

סוג הבחינה: מבחן לדוגמה.  
מועד הבחינה: סמסטר א', תשפ"ג 2023  
מחבר: שליו בן דוד.

**שים לב:** בבחינה זו יש הנחיות מיוחדות.  
יש לענות על השאלות על פי הנחיות אלה.

## תכנות מערכות

### הוראות לנבחן

(א) משך הבחינה: 2 שעות.

(ב) מבנה השאלון:  
שאלה 1 - קטע קוד.  
שאלה 2 - קבצים.  
שאלה 3 - מחרוזות.  
שאלה 4 - מבנים.

(ג) חומר עזר מותר בשימוש: אין להשתמש בכל חומר עזר מלבד מחשבון מדעי שאינו גרפי.

(ד) הנחיות: יש לפתור את כל השאלות. יש להסביר בקצרה במידה הצורך במקרה של טעות קטנה או שגיאת חישוב.

כתוב במחברת הבחינה בלבד. רשום "טיוטה" בראש כל עמוד המשמש טיוטה.  
כתיבת טיוטה בדפים שאינם במחברת הבחינה עלולה לגרום לפסילת הבחינה.

**ההנחיות בשאלון זה מנוסחות בלשון זכר ומכוונות לנבחנות ולנבחנים כאחד.**

**בהצלחה!**

## השאלות

1) נתון קטע הקוד הבא:

```
#define A 1

int foo (char *str1) {
    char *str2 = str1;
    if (A)
        str1[0] = 's';
    else
        str1[0] = '\0';
    while(*str1) {str1++;}
    return (str1 - str2);
}

int main() {
    char *str = "Shalev";
    #define A 0
    printf("%d", foo(str));
    return 0;
}
```

האם קיימת שגיאה בתוכנית? אם כן הסבירו מהי בקצרה וכתבו תיקון לבעיה.

מה הפונקציה תדפיס בסוף הריצה?

2) **א)** כתבו פונקציה `printLastN` שמקבלת מצביע לקובץ בעל הרשאות כתיבה וקריאה

ומספר שלם  $n$  ומדפיסה את  $n$  האותיות האחרונות בקובץ.

. ניתן להניח כי הקובץ מכיל לפחות  $n$  תווים.

**ב)** כתבו פונקציה `Upper` שמקבלת מצביע לקובץ בעל הרשאות כתיבה וקריאה

והופכת את כל האותיות בו לאותיות גדולות.

3) **מחרוזת תיקרא בר-פולינדרומית** אם קיימת דרך לשנות את סדר המחרוזת ולקבל פולינדרום.

לדוגמה: `ababc` היא בר פולינדרומית שכן ניתן לקבל ממנה את `abcba`.

כתבו פונקציה `BarPolindrom` שמקבלת מחרוזת `str` ומדפיסה האם היא בר פולינדרומית.

. ניתן להניח כי המחרוזת מורכבת מותיות קטנות בלבד וללא רווחים.

(4

בתרגיל זה נייצג קבוצה (Set) באמצעות מבנים.

**(א)** כתבו Struct שמייצג קבוצה.

על המבנה להכיל מצביע לאוסף של מספרים שלמים ומספר שלם שמייצג את גודל הקבוצה.

**(ב)** כתבו פונקציה שמאתחלת קבוצה ריקה ומחזירה מצביע לקבוצה.

**(ג)** כתבו פונקציה שמקבלת איבר ומצביע לקבוצה ומוסיפה אותו לקבוצה.

שימו לב כי בקבוצה אין חשיבות לחזרות.

**(ד)** כתבו פונקציה שמקבלת 2 קבוצות  $A$  ו- $B$  ומחזירה 1 אם  $A \subseteq B$  ו-0 אחרת.

שימו לב כי בקבוצה אין חשיבות לסדר.

**(ה)** כתבו פונקציה שמקבלת 2 קבוצות  $A$  ו- $B$  ומדפיסה אם  $A = B$ .

**בהצלחה!**

## התשובות

הבהרה: פיתרון זה של המבחן לדוגמה אינו אלא הצעה בלבד. ייתכנו כמה אפשרויות פיתון לשאלות, ופיתרון זה אינו סופי. במהלך הבדיקה, ייתכן שייעשו שינויים או הרחבות באפשרויות התשובה שיתקבלו.

**1.** קיימת שגיאה בתוכנית. הפוינטר str מצביע למחרוזת שנמצאת באזור הקבועים, ולכן ניתנת לקריאה בלבד. ניתן לשנות את שורה ההצהרה להיות:

```
char str[] = "Shalev";
```

במקרה זה, הפונקציה תדפיס 6 (ה-define שהוגדר באופן גלובלי הוא העדכני ביותר לבלוק של foo).

**2. א)** הקוד להלן: (אנו משתמשים ב-calloc ולכן אין צורך להכניס בסוף '\0').

```
void printLastN(FILE* fpt, int n) {
    fseek(fpt, -1, SEEK_END);
    char* str = (char*) calloc ((n + 1), sizeof (char));
    if (!str) {exit(1);}
    for (int i = 0; i < n; i++) {
        fscanf(fpt, "%c", &str[n - i - 1]);
        fseek(fpt, -2, SEEK_CUR);
    }
    printf("%s", str);
    free(str);
}
```

**ב)** הקוד להלן:

```
void Upper (FILE* fpt) {
    fseek(fpt, 0, SEEK_SET);
    while (!feof(fpt)) {
        int c = fgetc(fpt);
        if (c >= 'a' && c <= 'z') {
            c = c - 'a' + 'A';
            fseek(fpt, -1, SEEK_CUR);
            fputc(c, fpt);
        }
    }
}
```

**3. אבחנה:** מחרוזת  $w$  היא לא בר פולינדרומית  $\iff$  קיימים לפחות 2 אותיות שכמות המופעים שלהן ב- $w$  היא אי זוגית.  
 מדוע? אותיות שמופיעות כמות זוגית של פעמים ניתן לדחוף ל-2 הצדדים בהתאמה, ואם מופיעה גם בנוסף אות אחת שמופיעה כמות אי זוגית של פעמים ניתן לדחוף אותה באמצע.

הקוד להלן:

```
void BarPolindrom (char* str) {
    char letters['z'-'a' + 1] = {0};
    for (int i = 0; i < strlen(str); i++)
        letters[str[i] - 'a']++;
    int countOdd = 0;
    for (int i = 0; i < 'z'-'a'+1; i++)
        if (letters[i] % 2)
            countOdd++;
    if (countOdd >= 2)
        printf("Not Bar Polindrom\n");
    else
        printf("Bar Polindrom\n");
}
```

**4. א)** נכתוב את ה-struct:

```
typedef struct Group_ {
    int* set;
    int size;
} group, *pgroup;
```

**ב)** נכתוב פונקציית אתחול:

```
pgroup init_group () {
    pgroup A = (pgroup) malloc (sizeof (group));
    if (!A) {exit(1);}
    A -> set = NULL;
    A -> size = 0;
    return A;
}
```

**ג)** נכתוב פונקציית הוספת איבר לקבוצה:

```
void add_item (pgroup A, int item) {
    if (!A) {exit(1);}
    if (!(A -> set)) { // if A is empty.
        A -> set = (int *) malloc(sizeof(int));
        if (!(A -> set)) {
            free(A);
            exit(1);
        }
        A -> set[0] = item;
        A -> size++;
    }
}
```

המשך קוד בעמוד הבא (המקרה שהקבוצה לא ריקה):

```

else {
    for (int i = 0; i < A->size; i++)
        if (A->set[i] == item)
            return;
    A->set = (int*) realloc(A->set, (A->size++) * sizeof(int));
    if (!(A->set)) {
        free(A->set);
        free(A);
        exit(1);
    }
    A->set[A->size-1] = item;
}

```

ד) נכתוב פונקצייה שבודקת הכלה. נבחין כי אם  $A = \emptyset$  אז ההכלה טריוויאלית ונחזיר 1. אחרת, נכתוב פונקציית עזר שבודקת אם איבר נמצא ב-B ונריץ אותה עם כל איבר ב-A. אם נמצא איבר ששייך ל-A לא נמצא ב-B נחזיר מיידית 0.

```

int hasX (pgroup B, int X) {
    int ans = 0;
    for (int i = 0; i < B->size; i++)
        if (B->set[i] == X)
            ans = 1;
    return ans;
}

```

```

int isSubset (pgroup A, pgroup B) {
    if (!A || !B) {exit(1);}
    if (!(A->set)) {return 1;} // the empty set is a subset of all sets.
    for (int i = 0; i < A->size; i++) {
        if (!hasX(B, A->set[i]))
            return 0;
    }
    return 1;
}

```

ה) נכתוב פונקצייה שבודקת אם  $A = B$ . אבחנה:  $A = B \iff A \subseteq B \text{ וגם } B \subseteq A$ . לכן ניעזר בסעיף קודם.

```

Void isEqual (pgroup A, pgroup B) {
    if (isSubset(A, B) * isSubset(B, A))
        printf("Are equal");
    else
        printf("Are not equal");
}

```