מבוא לתכנות מערכות 10010 סמסטר א' – תשע"ט

תרגיל בית מס' 2

נושא התרגיל: מצביעים ומחרוזות

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 רק את קבצי הקוד (\*.c \*.h), 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.

נתונה הקוד של קובץ main.c, העתק אותו לפרויקט והשלם את הפונקציות כפי שמתואר בהמשך.

**#define** NEW\_STRING 1

**#define** CURRENT\_STRING 2

**#define** COUNT\_WORDS 3

**#define** LONGEST\_WORD 4

**#define** REVERT\_WORD 5

**#define** CHECK\_ERASE\_CHARS 6

**#define** CHECK\_PALINDROME 7

**#define** EXIT -1

**#define** MAX\_LEN 100

void printMenu() {

printf("\n\n");

printf("Please choose one of the following options\n");

printf("%d - Enter new string\n", NEW\_STRING);

printf("%d - Print current string\n", CURRENT\_STRING);

printf("%d - Count words in string\n", COUNT\_WORDS);

printf("%d - Print longest word in string CAPITALIZED\n", LONGEST\_WORD);

printf("%d - Revert each word in string\n", REVERT\_WORD);

printf("%d - Check eraseCharsFromStr\n", CHECK\_ERASE\_CHARS);

printf("%d - Check palindrome\n", CHECK\_PALINDROME);

printf("%d - Exit\n", EXIT);

}

void performChoice(int option, char \*str) {

switch (option)

{

case NEW\_STRING:

initString(str, MAX\_LEN);

break;

case CURRENT\_STRING:

printString(str);

break;

case COUNT\_WORDS:

{

int count = countWords(str);

printf("There are %d words in string\n", count);

break;

}

case LONGEST\_WORD:

longestInCaptital(str);

printf("String after longest capital %s\n", str);

break;

case REVERT\_WORD:

revertWords(str);

printf("String after revert %s\n", str);

break;

case CHECK\_ERASE\_CHARS:

eraseCharsFromString(str, ":, ?!-;");

printf("String after erase %s\n", str);

break;

case CHECK\_PALINDROME:

if (isPalindrome(str))

printf("String %s is palindrome\n", str);

else

printf("String %s is NOT palindrome\n", str);

break;

case EXIT:

printf("Bye bye\n");

break;

default:

printf("Wrong option\n");

break;

}

}

**int** **main**()

{

**int** option;

**char** str[MAX\_LEN] = "\0";

**do** {

printMenu();

**scanf**("%d",&option);

performChoice(option, str);

} **while**(option != EXIT);

**return** EXIT\_SUCCESS;

}

תאור הפונקציות השונות:

1. **initString** – פונקציה המאתחלת מחרוזת באורך מקסימאלי MAX\_LEN, נניח 100.

### המחרוזת יכולה להיות מורכבת מכמה מילים, מסימנים, מספרות ורווחים. אין צורך לבדוק תקינות המחרוזת.

### נגדיר תו מיוחד כאחד מהתווים הבאים: רווח , :? -

### נגדיר מילה כרצף אותיות ללא התווים המיוחדים

### יתכן שבטקסט הנקלט מופיע יותר מתוו מיוחד בין המילים.

### יתכן שהמחרוזת מתחילה ו/או מסתיימת בתו מיוחד (או כמה).

### יתכן גם שהמחרוזת אינה מכילה מילים כלל.

### **printString** הדפסת המחרוזת.

### **countWords** – פונקציה המחזירה כמה מילים יש במחרוזת, ללא שינוי המחרוזת המקורית.

### דוגמא: עבור המחרוזת:

### " Fake news: What exactly is it - and how can you spot it**?"**

התוכנית תדפיס: 12 .

### **longestInCapital** – שינוי המחרוזת כך שהמילה הארוכה ביותר תהיה באותיות גדולות.

דוגמא: עבור המחרוזת:

### " Fake news: What exactly is it - and how can you spot it**?"**

המחרוזת החדשה תהיה:

Fake news: What **EXACTLY** is it - and how can you spot it**?”"**

### **revertWords** פונקציה ההופכת כל מילה בטקסט ומחליפה את כל התווים המיוחדים בכוכבית \*

דוגמא: עבור המחרוזת:

### " Fake news: What exactly is it - and how can you spot it**?"**

התוכנית תדפיס (ללא שינוי סדר המילים):

### "**\*\*\*\***ekaF\*swen**\*\***tahW**\***yltcaxe**\***si**\***ti**\*\*\***dna\*who**\***nac\*uoy\*tops**\***ti?**"**

1. כתוב פונקציה:

**void eraseCharsFromString (char \*str, const char \*symbols);**

הפונקציה תבצע הורדה ממחרוזת **str** של כל התווים שנמצאים במחרוזת **symbols** . יש לבצע **בסיבוכיות של O(n)**

דוגמה עבור:

**str = "EU won't offer 'better deal' on Brexit!” symbols = "E'!”**

הפונקציה תשנה את המחרוזת **str**

**str = "U wont offer better deal on Brexit”**

1. כתוב פונקציה:

I**nt isPolyndrome(const char \*str)**

הפונקציה בודקת האם המחרוזת **str** היא פלינדרום (palindrome) , כלומר מתקבלת אותה מחרוזת אם לעבור מההתחלה עד הסוף או מהסוף להתחלה. ההשוואה מתבצעת מהתו הראשון עד לתו אחד לפני תו סוף המחרוזת `**\n**` . הפונקציה לא מבדילה בין אותיות קטנות וגדולות ולא לוקחת בחשבון את התווים שאינם אותיות, כמו מספרים, רווחים, פסיקים, סימני קריאה, סימני שאלה, נקודותיים וכו'.

לדוגמה, הפונקציה תחזיר אמת עבור המחרוזות

**"N ever od?d o?r ev en"**

**"Was it a car o5r a ca7t I saw?"**

ושקר עבור המחרוזת

**“Madam, I am Adam”**

**אילוצים:** בפונקציה אסור להשתמש במשתנים מסוג **int**.

*בהצלחה*