מבוא לתכנות מערכות 10010 סמסטר א' – תשע"ט

תרגיל בית מס' 4

נושא התרגיל: קבצים בינארים, מצביעים לפונקציה, סיביות, פרמטר ל main, Variadic_function

יש להגיש אך ורק דרך תפריט המטלות שבאתר הקורס, כפי שהוסבר בתרגול .

ניתן לעבוד בזוגות

הנחיות הגשה כלליות:

- התרגיל ייבדק בסביבת Eclipse תחת מערכת הפעלה.Ubuntu
- הקוד חייב לעבור קומפילציה, קוד שאינו מתקמפל לא ייבדק
- **יש להגיש את כל תיקיית הפרויקט הכולל את הקובץ project. ,** מקובצים לקובץ zip/rar ששמו כשם הסטודנט. שם פרטי ומשפחה. **במידה ומגישים בזוגות שם בקובץ יהיה כשם שני האנשים המגישים ושניהם צריכים לעלות את העבודה למודל.**

יש להקפיד על כללי הנדסת התוכנה:

- מבנה התכנית (הזחות) ותיעוד במידת הצורך .
- חובה להשתמש בקבועים במקומות המתאימים .
- יש להשתמש בפונקציות בצורה נכונה לשמירה על סדר, קריאות ושימושיות כמו שנלמד.
- יש להקפיד על בדיקת תקינות קלט .
- הפלט צריך להיות כפי שניתן בתרגיל.
- קוד קצר, לא מסורבל ויעיל הן מבחינת כתיבתו והן מבחינת ריצת התוכנית.

נמשיך מפתרון תרגיל בית 3 המפורסם תוך הוספת יכולות נוספות

קבצים בינאריים, פרמטר ל main מניפולציה של Bits

1. אפשר קריאה של נתוני העיר ושמירה של נתונים אילו לקובץ בינארי. סדר הנתונים יהיה זהה לסדר בקובץ הטקסט בתרגיל 3. שם הגן שהוא מצביע שמור בקובץ בפורמט הבא: מספר המציין כמה תווים יש בשם כולל ה '\0' ולאחר מכן התווים עצמם.
   ההחלטה האם עובדים עם קובץ בינארי או טקסט תהיה לפי פרמטר שיינתן לפונקציית main. (1 מציין קריאה בינארית 0 text)
   כיוון שעדין ניתן גם להשתמש באופציה של קובץ טקסט בנה את הפונקציות כך שיהיה כמה שפחות שכפול קוד.
   לא ניתן לשנות את המבנים הקיימים אן ליצור חדשים
   בקובץ הבינארי נדחוס את האינפורמציה באופן הבא:
   - נתוני מספר הילדים וסוג הגן יהיו ב Byte בודד: מקסימום מס' הילדים בגן הוא 40 (6 bits) ויש 3 סוגים שונים של גן (2 bits) נתון סוג הגן יהיה ב bit הנמוכים.
   - נתוני ילד 2 Bytes: גיל הילד עד 7, (3 bits), ת"ז שלו – נניח עד 8191 (13 bits). נתון ת:ז יהיה ב bit הנמוכים.

פונקציות גנריות מיונים

2. שנה את פונקציית המיון insertionSort הנתונה כך שהיא תהיה גנרית, כלומר היא תוכל למיין כל סוג של מערך.

```
void insertionSort(int* arr, int size)
{
	int i,j,key;
	for (i = 1; i < size; i++)
	{
		key = arr[i];
		for (j = i - 1; j >= 0 && (arr[j] > key); j--)
			arr[j+1] = arr[j];
		arr[j + 1] = key;
	}
}
```

3. אפשר את המיונים הבאים תוך שימוש ביכולות גנריות:
   a. מיון גני הילדים לפי שמם
   b. מיון ילדים בגן לפי ת"ז
   c. מיון גני הילדים לפי סוג הגן ומיון משני לפי מספר הילדים בכל גן.

הוסף לתפריט אפשרות המאפשרת לבדוק את שלושת האופציות

פונקציות qsort & bsearch

4. שנה את הפונקציה `findChildById` כך שתדאג שהחיפוש יהיה יעיל, כלומר מיון ולאחריו חיפוש בינארי. לצורך זה השתמש בפונקציות הספרייה qsort ו bsearch **(לא בפונקציות מסעיף 2 למרות שכמובן הגיוני)**

Variadic Function

5. הכן פונקציה כללית המקבלת רשימה באורך לא ידוע של זוגות פרמטרים: מחרוזת ומספר. הרשימה מסתיימת במחרוזת NULL , הפונקציה מדפיסה את הזוגות. כדי לבדוק את הפונקציה בבחירה התפריט באופציה "Show all Kindergartens " במידה ויש 3 גנים ומעלה, קרא **גם** לפונקציה מהסעיף הקודם עם הנתונים של שם הגן ומספר הילדים בגן עבור 3 הגנים הראשונים בעיר.

רשימות מקושרות גנריות

6. כתוב פונקציה createLinkedListForKindergartenType הבונה רשימה מקושרת של כל הגנים בעיר מסוג מסוים. הפונקציה מקבלת שני פרמטרים – עיר וסוג הגן. הפונקציה מחזירה מצביע לרשימה גנרית שנבנתה.

כתוב פונקציה displayKindergartensFroList שמקבלת רשימה מקושרת גנרית ומדפיסה נתונים של כל איבר ברשימה בידיעה שזאת רשימת גנים.

הוסף לתפריט אפשרות הקוראת לפונקציה kindergartensLinkedList המאפשרת לבדוק את שתי הפונקציות יחד.

בהצלחה.