שפת C – תרגיל בית 4

הנחיות כלליות:

- ההגשה היא בבודדים ולא בזוגות
- יש להקפיד על נתינת שמות משמעותיים למשתנים
- יש להקפיד על הזחות (TAB) נכונות.
- יש להקפיד על הגדרת "מספרי קסם" כ #define
- לחלק את הקוד ל קבצי .c ולקבצי .h
- יש לכתוב כמה שיותר פונקציות עזר אבל לא יותר מדי – יש לאחד פונקציות כשאפשר.
- יש לכתוב את הקוד ביעילות מקום/זמן הטובה ביותר.
- יש לדאוג שהקוד מתקמפל ורץ ואז לדאבג (debug) אותו ע"י הרצות חיוביות ושליליות.
- יש למצוא את מבנה הנתונים שיביא את השאלה לפתרון ביעילות הגבוהה ביותר.

תרגיל 1:

א. יש לכתוב פונקציה take()
אשר מקבלת מספר אחד בכל פעם, ומחזירה את המספר האמצעי מכל המספרים שקיבלה עד כה.
הפונקציה צריכה להחזיק את כל המספרים במבנה נתונים מסוים. אין לדעת כמה פעמים תיקרא הפונקציה, ועליה להקצות זיכרון בדיוק בגודל שהיא צריכה להחזיק ולא יותר.

דוגמת הרצה :

```
take(20)
>> 20
take(10)
>> 10
take(30)
>> 10
take(5)
>>30
take(40)
>>30
```

על הפונקציה לעבוד ביעילות של O(1)

ב. כעת יש לכתוב את הפונקציה take2() כך שתחזיר את החציון, כלומר המספר שחצי מהמספרים נמצאים מתחתיו וחצי מעליו . על הפונקציה למצוא את החציון עדיין ב O(1)- עליכם לחשוב איזה מבנה/מבני נתונים יכולים לעשות זאת, ולממש את הפונקציה.

דוגמת הרצה :

```
take2(20)
>> 20
take2(10)
>> 10
take2(30)
>> 20
take2(5)
>>10
take2(40)
>>20
```

תרגיל 2:

נממש את הפונקציה bool **isStringBalanced**(char** sentence, int size)
אשר מקבלת רשימה של מילים ומחזירה האם הסוגריים מאוזנות.

ישנם שלושה סוגי סוגריים פותחים }[) ושלושה שסוגרים, כל אחד סוגר את חברו {[(

סוגרים מאוזנים נראים כך : {[]}
סוגרים לא מאוזנים נראים כך : {[]}) או כך ({}

הקלט עשוי להראות כך : if (x[i] > x[j]) { return;}
במקרה זה הפונקציה תחזיר false .
עליכם לחשוב איזה מבנה נתונים יהיה הכי יעיל כאן, ולממש את הפונקציה.

תרגיל 3:

א. אנו מעוניינים לכתוב מחלקה (class) בשפת C . למרות שאין כזה דבר אנחנו נשתמש במבנים (struct) . המבנה שלנו ייקרא myClass ויהיה בו איבר מסוג int m_x
ופוינטר לפונקציה add() אשר תקבל שני פרמטרים: מצביע ל myClass ובו יועבר מצביע לאובייקט עצמו (this), ופרמטר y . הפונקציה תוסיף את y ל m_x.

עליך לכתוב תוכנית אשר תגדיר אובייקט מסוג myClass . תאתחל אותו, תדפיס את m_x ותקרא לפונקציה add() ואח"כ שוב תדפיס את m_x .

ב. כעת אנו מעוניינים לייצר מנגנון הורשה .
ישנם שני סוגי הורשה :

- **הורשה דורסת (overriding)**
הורשה דורסת משמרת את המבנה הקיים אך מחליפה חלק מהפונקציות.
למשל המחלקה Derived תשמר את המבנה של myClass, אך במקום שהפונקציה add() תוסיף את y ל m_x היא תעשה xor ביניהם.
כיצד ממשים הורשה כזו בסי ? הדגם בקוד.

- **הורשה מרחיבה (extending)**
הורשה מרחיבה מוסיפה על המבנה הבסיסי של מחלקת האב עוד איברים ופונקציות.
לדוגמא המחלקה Derived2 יהיה לה עוד איבר m_y. בקריאה לפונקציה add() חוץ מהוספת y לאיבר m_x, גם נשמור את ערך הפרמטר y לתוך האיבר m_y. איך ממשים דבר כזה ? הדגם בקוד.

בהצלחה !!!