

Exam number: _____ מספר בחינה:

I.D. Number: _____ מספר ת"ז:

Interdisciplinary Center Herzliya המרכז הבינתחומי הרצליה
Efi Arazi School of Computer Science בית ספר אפי ארזי למדעי המחשב
Spring semester – MOED GIMMEL, 2020 מוסטר אביב – מועד ג', 2020

Final Exam – System Programming in C מבחן מסכם – תכנות מערכות בשפת C

Lecturers: Ilan Gronau and Sara H. Geizhals

מרצים: אילן גרונאו, שרה גייצהאלס

Date: November 10, 2020

תאריך: 10 בנובמבר, 2020

Duration: 3 hours. No extensions will be given – plan your time wisely!

Material: Any written or typed material is allowed

Instructions: All answers should be written **in the solution file only**

משך הבחינה: 3 שעות. לא תינתן הארכת זמן – תכננו את זמנכם בקפידה!

חומר עזר: כל חומר כתוב או מודפס מותר

הנחיות: יש לענות על כל השאלות **על גבי תבנית הפתרון בלבד**

→ **Do not submit any draft notebooks**

← **אין להגיש מחברות טיוטא**

Question 1	/15	שאלה 1
Question 2	/30	שאלה 2
Question 3	/20	שאלה 3
Question 4	/35	שאלה 4
Final grade	/100	ציון סופי

Good luck!

בהצלחה!

Question 1 (15 points):

שאלה 1 (15 נקודות):

- ▶ The current directory contains exactly two files named `words.txt` and `myprog.sh`. Their contents are specified below.

התיקה הנוכחית מכילה בדיוק שני קבצים בשם `words.txt` ו `myprog.sh`. תוכן הקבצים מפורט למטה. ◀

`words.txt:`

```
Dear sir,  
  
My name is Abraham.  
How are you?  
I will abbreviate my message.  
I saw a bridge yesterday.  
The bridge was a broken.  
  
Sincerely,  
Abraham
```

`myprog.sh:`

```
#!/bin/bash  
  
if ! [ -e $1 ]; then  
    echo "Error with $0 $*" >> /dev/stderr  
elif [ $2 == "-a" ]; then  
    grep '[Aa]b*r' $1  
elif [ $2 == "-b" ]; then  
    sed -r 's/[[:upper:]]*[[:lower:]]*/XX/' $1  
elif [ $2 == "-c" ]; then  
    ((numLines=end-start+1))  
    tail -n+$start $1 | head -n$numLines  
fi
```

- ▶ Specify the expected output for each of the following three commands. Briefly explain your answer.

כתבו את הפלט הצפוי עבור כל אחת משלשת הפקודות הבאות. הסבירו בקצרה את תשובתכם. ◀

a) (3 pts)

(נק' 3) א

```
==> bash myprog.sh words.txt -a
```

Output:

```
Dear sir,  
My name is Abraham.  
How are you?  
I will abbreviate my message.  
Abraham
```

Explanation: words.txt is an existing file, so the if condition is not satisfied. The second parameter (\$2) is -a, so first else if is satisfied. The pattern in the grep command matches strings that start with 'a' or 'A', end with 'r' and in between have any number of (even zero) 'b's. The strings in the file that match this pattern are: ar (in Dear and are), Abr (in Abraham), and abbr (in abbreviate). The lines that contain these strings are printed.

b) (3 pts)

(נק' 3) ב

```
==> bash myprog.sh -a word.txt
```

Output:

```
Error with myprog.sh -a words.txt
```

Explanation: -a is not an existing path (file or directory), so the if condition is satisfied, and the error message is printed. Note that \$0 refers to the called program and \$* refers to all (other) parameters.

c) (4 pts)

(נק' 4) ג

```
==> bash myprog.sh word.txt -b
```

Output:

```
XX sir,  
XX  
XX name is Abraham.  
XX are you?  
XX will abbreviate my message.  
XX saw a bridge yesterday.  
XX bridge was a broken.  
XX  
XX,  
XX
```

Explanation: words.txt is an existing file, so the if condition is not satisfied. The second parameter (\$2) is -b, so the second else if condition is satisfied. The pattern in the sed command matches the first string in each line that starts with a sequence of upper-case letters, followed by a sequence of lower-case letters (both sequences can be empty). This string is then replaced by XX. Note that every line contains such a string (could be the empty string), and the first such string is the one that is matched and replaced (due to regexp's greedy matching policy, the matched string is the longest among those starting first).

- d) (5 pts) Write a sequence of three commands that result in printing lines 3-7 of file `words.txt` on to the screen. The first two commands should assign values to bash variable, and the third command should call `myprog.sh`, such that it prints lines 3-7 of file `words.txt`. The commands are executed one after the other in sequence (not piped). Briefly explain your answer.

(ד) (5 נק') כתבו רצף של שלוש פקודות שבסופו שורות 3-7 של קובץ `words.txt` מודפסות למסך. שתי הפקודות הראשונות צריכות להציב ערכים במשתני `bash`, והפקודה השלישית אמורה לקרוא ל `myprog.sh`, כך שהוא מדפיס את שורות 3-7 של קובץ `words.txt`. הפקודות נקראות ברצף אחת אחרי השניה (לא משורשרות). הסבירו את תשובתכם בקצרה.

Sequence of commands:

```
==> start=3
==> end=7
==> source myprog.sh words.txt -c
```

Explanation: We set the bash variables `$start` and `$end` to the start and end lines that we wish to be printed, and then execute the script using 'source' and the option `-c`. Executing the script with 'source' means that the script has access to the variables defined in the calling shell (`$start` and `$end`). The second parameter (`$2`) is `-c`, so the third else if condition is satisfied. The variable `$numLines` is set to be 5 ($= \$end - \$start + 1$), which is the number of lines that we wish to be printed. The `tail` and `head` commands result in lines 3-7 being printed.

Question 2 (30 points):

שאלה 2 (30 נקודות):

Consider the following program:

התבוננו בתוכנית הבאה:

```
#include <stdlib.h>
int var;

int main() {
    int i=3, j=0, k;
    double x=1.2, y=2.0;
    int arr1[4] = {1,2,3,4};
    char arr2[10] = "hello!";
    char* str1;
    char* str2 = malloc(10);
    if(!str2) return 1;

    /** vvv YOUR EXPRESSIONS BELOW THIS POINT vvv ***/

    /** ^^^ YOUR EXPRESSIONS ABOVE THIS POINT ^^^ ***/
    free(str2);
    return 0;
}
```

For 10 of the following 11 scenarios write a line of code that satisfies the requirements of the scenario. Make sure to follow these instructions:

- Your expressions should contain only variables and operators; it should not contain constant literals. Examples of constant literals: 4 (for **int**), 5.0 (for **double**), 'A' (for **char**), and "A" (for **char***).
- Assume that the line of code is written between the two comment lines in the program.
- Each scenario should be considered separately, ignoring the expressions you suggested for other scenarios.
- Keep one scenario unanswered. If you answer all scenarios, then only the first ten will be checked.
- Briefly explain how your line of code satisfies the scenario requirements.

עבור 10 מתוך 11 התרחישים למטה כתבו שורת קוד שעומדת בדרישות של התרחיש. שימו לב לעמוד בהנחיות הבאות:

- הביטויים שלכם צריכים להכיל רק משתנים ואופרטורים, ללא ליטרלים קבועים. דוגמאות לליטרלים קבועים: 4 (עבור **int**), 5.0 (עבור **double**), 'A' (עבור **char**), ו "A" (עבור **char***).
- הניחו ששורת הקוד נכתבת בין שתי שורות ההערה שבתוכנית.
- התייחסו לכל תרחיש בנפרד, והתעלמו מהביטויים שכתבתם עבור תרחישים אחרים.
- שמרו על תרחיש אחד ללא מענה. אם תענו על כל התרחישים, אז רק העשרה הראשונים ייבדקו.
- הסבירו בקצרה מדוע שורת הקוד שלכם עומדת בדרישות התרחיש.

Example scenario: An expression that consists of two variables and an arithmetic operator and evaluates to 6.0.

תרחיש לדוגמא: ביטוי שמכיל שני משתנים ואופרטור אריתמטי וערכו 6.0.

Solution:

פתרון:

Expression: `y*i;`

Explanation: `y==2.0 ; i==3 → y*i==6.0.`

*Note that `y*2.0` is a wrong answer, because it contains the literal 2.0.

*שימו לב ש `y*2.0` היא תשובה שגויה כי היא מכילה את הליטרל 2.0.

Scenarios: (3 pts each)

a) An expression that consists of two variables and one of the logical operators `&&` or `||`, and the expression evaluates to 0.

א) ביטוי שמורכב משני משתנים ואחד מהאופרטורים הלוגיים `&&` או `||`, כאשר הביטוי מוערך ל 0.

Expression: `i&&j;`

Explanation: `i==3 ; j==0 → i && j == TRUE AND FALSE == FALSE == 0.`

b) An expression that consists of two variables and one of the logical operators `&&` or `||`, and the expression evaluates to 1.

ב) ביטוי שמורכב משני משתנים ואחד מהאופרטורים הלוגיים `&&` או `||`, כאשר הביטוי מוערך ל 1.

Expression: `i||j;`

Explanation: `i==3 ; j==0 → i || j == TRUE OR FALSE == TRUE == 1.`

c) An expression that contains a logical operator, which is short circuited, such that that the second part of the operation is not evaluated.

ג) ביטוי שמכיל אופרטור לוגי, ועובר קצר (short circuit), כך שהחלק השני של הביטוי לא מוערך.

Expression: `i||j;`

Explanation: `i==3 ; j==0 → the first part of i||j is evaluated to TRUE, which implies that the entire expression is evaluated to TRUE regardless of the second part.`

- d) An expression that contains one type conversion triggered by an assignment. The expression should not trigger additional type conversions other than this one.

ד) ביטוי שמכיל המרת ערך אחת שנגרמת מהשמה. הביטוי לא אמור לגרום להמרת ערך נוספת מעבר להמרה הזו.

Expression: `y=i;`

Explanation: `i==3` is int, and `y` is a double variable, so 3 is converted to 3.0.

- e) An expression that contains a type conversion triggered by a casting operator. The conversion should be a demotion (from a "higher ranked" type to a "lower ranked" type) but results in no change of value. The expression should not trigger additional type conversions other than this one.

ה) ביטוי שמכיל המרת ערך אחת שנגרמת מהמרה מפורשת (casting). ההמרה צריכה להמיר טיפוס "גבוה" לטיפוס "נמוך" (demotion), אבל ללא שינוי הערך המיוצג. הביטוי לא אמור לגרום להמרת ערך נוספת מעבר להמרה הזו.

Expression: `i=(int)y;`

Explanation: `y==2.0` so the casting operator (int) converts it to int 2, which represents the same value.

- f) An expression that contains a type conversion triggered by a casting operator. The conversion should be a demotion (from a "higher ranked" type to a "lower ranked" type), and results in a change of value. The expression should not trigger additional type conversions other than this one.

ו) ביטוי שמכיל המרת ערך אחת שנגרמת מהמרה מפורשת (casting). ההמרה צריכה להמיר טיפוס "גבוה" לטיפוס "נמוך" (demotion), עם שינוי הערך המיוצג. הביטוי לא אמור לגרום להמרת ערך נוספת מעבר להמרה הזו.

Expression: `i=(int)x;`

Explanation: `x==1.2` so the casting operator (int) converts it to int 1, which represents a different value.

- g) An expression that contains two type conversions: one triggered by an arithmetic operation and one triggered by an assignment.

ז) ביטוי שמכיל שתי המרות טיפוסים: אחת הנגרמת מפעולה אריתמטית ואחת הנגרמת מהשמה.

Expression: `i = x + j;`

Explanation: `x==1.2` and `j=0`, so the arithmetic operator (+) converts `j` to double `0.0`. The result of the operation (`1.2 = 1.2+0.0`) gets converted to type `int 1` because of the assignment into an `int` variable (`i`).

- h) An expression that accesses an uninitialized variable (with arbitrary "garbage" value).

ח) ביטוי שעושה שימוש במשתנה שלא אותחל (עם ערך "זבל" שרירותי).

Expression: `i = k;`

Explanation: `k` and `str1` are the only uninitialized variables in this program. Note that the global variable `var` is initialized to `0` by default.

- i) An expression that results in a type mismatch compilation warning.

ט) ביטוי שגורם לאזהרת קומפילציה בגלל חוסר התאמה בטיפוסים.

Expression: `str1 = arr1;`

Explanation: `str1` is a pointer of type `char*`, and this expression tries to assign it with an address of type `int*`. Note that conversions between primitive types (non-pointers) do not result in any warnings.

- j) An expression that will likely result in a segmentation fault.

י) ביטוי שיגרום בסבירות גבוהה ל `segmentation fault`.

Expression: `str1[2];`

Explanation: `str1` is an uninitialized pointer variable, so de-referencing it will likely cause a segmentation fault.

- k) An expression that results in a memory leak.

יא) ביטוי שגורם לדליפת זיכרון.

Expression: `str2 = arr2;`

Explanation: `str2` is the only variable that holds the address of the memory block dynamically allocated by `malloc`. Once we make it point to another address, we will not be able to free the allocated block.

- The file `gan_chovah.txt` contains a list of children in a kindergarten class (גן חובה) in Herzliya. In particular, each line of the file lists the child's first name, last name, and number of times that the child attended the class's daily Zoom session during this school year. Each line in the file is formatted as follows:

`<first><comma><last><comma><number>`

where:

- `<first>` and `<last>` are the first and last names of the child (respectively). They consist only of letters (upper or lower case), the dash character ('-'), or the space character (' ').
- `<number>` is the number of times that the child attended the class's daily Zoom session during this school year. If the child has never attended, then this field is kept blank. Also, being that so far there have been only 52 school days so far, then it is guaranteed that `<number>` consists of at most two digits.
- `<comma>` is the comma character (',').
- You may also assume that lines do not start or end with space characters and that there are no blank lines in the file.

◀ הקובץ `gan_chovah.txt` מכיל רשימה של ילדים בכיתת גן חובה בהרצליה. בפרט, כל שורה בקובץ מפרטת את השם הפרטי של הילד, שם המשפחה שלו, ומספר הפעמים שהילד נכח בשיעור הזום היומי במהלך שנת הלימודים. כל שורה בקובץ היא בעלת הפורמט הבא:

`<first><comma><last><comma><number>`

כאשר:

- `<first>` ו `<last>` הם השם הפרטי ושם המשפחה של הילד (בהתאמה). השמות מכילים רק אותיות (גדולות וקטנות), תו המקף ('-') או תו הרווח (' ').
- `<number>` הוא מספר הפעמים שהילד נכח בשיעור הזום היומי במהלך שנת הלימודים. אם הילד לא נכח באף שיעור, אז השדה הזה נשאר ריק. בנוסף, מכיוון שעד כה היו רק 52 ימי לימוד, אז מובטח ששדה ה `<number>` יכיל לכל היותר שתי ספרות.
- `<comma>` הוא תו הפסיק (',').
- אתם רשאים להניח ששורות אינן מתחילות או מסתיימות בתו רווח ושאינן שורות ריקות בקובץ.

Example for the contents of file

gan_chovah.txt

תוכן הקובץ לדוגמא

```
Elad, Kuttav, 27
Yonatan, Har-Zahav, 32
Avigail, Lebzelter, 48
Alon, Ben-Zev,
Ofer, Segalowitch, 12
Chanit, Isaaman, 22
Galia, Okun, 43
Chavatzelet, Fried, 52
Lev, Bar Gal, 5
Yeshaya, Cohen,
Betzalel Oz, Gold, 17
Yavin, Alter, 45
Netanel, Mendenhall, 52
```

- a) (4 pts.) Write a piped sequence of commands, which takes the file `gan_chovah.txt` and creates a file named `never_attended.txt` that contains a list of the first names of children that have never attended the class's daily Zoom session this school year. The names should be listed in the order in which they appear in the original file. After applying the piped sequence to the sample file above, the file `never_attended.txt` should contain these two lines:

```
Alon
Yeshaya
```

א) (4 נק') כתבו רצף פקודות משורשרות שלוקח את הקובץ `gan_chovah.txt` ויוצר קובץ בשם `never_attended.txt`, שמכיל את כל השמות הפרטיים של ילדים שלא נכחו באף אחד ממפגשי הזום היומיים בשנת הלימודים. השמות אמורים להיות רשומים על פי סדר הופעתם בקובץ המקורי. אחרי הפעלת רצף הפקודות המשורשרות על קובץ הדוגמא למעלה, הקובץ `never_attended.txt` צריך להכיל את שתי השורות הבאות:

```
Alon
Yeshaya
```

```
cat gan_chovah.txt | \
  grep -v [[:digit:]] | cut -f1 -d", "
> never_attended.txt
```

The `grep` command matches lines which contain no digits. Then, the `cut` command extracts the first field. The resulting names are put into the appropriate file.

b) (4 pts.) Write a piped sequence of commands that takes the file gan_chovah.txt and prints to the screen a sentence about each child: `<first> is a student in this gan chovah class`

, but where `<first>` is printed with no unnecessary spaces. For example, Betzalel Oz (with three space characters) is modified to Betzalel Oz with a single space character.

ב) (4 נק') כתבו רצף פקודות שלוקח את הקובץ gan_chovah.txt ומדפיס למסך את המשפט הבא לכל ילד:

`<first> is a student in this gan chovah class`

, כאשר `<first>` מודפס ללא רווחים מיותרים. למשל, Betzalel Oz (עם שלושה תווי רווח) מעודכן ל Betzalel Oz עם תו רווח אחד.

```
cat gan_chovah.txt | sed -r 's/(.*) ,.* ,.* /\1 is a student in this gan chovah class/' | tr -s " "
```

The sed command matches the first name and then places it in the context of the sentence "is a student in this gan chovah class." Then the translate command removes additional spaces before the printing of the sentence to the screen.

c) (6 pts.) Write a piped sequence of commands that takes the file gan_chovah.txt and prints to the screen the first and last names of all children whose first name contains the lower case letters 'a' and 'e'. The first and last names of every child should be separated by an asterisk character ('*'). After applying the piped sequence to the sample file from page 7, the following lines should be printed to the screen:

```
Chavatzelet*Fried
Yeshaya*Cohen
Betzalel Oz*Gold
Netanel*Mendenhall
```

ג) (6 נק') כתבו רצף פקודות שלוקח את הקובץ gan_chovah.txt ומדפיס למסך את השם הפרטי ושם המשפחה של כל הילדים שהשם הפרטי שלהם מכיל את האותיות הקטנות 'a' ו 'e'. השם הפרטי ושם המשפחה של כל ילד צריכים להיות מופרדים ע"י תו כוכבית ('*'). אחרי הפעלת רצף הפקודות המפורשות על קובץ הדוגמא בעמוד 7, השורות הבאות צריכות להיות מודפסות למסך:

```
Chavatzelet*Fried
Yeshaya*Cohen
Betzalel Oz*Gold
Netanel*Mendenhall
```

```
cat gan_chovah.txt | grep -E '^[^,]*(a[^,]*e|e[^,]*a)' |  
cut -f1-2 -d"," | tr ", " "*"
```

The extended grep command matches lines containing both a and e before the first ','. Then the cut command isolates the name fields, and the tr command turns the comma into an asterisk.

- d) (6 pts.) Write a piped sequence of commands that takes the file gan_chovah.txt and prints to the screen the last name that is alphabetically last among all children who attended at least 20 daily Zoom sessions. The piped sequence should print a single last name as output. After applying the piped sequence to the sample file from page 7, it should print the name Okun.

(ד) (6 נק') כתבו רצף פקודות שלוקח את הקובץ gan_chovah.txt ומדפיס למסך את שם המשפחה האחרון ע"פ סדר אלפביתי מבין כל הילדים שנכחו בלפחות 20 מפגשי זום יומיים. רצף הפקודות המשורשר צריך לפלוט שם משפחה יחיד למסך. אחרי הפעלת רצף הפקודות המשורשרות על קובץ הדוגמא בעמוד 7, הוא אמור להדפיס למסך את השם Okun.

```
grep [2-9][0-9]$ gan_chovah.txt | cut -f2 -d"," | sort |  
tail -n1
```

The grep command will match lines that end with a two-digit number whose tens digit is at least 2 (≥ 20). The cut command extracts the last name, and the sort and head commands extract the last name alphabetically (also possible to sort in reverse and use head).

Question 4 (35 points):

שאלה 4 (35 נקודות):

- This question deals with representation of binary vectors as strings. A string represents a valid binary vector if all its characters are '0' or '1'. Its value is determined as the non-negative value represented by the binary representation (as in unsigned types). For example, the strings "101", "0101", and "000101" all represent the value 5, and the strings "0000", "0", and "" (the empty string) all represent value 0.

◀ שאלה זו עוסקת בייצוג של וקטורים בינריים כמחרוזות. מחרוזת מייצגת וקטור בינרי תקני אם כל התווים שלה הם '0' או '1'. הערך שלה נקבע על פי הערך האי-שלילי המיוצג ע"י הייצוג הבינרי (כטיפוס ללא סימן). לדוגמא, המחרוזות "101", "0101", ו"000101" כולן מייצגות את הערך 5, והמחרוזות "0000", "0", ו"" (המחרוזת הריקה) כולן מייצגות את הערך 0.

- Implement the five functions specified in (a)-(e) for binary strings.

- There is no need to write documentation.
- You may use functions from the **string.h** library in your implementation. A description of some relevant functions is given on page 23.

◀ ממשו את חמשת הפונקציות המתוארות ב (א)-(ה) עבור מחרוזות בינריות.

- אין צורך בכתיבת תיעוד לקוד.
- במימוש שלכם אתם יכולים להיעזר בקריאות לפונקציות מהספרייה **string.h**. תקציר של מספר פונקציות רלוונטיות נמצא בעמוד 23.

- a) (7 pts.) Implement a function named `numToBinaryString`, which takes as input an integer number (`num`), a character buffer (`buff`), and the buffer length (`buffLen`). It then writes the appropriate binary string inside the buffer.
- The string written in the buffer should contain exactly `buffLen-1` characters ('0' or '1'), based on the binary representation of `num`.
 - If `buffLen ≤ 0`, then the function does nothing and returns 0. Otherwise, it returns 1 after following the instructions below.
 - If `num` can be represented by fewer than `buffLen-1` bits, then the string written in the buffer should contain leading zeros. For example, the call `numToBinaryString(5, buff, 5)` should write the string "0101" into the buffer (with one trailing zero).
 - If `num` cannot be accurately represented by `buffLen-1` bits, then the string written in the buffer should contain the `buffLen-1` least significant (right-most) bits of `num`. For example, the call `numToBinaryString(5, buff, 3)` should write the string "01" into the buffer (dropping the leading left-most '1').
 - Don't forget the terminating '\0' at the end of the string.

Implementation tip: write the string from right to left.

(א) (7 נק') ממשו פונקציה בשם `numToBinaryString`, שמקבלת כקלט מספר שלם (`num`), מערך של תווים (`buff`), ואת אורך המערך (`buffLen`). היא כותבת את המחרוזת הבינרית המתאימה בתוך המערך.

- המחרוזת הנכתבת במערך צריכה להכיל בדיוק `buffLen-1` תווים ('0' או '1'), בהתאם לייצוג הבינרי של `num`.
 - אם `buffLen ≤ 0`, אז הפונקציה לא עושה כלום ומחזירה ערך 0. אחרת, היא מחזירה 1 אחרי שפועלת לפי ההנחיות הבאות.
 - אם `num` ניתן לייצוג ע"י פחות מ `buffLen-1` ביטים, אז המחרוזת הנכתבת במערך צריכה להכיל אפסים מובילים. לדוגמא, הקריאה `numToBinaryString(5, buff, 5)` צריכה לכתוב את המחרוזת "0101" לתוך המערך (עם האפס המוביל).
 - אם `num` אינו ניתן לייצוג מדוייק ע"י `buffLen-1` ביטים, אז המחרוזת הנכתבת במערך צריכה להכיל את ה `buffLen-1` ביטים הנמוכים (מימין) של `num`. לדוגמא, הקריאה `numToBinaryString(5, buff, 3)` צריכה לכתוב את המחרוזת "01" לתוך המערך (לזרוק את ה '1' המוביל משמאל).
 - אל תשכחו להוסיף את ה '\0' המסיים את המחרוזת.
- עצת מימוש:** כתבו את המחרוזת מימין לשמאל.

```
int numToBinaryString(int num, char* buff, int buffLen) {
    int ind;

    if(buffLen <= 0)
        return 0;

    buff[buffLen-1] = '\\0';

    for(ind=buffLen-2; ind>=0; ind--) {
        buff[ind] = '0' + (num & 1);
        num >> 1;
    }

    return 1;
}
```

b) (7 pts.) Implement a function named `binaryStringToNum`, which takes a string holding the binary representation of some number, and it returns the appropriate integer value.

- If the string contains characters other than '0' or '1', the function should return -1.
- If the string is empty, the function should return 0.
- If the string contains more characters than the number of bits that an `int` takes, then the returned value should correspond to the least significant (right-most) characters in the string.
- Your code should be portable across platforms, so you should not assume that you know the number of bits that an `int` takes.

Implementation tip: read the string from left to right and use bitwise operators to update the value of the binary vector that you have read up to the present point.

(ב) (7 נק') ממשו פונקציה בשם `binaryStringToNum`, שלוקחת מחרוזת שמכילה ייצוג בינרי של מספר כלשהו, ומחזירה את הערך המספרי המתאים.

- אם המחרוזת מכילה תווים שונים מ '0' או '1', אז הפונקציה צריכה להחזיר -1.
- אם המחרוזת ריקה, אז הפונקציה צריכה להחזיר 0.
- אם המחרוזת מכילה יותר תווים ממספר הביטים שטיפוס `int` מכיל, אז הערך המוחזר צריך לייצג את התווים ה"נמוכים" (משמאל) במחרוזת.
- הקוד שלכם צריך להיות נייד בין מערכות שונות, כך שאסור לכם להניח שאתם יודעים את מספר הביטים שטיפוס `int` מכיל.

עצת מימוש: קראו את המחרוזת משמאל לימין, והיעזרו באופרטורים על ביטים כדי לעדכן את הערך המתאים לווקטור הבינרי שנקרא עד עתה.

```
int binaryStringToNum(const char* binaryStr) {
    int val = 0;

    while(*binaryStr) {
        if(*binaryStr != '0' && *binaryStr != '1')
            return -1;
        val = (val << 1) + *binaryStr - '0';
        binaryStr++;
    }
    return val;
}
```

c) (7 pts.) Implement a function named `isEven`, which takes a string holding the binary representation of some number and returns 1 if the represented number is even and 0 if it is odd.

- The function may assume that the input string contains only '0' and '1' characters.
- The empty string is considered to be even (value 0).
- Do not call any of the functions you implemented in a-b.
- See sample executions below.

ג) (7 נק') ממשו פונקציה בשם `isEven`, שלוקחת מחרוזת שמחזיקה ייצוג בינרי של מספר ומחזירה 1 אם המספר המיוצג הוא זוגי ו 0 אם הוא אי-זוגי.

- הפונקציה יכולה להניח שמחרוזת הקלט מכילה רק תווי '0' ו '1'.
- המחרוזת הריקה נחשבת זוגית (ערך 0).
- אל תקראו לפונקציות שמימשתם בסעיפים א-ב.
- ראו קריאות לדוגמא למטה.

Sample executions:

קריאות לדוגמא:

```
# define BUFFLEN 5
char buffer[BUFFLEN];
int flag;

intToBinaryString(3,buffer,BUFFLEN);
flag = isEven(buffer);          // flag = 0
intToBinaryString(14,buffer,BUFFLEN);
flag = isEven(buffer);         // flag = 1
intToBinaryString(0,buffer,0);
flag = isEven(buffer);         // flag = 1 (empty string)
```

```
int isEven(const char* binaryStr) {
    int ind = strlen(binaryStr)-1;
    /** empty string - return 1   ***/
    if(ind < 0) return 1;

    /** 1 if binaryStr[ind]=='0'
        0 if binaryStr[ind]=='1'   ***/
    return binaryStr[ind] == '0';
}
```

- d) (7 pts.) Implement a function named `shiftLeft`, which takes a string holding the binary representation of some number and performs the shift-left (`<<`) operation on that number.
- The function may assume that the input string contains only '0' and '1' characters.
 - The number of the characters in the string should not be changed by the operation.
 - Follow the standard rules of the shift-left operation, including bit overflow.
 - Do not call any of the functions you implemented in a-c.
 - See sample executions below.

Implementation tip: perform the operation from left to right.

(ד) (7 נק') ממשו פונקציה בשם `shiftLeft`, שלוקחת מחרוזת שמחזיקה ייצוג בינרי של מספר ומבצעת עליו פעולת הזזה לשמאל (`<<`).

- הפונקציה יכולה להניח שמחרוזת הקלט מכילה רק תווי '0' ו '1'.
- מספר התווים במחרוזת לא אמור להשתנות ע"י הפונקציה.
- פעלו על פי הכללים הסטנדרטיים של פעולת ההזזה לשמאל, כולל גלישת ביטים (overflow).
- אל תקראו לפונקציות שמימשתם בסעיפים א-ג.
- ראו קריאות לדוגמא למטה.

עצת מימוש: בצעו את הפעולה משמאל לימין.

Sample executions:

קריאות לדוגמא:

```
# define BUFFLEN 5
char buffer[BUFFLEN];
int num=3; // num = 3

intToBinaryString(num,buffer,BUFFLEN);
shiftLeft(buffer);
num = binaryStringToNum(buffer); // num = 6
shiftLeft(buffer);
num = binaryStringToNum(buffer); // num = 12
shiftLeft(buffer);
num = binaryStringToNum(buffer); // num = 8 (overflow)
shiftLeft(buffer);
num = binaryStringToNum(buffer); // num = 0 (overflow)
```

```
void shiftLeft(char* binaryStr) {
    /** empty string - do nothing */
    if(!*binaryStr) return;
    /** shift */
    while(binaryStr[1]) {
        binaryStr[0] = binaryStr[1];
        binaryStr++;
    }
    /** push 0 to the right */
    binaryStr[0] = '0';
    return;
}
```

e) (7 pts.) Implement a function named `sumBinary`, which takes two strings holding binary representations of two number (`binaryStr1` and `binaryStr2`) and adds the second number to the first.

- The string held by `binaryStr1` after the invocation of the function should hold a binary vector representing the sum of the two numbers, and the string held by `binaryStr2` should be unchanged by the function.
- The length of the string in `binaryStr1` should not be changed by the function. If the value of the sum cannot be represented by this number of bits, then overflow occurs, as expected in an operation of two integer numbers.
- The function may assume that both input strings contains only '0' and '1' characters.
- Do not call any of the functions you implemented in a-d.
- See sample executions below.

Implementation tips: perform the operation by scanning both strings simultaneously from right to left, until you reach the start of `binaryStr1`. In each step, you need to add the two bits in the current position in the two strings, plus a possible "carried over" value from the previous position. For example, adding bits $1 + 1$ gives bit 0 and a value of 1 carried over to the next position. Make sure to keep the "carry-over" value as you scan the strings.

ה) (7 נק') ממשו פונקציה בשם `sumBinary`, שלוקחת שתי מחרוזות המכילות ייצוג בינרי של שני מספרים (`binaryStr1` ו `binaryStr2`), ומחברת את המספר השני לראשון.

- המחרוזת הנמצאת ב `binaryStr1` אחרי הקריאה לפונקציה צריכה להחזיק וקטור בינרי המייצג את הסכום של שני המספרים, והמחרוזת הנמצאת ב `binaryStr2` אמורה לא להשתנות ע"י הקריאה לפונקציה.
- מספר התווים במחרוזת `binaryStr1` לא אמור להשתנות ע"י הפונקציה. אם מספר הביטים הדרוש לייצוג מדויק של הסכום גדול ממספר תווים זה, אז מתרחשת גלישת ביטים, כפי שקורה בפעולות אריתמטיות של מספרים שלמים.
- הפונקציה יכולה להניח ששתי מחרוזות הקלט מכילות רק תווי '0' ו '1'.
- אל תקראו לפונקציות S שמישתם בסעיפים א-ד.
- ראו קריאות לדוגמא למטה.

עצות למימוש: בצעו את הפעולה ע"י סריקה של שתי המחרוזות משמאל לימין בו-זמנית, עד שאתם מגיעים להתחלה של מחרוזת `binaryStr1`. בכל צעד, עליכם לחבר את שני הביטים במיקום הנוכחי בשתי המחרוזות, ואולי גם ערך ש"נגרר" מהמיקום הקודם. לדוגמא, חיבור של הביטים $1 + 1$ מביא לביט 0 וערך 1 שנגרר למיקום הבא. וודאו שאתם שומרים את הערך ה"נגרר" תוך כדי סריקת המחרוזות.

Sample executions:

קריאות לדוגמא:

```
# define BUFFLEN 5
char buffer1[BUFFLEN], buffer2[BUFFLEN];
int num;

intToBinaryString(3,buffer1,BUFFLEN);
intToBinaryString(5,buffer2,BUFFLEN);
sumBinary(buffer1, buffer2);
num = binaryStringToNum(buffer1); // num = 8
sumBinary(buffer1, buffer2);
num = binaryStringToNum(buffer1); // num = 13
sumBinary(buffer1, buffer2);
num = binaryStringToNum(buffer1); // num = 2 (overflow)
sumBinary(buffer1, buffer2);
num = binaryStringToNum(buffer1); // num = 7
```

```

void sumBinary(char* binaryStr1, const char* binaryStr2) {
    int ind1 = strlen(binaryStr1)-1,
        ind2 = strlen(binaryStr2)-1,
        carry = 0;

    /*** scan both vectors from right to left ***/
    while(ind1>=0 && ind2>=0) {
        binaryStr1[ind1] += carry + binaryStr2[ind2] - '0';
        /*** check for carry for next position ***/
        if(binaryStr1[ind1] > '1') {
            binaryStr1[ind1] -= 2;
            carry = 1;
        } else {
            carry = 0;
        }
        ind1--;
        ind2--;
    }

    /*** if there is left over in binaryStr1,
        then propagate the carry until it reaches 0 ***/
    while(ind1>=0 && carry > 0) {
        binaryStr1[ind1] += carry;
        /*** check for carry for next position ***/
        if(binaryStr1[ind1] > '1') {
            binaryStr1[ind1] -= 2;
            carry = 1;
        } else {
            carry = 0;
        }
        ind1--;
    }

    return;
}

```

Functions from standard library string.h

strlen - calculate the length of a string

- `size_t strlen(const char *s);`

Function description:

The **strlen()** function calculates the length of the string *s*, excluding the terminating null byte ('\0').

Return value:

Function returns the number of bytes (characters) in the string *s*.

strcpy, strncpy - copy a string

- `char *strcpy(char *dest, const char *src);`
- `char *strncpy(char *dest, const char *src, size_t n);`

Function description:

The **strcpy()** function copies the string pointed to by *src*, including the terminating null byte ('\0'), to the buffer pointed to by *dest*. The strings may not overlap, and the destination string *dest* must be large enough to receive the copy. *Beware of buffer overruns!*

The **strncpy()** function is similar, except that at most *n* bytes of *src* are copied. If the length of *src* is less than *n*, **strncpy()** writes additional null bytes to *dest* to ensure that a total of *n* bytes are written.

Warning: If there is no null byte among the first *n* bytes of *src*, the string placed in *dest* will not be null-terminated.

Return value:

Both functions return a pointer to the destination string *dest*
