומחזירה 1. אחרת, היא מחזירה 0, ולא משנה את הערימה.
- **remove** אם הערימה ריקה, הפונקציה מחזירה 0, ולא עושה דבר. אחרת, היא מחזירה 1, מוציאה את הקלף העליון, ומעבירה אותו לפונקציה הראשית באמצעות פרמטר-מצביע.
- **main** מקצה זיכרון באופן דינמי למערך cards, ולאחר מכן מבקשת שוב ושוב מהמשתמש לבצע פעולות על ערימת הקלפים, עד שהמשתמש בוחר לסיים. בנוסף, הפונקציה מבצעת את הפעולות הבאות:
 - הפונקציה מדפיסה הודעות שגיאה אם הפונקציות insert, remove מחזירו 0.
 - אם המשתמש מנסה להוסיף לערימה קלף בצבע שאינו קיים, תודפס הודעת שגיאה, ולא קוראים לפונקציה insert.
 - מודפסת הודעה מתאימה לאחר קריאה ל-is_full, is_empty.
 - לאחר הוצאת קלף מהמחסנית, הפונקציה מדפיסה את צבעו.

שימו לב: להעביר לכל פונקציה את הפרמטרים הדרושים לה. הפונקציות insert, remove משנות את המשתנה top שנמצא בפונקציה הראשית.

דוגמא לריצת התכנית:

Enter max number of cards: 4

e

Empty

f

Not full

p

i orange

p

orange

e

Not empty

f

Not full

i red

i brown

i pink

i purple

Error: Full

p

pink

brown

red

orange

e

Not empty

f

Full

d

pink

d

brown

p

red

orange

d

red

d

orange

d

Error: Empty

i grin

Wrong color

e

Empty

q

Goodbye

בהצלחה !!!