

מבוא למדעי המחשב

מטלה 5

פוינטרים והקצאות דינאמיות

תאריך אחרון להגשה: 02.01.2020

מתרגל אחראי: מתן דוגוט

שאלה 1:

`char* PairSortedArrays(char a[], char b[])`

כתבו פונקציה `PairSortedArray` אשר מקבלת שתי מחרוזות `a` ו-`b` ממוינות בסדר תווים עולה (לפי טבלת ASCII), ומחזירה מחרוזת אחת אשר תכיל את כלל התווים משתי המחרוזות בסדר עולה. נדרש להשתמש בהקצאות דינאמיות על מנת שהמחרוזת אותה נחזיר תהייה בגודל מדויק.

דוגמאות:

a: acdty

b: berz

result: abcdertyz

a: acdty

b: berz

result: abcdertyz

שאלה 2:

`char** LowerSTR(char* arr[], int size_arr, char* str, int* size_res)`

כתבו פונקציה `LowerSTR` אשר מקבלת מערך של מחרוזות, את גודלו, ומחרוזת `str` נוספת. הפונקציה תחזיר מערך של מחרוזות הקטנות לקסיקוגרפית למחרוזת `str`. במידה ואין כאלה יוחזר `NULL`. גודל המערך יוחזר בעזרת משתנה שנשלח `by address` לפונקציה (`size_res`).

דוגמאות:

Arr[]: hello hola bonjour yasou namaste salam

Str: shalom

Result: hello hola bonjour namaste salam

Str: baba

Result: NULL

שאלה 3:

char* AddNumbers(char* num1, char* num2)

כתבו פונקציה AddNumbers אשר מקבלת שתי מחרוזות המייצגות מספר שלם ארוך (ניתן להניח כי המחרוזות תקינות). הפונקציה תחזיר את תוצאת החיבור של שני המספרים בתוך מחרוזת. ניתן להניח כי המספרים אינם מתחילים ב-0. נדרש להשתמש בהקצאות דינאמיות על מנת שהמחרוזת אותה נחזיר תהייה בגודל מדויק ושאינה תתחיל באפסים מצד שמאל.

דוגמאות:

Num1: 123,456,789

Num2: 987,654,321

Result: 1,111,111,110

Num1: 12,587,749

Num2: 58,965,478,528

Result: 58,978,066,277

שאלה 4:

int* CreateRange(int start, int end, int jump, int* sizeRage)

כתבו את הפונקציה CreateRange המקבלת 3 מספרים שלמים המהווים מספר מינימלי, מספר מקסימלי, ומספר שלם המהווה הפרש. הפונקציה תחזיר מצביע לתחילת מערך של מספרים שלמים כאשר האיבר הראשון הוא המינימלי ושאר האיברים הינם בקפיצות של ההפרש, עד הגעה למספר המקסימלי או למספר הקרוב אליו ביותר שקטן ממנו. את כמות המספרים המאוחסנים יש להחזיר במשתנה sizeRange אשר נשלח by address. במידה ולא ניתן ליצור את הערכים, ערכי המצביעים שחוזרים יהיו NULL.

לדוגמה:

Min: 1 Max: 11 Jump: 3

Result: 1, 4, 7, 10 sizeRange:4

Min: 7 Max: 11 Jump: 8

Result: 7 sizeRange:1

שאלה 5!

מספרים
מיון
מספרים
מיון
מספרים
מיון
מספרים
מיון
מספרים
מיון

int** GradeStat(int* Grades, int size_grades, int grd_range, int* count_grd, int* avg_grd)

כתבו פונקציה GradeStat אשר מקבלת מערך ציונים, את גודלו, מספר שלם המציין טווח ציונים, מערך מונים ריק של מספרים שלמים ומערך ממוצעים ריק של מספרים ממשיים. הפונקציה תבצע חישובים סטטיסטיים על הנתונים באופן הבא:

1. הפונקציה תפצל את הציונים לתתי קבוצות אשר תלויות בטווח הנתון. למשל, אם הטווח הוא 10, התחומים יהיו: 0-9, 10-19, 20-29, 30-39, 90-99, 100. לעומת זאת, אם הטווח הוא 7, התחומים יהיו: 0-6, 7-13, 14-20, 91-97, 98-100. שימו לב שאם הטווח הנתון הוא grd_range אז מספר תתי הקבוצות שיווצרו יהיה: $100/\text{grd_range} + 1$.
2. הפונקציה תחזיר מערך מצביעים למערכים, כשאר כל אינדקס במערך מייצג תחום שונה לפי סדר עולה. כל תא במערך המצביעים יצביע למערך של ציונים הנכללים באותו תחום אשר הוא מייצג. במידה ואין ערכים בטווח של האינדקס, יהיה ערך NULL.

למשל, כאשר $10 = \text{grd_range}$ ובהינתן מערך ציונים:

93	66	84	100	69	45	85	80
----	----	----	-----	----	----	----	----

יוחזור:

100	90-99	80-89	20-29	10-19	0-9
-----	-------	-------	------	----	------	------	-------	-------	-----

93

84	85	80
----	----	----

29

3. הפונקציה תעדכן את מערך המונים ומערך הממוצעים לפי הכמות של הציונים והממוצע של

הציונים לפי אינדקס בהתאמה. ניתן להניח שמערך המונים והממוצעים הוא בדיוק בגודל

$100/\text{grd_range} + 1$.

מערך המונים הינו מערך אשר בכל תא בו מצוין מספר המגדיר כמה מספרים ממערך הציונים

רלוונטיים לתא הנוכחי.

מערך הממוצעים מכיל בכל תא את ממוצע הציונים ממערך הציונים הרלוונטיים לתא לפי

הטווח של התא.

למשל, לפי הדוגמה הקודמת:

המונים יהיו:

1	1	3	0	2	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---

הממוצעים יהיו:

100	93	83	0	67	0	45	0	0	0	0
-----	----	----	---	----	---	----	---	---	---	---