#### logo

**מחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT**

סדנה מתקדמת בתכנות 61108

סמסטר ב' תש"פ

**מטלה 1**

**מערכים דינאמיים, מבנים, רשימות מקושרות**

**שאלה 1:**

כתבו פונקציה בשם powerArray המקבלת מספר שלם n ומחזירה מצביע למערך בגודלn ומטיפוס unsigned int, שהוקצה באופן דינאמי בתוך הפונקציה.

כל תא במערך אמור להכיל 2 בחזקת שארית החלוקה של האינדקס ב-32. כלומר, ערכו של תא במקום ה- i יהיה 2i mod 32.

**למשל**,

אם נתון:

unsigned int \*p;

בעקבות ההוראה:

p=powerArray(5);

יתקבל המצב הבא:

p

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 16 | 8 | 4 | 2 | 1 |

0 1 2 3 4

יש לטפל (באופן המועדף אליכם) גם במצב שבו אין מספיק מקום בזיכרון להקצאה!

**שאלה 2:**

כתבו פונקציה בשם matrixMultiplication שמממשת כפל מטריצות.

הפונקציה תקבל שני מערכים דו-ממדיים סטטיים וגודלם, תקצה מקום למטריצה החדשה (דינאמית!), תחשב את מכפלת המטריצות, ותחזיר את כתובתו של המערך החדש.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

אם A=(aij) היא מטריצה m×n ו-B=(bjk) היא מטריצה n×p אז C=AB היא מטריצה m×p שלכל איבר cik שלה מתקיים



**לינק לויקיפדיה בנושא כפל מטריצות:**

<https://he.wikipedia.org/wiki/%D7%9B%D7%A4%D7%9C_%D7%9E%D7%98%D7%A8%D7%99%D7%A6%D7%95%D7%AA>

שימו לב!

בכדי שנוכל להכפיל מטריצות מס' העמודות של המטריצה הראשונה צריך להיות זהה למספר השורות של המטריצה השנייה.

**שאלה 3:**

כתבו פונקציה בשם createArrayAndList המקבלת מטריצה דינאמית ומוצאת את כל איברי המטריצה שמכילים ערך ששווה לסכום הקואורדינטות שלהם (i+j).

על הפונקציה לבנות מערך דינאמי חד-ממדי של שלשות, וכן רשימה מקושרת של שלשות. כל שלשה במערך או ברשימה תכיל את ערכו של איבר המטריצה שעונה לתנאי הנ"ל, והקואורדינטות i ו-j (מס' השורה ומס' העמודה) של אותו האיבר במטריצה. הפונקציה תחזיר את גודל המערך (ששווה לאורכה של הרשימה), ותעביר (by reference) את המערך ואת הרשימה.

**לדוגמא, עבור המטריצה מקורית:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **i \ j** | **0** | **1** | **2** | **3** | **4** |
| **0** | 3 | 2 | 5 | 3 | 6 |
| **1** | 8 | 3 | 4 | 6 | 5 |
| **2** | 0 | 6 | 4 | 5 | 7 |
| **3** | 9 | 8 | 6 | 6 | 1 |

**סדרת השלשות תהיה:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 6 | 5 | 4 | 5 | 3 | איבר |
| 3 | 2 | 2 | 1 | 0 | i |
| 3 | 3 | 2 | 4 | 3 | j |

**והפונקציה תחזיר 5.**

למימוש השלשות יש להשתמש בטיפוס struct בעל שלושה שדות נומריים.

איברי המערך יהיו מבנים מסוג זה.

כל איבר של הרשימה יהיה מורכב משני שדות: נתון ומצביע לאיבר הבא. הנתון הוא struct שלשה (כמו איברי המערך).

יש להשתמש בפונקציות עזר הבאות:

1. פונקציה בשם createThree המקבלת שלושה מספרים שלמים ומחזירה שלשה אחת המורכבת משלושת הפרמטרים שקיבלה.
2. פונקציה בשם createThreeList המחזירה רשימה מקושרת של שלשות על פי תנאי התרגיל. בחרו את הפרמטרים הכי מתאימים שהפונקציה תקבל.
3. פונקציה בשם createThreeArr המחזירה מערך של שלשות על פי תנאי התרגיל. בחרו את הפרמטרים הכי מתאימים שהפונקציה תקבל.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

**שאלה 4:**

כתבו פונקציה בשם createList אשר מקבלת רשימה מקושרת (ללא איבר דמה) של מספרים שלמים אי-שליליים.

על הפונקציה להסיר את האיברים האי-זוגיים מהרשימה ולבנות רשימה חדשה (ללא איבר דמה) המורכבת מהמספרים האי-זוגיים של הרשימה המקורית. יש לשמור ברשימה החדשה על הסדר של האיברים האי-זוגיים של הרשימה המקורית.

**לדוגמא**,

הרשימה המקורית בעלת תוכן 3, 6, 1, 9, 8, 4, 5 (משמאל לימין) תעודכן לרשימה המורכבת ממספרים 6, 8, 4 (משמאל לימין). איברי הרשימה החדשה יהיו 3, 1, 9, 5 (משמאל לימין).

הפונקציה תחזיר מצביע לרשימה החדשה (של מספרים אי-זוגיים).

שימו לב:

* המצביע לרשימה המקורית יכול להשתנות אם האיבר הראשון שלה נמחק.
* הרשימה המקורית יכולה להיות ריקה, יכולה להיות מורכבת רק ממספרים זוגיים ורק ממספריים אי-זוגיים.

יש לכתוב ולהשתמש בפונקציה בשם Delete למחיקת איבר מהרשימה המקורית (כולל שחרור זיכרון) ובפונקציה בשם Insert להכנסת איבר לרשימה החדשה (כולל הקצאת זיכרון). ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

על פונקציית העזר אשר תקלוט את הנתונים ותבנה את הרשימה המקורית להשתמש במספר שלילי כזקיף לסיום הקלט.

**שאלה 5:**

כתבו פונקציה בשם createList2 אשר מבצעת את אותה המשימה כמו הפונקציה מסעיף 4, אבל בניגוד לפונקציה מסעיף 4, הפונקציה הנוכחית לא תשתמש בשחרור ובהקצאת זיכרון.

על הפונקציה לשנות את הקישורים ברשימה המקורית, לנתק את האיברים האי-זוגיים ממנה ולקשר אותם ברשימה החדשה.

**הערות:**

1. **שמות הפונקציה צריכות להיות זהות לשמות הרשומים במטלה.**
2. **יש לאחד את כל השאלות הנ"ל לתכנית אחת באמצעות תפריט הפונקציות Ex1(), …, Ex5() (ראה להלן) המשמשות להפעלת פונקציות השאלות 1-5 בהתאמה ומטפלות בהכנה ובקליטה של הפרמטרים, בהדפסת התוצאות ובשחרר זיכרון דינאמי (חובה כאשר אין בו צורך).**
3. **יש להשתמש בפונקציות עזר לבניית אובייקטים דינאמיים מקוריים (עבור שאלות 3-5) ולשחרור זיכרון דינאמי (עבור שאלות 2-5). ניתן להניח שיש בזיכרון מספיק מקום להקצאה.**
4. **יש לקלוט את כל הנתונים המקוריים. לקליטת אובייקטים מורכבים (עבור שאלות 2-5) להשתמש בפונקציות עזר.**
5. **יש להשתמש ב-#define לקביעת גדלי המערכים הסטטיים המקוריים (שאלה 2).**
6. **יש להציג את כל התוצאות (תוך שימוש בפונקציות עזר). בשאלות 2-5 יש גם להציג את נתוני הקלט.**
7. **יש להשתמש בשמות משמעותיים וגם בהערות.**
8. **יש להקפיד לכתוב בצורה מבנית.**
9. **יש להקפיד על ממשק ידידותי ככל האפשר. אין צורך בבדיקת תקינות הקלט.**
10. **תכנית שלא עוברת קומפילציה לא תתקבל!**

int main()

{

int select=0, i, all\_Ex\_in\_loop=0;

printf("Run menu once or cyclically?\n(Once - enter 0, cyclically - enter other number) ");

if (scanf("%d", &all\_Ex\_in\_loop) == 1)

do

{

for (i=1; i<=5; i++)

printf("Ex%d--->%d\n", i, i);

printf("EXIT-->0\n");

do {

select=0;

printf("please select 0-5 : ");

scanf("%d",&select);

} while ((select<0)||(select>5));

switch (select)

{

case 1: Ex1(); break;

case 2: Ex2(); break;

case 3: Ex3(); break;

case 4: Ex4(); break;

case 5: Ex5(); break;

}

} while (all\_Ex\_in\_loop && select);

return 0;

}

יש להשתמש כתבנית בקובץ Assignment\_1\_template.c המצורף המכיל את הקוד. צריך להכניס בתוכו את כל ההצהרות וההגדרות הפונקציות הנדרשות במקום המתאים.