שאלה 1

סעיף א'

כתוב פונקציה

```
int indexOfChar(const char* str, char c)
```

המקבלת מחרוזת (str) ותו (c) ומחזירה את האינדקס במחרוזת של המופע הראשון של התו c.
אם התו לא קיים במחרוזת, הפונקציה תחזיר 1-.

סעיף ב'

כתוב פונקציה המקבלת מחרוזת (str) המתארת תרגיל חיבור חשבוני של מספרים שלמים וחיוביים ומחזירה את ערך הפתרון של התרגיל.

לדוגמא הקריאה לפונקציה sumOfNums("3+4+17") תחזיר את המספר 24.

הדרכה: מיצאו את הערך של המחובר הראשון בתרגיל וחברו לו את ערך התרגיל, החל מהמחובר השני.

בשאלה זו ניתן להניח כי:

- המחרוזת מכילה רק מספרים שלמים חיוביים המופרדים ע"י התו '+', ללא רווחים.
- המחרוזת מכילה לפחות מספר אחד (כלומר היא לא ריקה).
- כל מספר לא מכיל יותר מ-10 ספרות (לא חובה להשתמש בנתון זה).
- ניתן לשנות את מחרוזת הקלט.

מומלץ להשתמש בסעיף זה בפונקציה int atoi (const char *str) הנמצאת בספרייה stdlib.h, המקבלת מחרוזת של מספר שלם מחזירה את המספר כערך מטיפוס int.

סעיף ג'

כתוב פונקציה

```
void splitString(const char* str, int index,
                 char* left, char* right)
```

המקבלת מחרוזת (str) ואינדקס (index), מפצלת את המחרוזת ל-2 מחרוזות עפ"י האינדקס ושמה אותם במערכים left ו-right בהתאמה.
כלומר, לאחר ריצת הפונקציה, המערך left יכיל את תת המחרוזת str[0…index-1] והמערך right יכיל את תת המחרוזת str[index+1…end]. שימו לב שהתו המופיע בתא str[index] לא יופיע באף אחת משתי המחרוזות.

לדוגמא:
עבור המחרוזת "1+3=2+2" והאינדקס 3, הפונקציה תשים ב-left את "1+3" וב-right את "2+2".

ניתן להניח שבמערכים left ו- right יש מספיק מקום כדי להכיל את תתי המחרוזות.
שימו לב – בשאלה זו **אסור** לשנות את המחרוזת str.

סעיף ד'

כתוב את הפונקציה isCorrect המקבלת מחרוזת של משוואה מתמטית (המכילה חיבור בלבד) ובודקת האם המשוואה נכונה. אם היא נכונה הפונקציה תחזיר TRUE (ערך שונה מאפס) ואם היא שגוייה היא תחזיר FALSE (אפס).

לדוגמא:

isCorrect("12+3=7+8") יחזיר TRUE

isCorrect("2+3=4") יחזיר FALSE

בשאלה זו ניתן להניח כי:

- המחרוזת מכילה רק מספרים שלמים חיוביים המופרדים ע"י התו '+' והתו '=', ללא רווחים.
- המחרוזת מכילה תו '=' אחד בלבד
- כל צד של המשוואה מכיל לפחות מספר אחד
- כל מספר לא מכיל יותר מ-10 ספרות (לא חובה להשתמש בנתון זה)
- **אסור** לשנות את מחרוזת הקלט.

בסעיף זה חובה להשתמש בפונקציות מהסעיפים הקודמים.

שימו לב כי יש ליצור תתי מחרוזות עבור כל צד של המשוואה ויש להשתמש בזכרון המספיק בדיוק בשביל להכיל אותם (ולא יותר).

1. כתוב פונקציה המקבלת מחרוזת ומשנה אותה, כך שבכל זוג איברים צמודים מוחלף סדר התווים.
לדוגמא : good תהפוך להיות ogdo.

2. נתון מערך דו מימדי שמימדיו 10X4 המכיל את שמות תלמידי כיתה היושבים ב 4 טורים.
ליד כל שולחן יושב תלמיד אחד. כל תא במערך מכיל את שמו הפרטי ושם המשפחה של תלמיד. אם במקום מסוים לא יושב תלמיד, תהיה באותו תא מחרוזת ריקה.

   כדי לבדוק את תקינות השמות יש לדאוג שכל שם (פרטי או משפחה) יהיה מורכב משתי אותיות לפחות, ושלא תהיינה יותר משלוש אותיות א,ה,ו,י רצופות.

   פתח ויישם בשלבים, אלגוריתם המדפיס את השמות השגויים, ואת מיקומם במערך.

3. כתוב פונקציה המקבלת מחרוזת. מחרוזת זו מכילה תווים שונים וכוכביות. על הפונקציה להחזיר את אורך תת המחרוזת הארוכה ביותר שמהווה פלינדרום ובדיוק במרכזה נמצאת המחרוזת "middle".

4. כתוב פונקציה המקבלת מחרוזת שאינה מכילה אך התווים "." ו – "-", ומחזירה את סכום המספרים שהופיעו במחרוזת. דוגמא : עבור המחרוזת ab12d580c600, תחזיר הפונקציה 1192 (12+580+600).

5. מעבד תמלילים "smarty" יודע לבדוק במשפט שלם אם יש שתי מילים רצופות זהות, ולבטל אחת מהן. כתוב פונקציה המקבלת מחרוזת המכילה משפט, ומבטלת בה הופעות חוזרות רצופות של מילה. כל מילה שהופיעה ברצף, תישאר פעם אחת.

   ידוע שבין מילה למילה מופיע התו רווח. הפונקציה תחזיר את המשפט המעודכן.

6. כתוב פונקציה המקבלת מחרוזת, מחרוזת להחלפה וכן מחרוזת מחליפה. הפעולה תחליף את כל המופעים של המחרוזת להחלפה במחרוזת המחליפה. יש לשים לב שאורך המחרוזות לא בהכרח שווה. ניתן להניח שיש מספיק מקום להחליף את כל המופעים.
לדוגמא:

מחרוזת: Hello to you today

מחרוזת להחלפה: o

מחרוזת מחליפה: www

המחרוזת לאחר הקריאה לפעולה תיראה כך: Hellwww twww ywwwu twwwday

שאלה 7

לפניך תוכנית בשפת C
עקוב בעזרת טבלת מעקב אחר התוכנית וציין את הפלט המדויק.
תשובה ללא מעקב לא תתקבל.

```c
#include <stdio.h>
#include <string.h>

void what(char *string){
      int i;
      char ch, *temp1, *temp2, arr[100];

      temp1 = string;
      temp2 = arr;

      while(*temp2++ = *temp1++){
            if(*temp1 >'2' && *temp1 <= '9'){
                  i = *temp1 - '0' - 1;

                  puts(arr);
                  for( ch = *(temp1-1); i;temp2++, i--)
                        *temp2 = ch;
                  temp1++;
            }
      }
      strcpy(string, arr);
}
void main(){
      char st[50]="a3b4ccad3";
      what(st);
      puts(st);
}
```

שאלה 8

א. נתונה קבוצה של n תווים: $a_1 = a$, $a_2 = b$, $a_3 = c$, $a_4 = d$, ... $a_n = m$

לפניך קטע קוד שמדפיס את כל המילים (המחרוזות) באורך n , כך שכל תו במחרוזות שייך לקבוצת התווים הנתונה.

דוגמה: עבור $n = 3$ קבוצת התווים היא: $a_1 = a$, $a_2 = b$, $a_3 = c$ והפלט הוא כל המילים (המחרוזות), באורך 3 , כמפורט מטה:

aaa

aab

aac

aba

abb

abc

aca

acb

acc

baa

bab

bac

bba

:::

ccc

בקוד חסרים **שלושה** ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1)–(3) בלבד, בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

```
#define n 3
void main(void)
 { char s[4]={'a','a','a',0};
   int i,j,t;
   t= n*n*n;
   while (t--)
     {
         for ( i=0;  i<n-1;   i++)
               printf("%1c", s[i]);
         printf("%1c\n",s[i]);
         if (++s[n-1]> 'c')
            {
               ____(1)_____ ;
               for( j=____(2)_____; j>=0;  j--)
                  {
                    if(____(3)_____)
                       s[j]='a';
                    else   break;
                  }
            }
     }
}
```

ב. לפניך ההגדרה הזאת:

```
char *p[2][3]={"question", "number", "'4'", "good", "luck", "!!!"};
```

מה יהיה הפלט שיודפס בעקבות כל אחת מן ההוראות שלהלן:

```
1.    printf("%s\n" , *p[0]+5);
2.    printf("%c\n" , *(p[1][2]+2));
3.    printf("%c\n" , **p[1]);
```

שאלה 9

עקוב אחר התוכנית הבא והשלם את הפלט

```
#include <stdio.h>
int z = 10, y = 234112;
int *p = &z;

void f1(int x) {
      int z = y;
      for (y=2; y<10; ++y);
      ++x;
      printf("x=%d y=%d z=%d\n", x, y, z);
}
void f2(int *x) {
      *x = z;
      y = (*x == z) && 1 && (*p == *x) || 0;
      ++*x;
      printf("x=%d y=%d z=%d\n", *x, y, z);
}
int main(void) {
      int x = 77, y = 0;
      f1(x);

      printf("x=%d y=%d z=%d\n", x, y, z);

      *p = 5678; x=1234;
      f2(&x);

      printf("x=%d y=%d z=%d\n", x, y, z);

      return 0;
}
```