

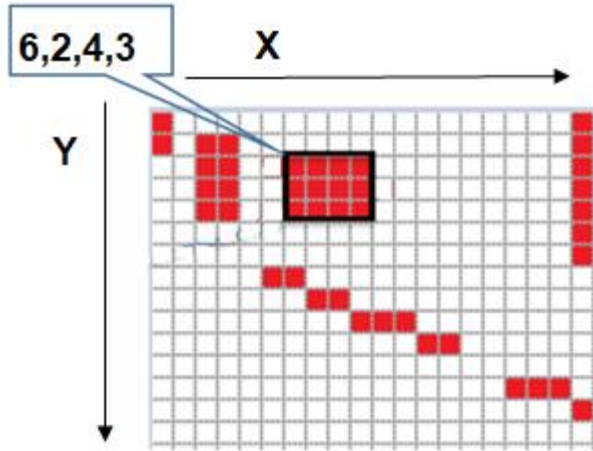
מבחן 10010 שאלון X

הוראות כלליות:

1. אין לשנות מבנים, הגדרות פונקציות למעט בנושא רשימות מקושרות
2. יש לממש כל שאלה בקבצים השייכים לה בלבד, אין להוסיף קבצים לפרויקט
3. אין להשתמש במשתנים גלובלים, בפונקציה exit וב goto
4. אין להגדיר מטריצות בצורה דינאמית ללא הקצאה דינאמית.
5. כל דבר שלא צוין שאסור אזי מותר

שאלה 1 (65 נקודות) : (קבצים q1.c q1.h)

בשאלה זו נטפל בנתונים של תמונה בעלת 2 צבעים בלבד, לבן ואדום (בטופס המבחן כל משבצת אדומה מודפסת כמשבצת אפורה). לבן מיוצג ע"י המספר 0 ואדום ע"י המספר 1. התמונה היא בעצם רקע לבן ובתוכו מלבנים אדומים. כל משבצת במטריצת התמונה באיור מתארת פיקסל של תמונה. ידוע שמלבן אחד יכול להיות צמוד למלבן אחר אבל רק בפינות.



שים לב: ההדפסה של הטופס היא בשחור לבן ולכן משבצת אדומה מודפסת כמשבצת אפורה.

נגדיר מבנה מלבן

מלבן מיוצג ע"י 4 מספרים המציינים את ה (x,y) של הנקודה השמאלית עליונה ורוחב ואורך המלבן בהתאמה.

```
typedef struct {
    int upperLeftX; //the col index of the upper left corner
    int upperLeftY; //the row index of the upper left corner
    int width;       //the width of the rect
    int height;      //the height of the rect
}Rect;
```

את נתוני התמונה נשמור בקובץ בינארי בפורמט הבא:

1. int - מספר השורות בתמונה (rows)
 2. int - מספר העמודות בתמונה (cols)
 3. ערכי התמונה: 1 לאדום 0 ללבן. כל ערך שמור כ int כך שסה"כ יהיו cols*rows ערכי int.
- שים לב: ערכי התמונה ישמרו כמערך חד ממדי.

סעיף א' (10 נקודות):

כתוב פונקציה

```
int* getImage(const char* fileName, int* pRow, int* pCol);
```

הפונקציה מקבלת שם קובץ בינארי ומשתנים אשר יחזירו את ממדי התמונה הפונקציה תקרא את הנתונים ותחזיר כתובת לתחילת ערכי התמונה (int*). פורמט הקובץ מתואר למעלה. לא יופיעו לולאות בפונקציה.

סעיף ב' (20 נקודות):

כתוב פונקציה

Rect* createRectArr(const int* image, int row, int col, int* pCount);

המקבלת את נתוני התמונה ומשתנה אשר יחזיר את אורך מערך המלבנים. הפונקציה מייצרת מערך של כל המלבנים בתמונה. **רמז:** עבור על המערך מתחילתו, בדוק אם פיקסל מגדיר מלבן חדש - פיקסל אדום שאין בשורה מעליו או בעמודה לפניו פיקסל אדום. אחרי מציאת פיקסל התחלת מלבן מצא את כל הפיקסלים השייכים לאותו מלבן.

סעיף ג' (10 נקודות):

כתוב פונקציה

int saveRectToFile(const char* fileName, Rect* rectArr, int count);

המקבלת שם קובץ, מערך של מלבנים ואורך המערך ושומרת בקובץ בפורמט text את נתוני המלבנים בפורמט הבא:

10					1. שורה ראשונה מספר המלבנים
0	0	1	2		
19	0	1	7		2. משורה שניה והלאה נתוני כל מלבן : upperLeftX, upperLeftY, width, height
2	1	2	4		
6	2	4	3		
5	7	2	1		הפונקציה תחזיר 1 להצלחה ו 0 לכישלון.
7	8	2	1		
9	9	3	1		התמונה משמאל מייצגת את קובץ המלבנים עבור קובץ התמונה שניתן בשאלה.
12	10	2	1		
16	12	3	1		
19	13	1	1		

סעיף ד' (20 נקודות):

כיוון שהתמונה מורכבת רק משתי אופציות 0/1 רוצים לדחוס את האינפורמציה ב Byte בודד ניתן לשמור נתוני 8 פיקסלים. במידה וממדי התמונה לא מתחלקים ב 8 יהיה Byte אחד נוסף בסוף עם נתונים חלקיים. את הנתונים בפורמט דחוס רוצים לכתוב לקובץ בינארי חדש. כתוב פונקציה

int saveCompressImageToFile(const char* fileName, const int* image, int row, int col);

המקבלת שם קובץ ואת נתוני התמונה. הפונקציה שומרת אל הקובץ את נתוני התמונה בצורה בינארית דחוסה. הפונקציה תחזיר 1 להצלחה ו 0 לכישלון.

פורמט הקובץ יהיה:

1. int - מספר השורות בתמונה (rows)
2. int - מספר העמודות בתמונה (cols)
3. כל ה Bytes המייצגים את התמונה.

סעיף ה' (5 נקודות):

נתון השלד של הפונקציה הראשית $q1()$ אשר מבצעת את הפעולות הבאות:

1. קוראת מקובץ בינארי בשם **mat1.bin** את התמונה
2. מייצרת את מערך המלבנים
3. שומרת את מערך המלבנים לקובץ **txt** בשם **rects.txt**
4. שומרת את התמונה בפורמט דחוס לקובץ בינארי בשם **mat1_comp.bin**

יש להוסיף בדיקות ופעולות אחרות אשר נדרשות.

שאלה 2 (35 נקודות) : (קבצים q2.c q2.h)

מוגדר פולינום

$$P(x) = \sum_{i=0}^{n-1} a_i x^i$$

כל איבר בפולינום נקרא מונום

$$M(x, i) = a_i x^i$$

אפשר לרשום

$$P(x) = \sum_{i=0}^{n-1} M(x, i)$$

נגדיר מבנים המייצגים מונום ופולינום:

```
typedef struct {
    float coefficient;
    int power;
} Monom;
```

```
typedef struct {
    Monom *monoms; // Array of Monoms
    int size;       // Number of Monoms in Polygon
} Polynom;
```

בשאלה זו הפולינום ייוצג כרשימה מקושרת חד כיוונית מטיפוס Monom המקיימת:

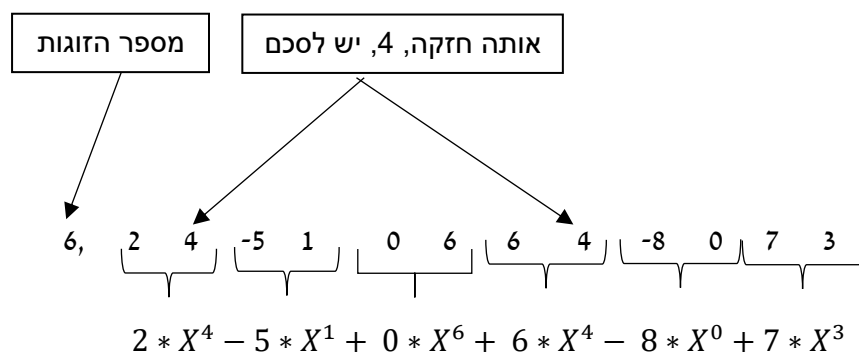
1. הרשימה לא תכיל מונום שמקדמו 0
2. המונומים יופיעו בסדר חזקות יורד ממש כלומר המונום עם החזקה הגדולה ביותר יופיע כאיבר הראשון ברשימה.

createAndPrintPoly

הפונקציה מקבלת כקלט זוגות של מספרים המייצגים מקדם וחזקה מספר הזוגות יכול להשתנות. הארגומנט הראשון שהפונקציה מקבלת מציין את מספר הזוגות בקלט. יתכן ויהיו כמה זוגות המציינים חזקה זהה (יש לסכם אותם). יתכן ויהיה מקדם 0 וממנו יש להתעלם, ידוע אין מקדם שלילי.

משימות הפונקציה:

1. (5 נקודות) טיפול במספר משתנה של זוגות הקלט
 2. (20 נקודות) ייצור רשימה מקושרת חד כיוונית מתאימה כך שהמונומים ברשימה יופיעו בסדר חזקות יורד ממש בכל רגע נתון. אין להשתמש בכל מבנה עזר.
 3. (10 נקודות) הדפסת נתוני הפולינום ע"י הדפסת הרשימה במדויק על פי פורמט הפולינום המתואר בהמשך.
- ניתן להשתמש בקוד הרשימה המקושרת המצורף ולשנותו כרצונך. מומלץ לחלק לתתי פונקציות נוספות לדוגמא:



מייצג את הפולינום הבא:

$$8 * X^4 + 7 * X^3 - 5 * X - 8$$

הדפסת פולינום:

יש להדפיס פולינום שנקלט בסדר חזקות יורד, אין להדפיס חזקות עם מקדמים 0, אין להדפיס X כשהחזקה היא 0 ואין להדפיס 1 כשהחזקה היא 1.

את הפולינום: $8x^4 + 7x^3 - 5x - 8$ יש להדפיס: $8x^4 + 7x^3 - 5x - 8$.

נתונה פונקציה $q(2)$ המאפשרת בדיקה של השאלה.