מכון טכנולוגי חולון
Holon Institute of Technology

המחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT

מעבדה למחשבים אישיים 61105
סמסטר ב' תשס"ח
מועד א'
15.06.2008

מרצה: ד"ר מרק קורנבליט

משך המבחן: שלוש שעות

אין להשתמש בחומרי עזר פרט לדף מצורף לשאלון

(35 נק')

1. כתוב פונקציה אשר מקבלת שני מערכים A ו-B (מהטיפוס int), ובונה מערך חדש C באופן הבא: איבר מס' 0 של C שווה לאיבר מס' 0 של A, איבר מס' 1 של C שווה לאיבר מס' 0 של B, איבר מס' 2 של C שווה לאיבר מס' 1 של A, איבר מס' 3 של C שווה לאיבר מס' 1 של B וכו'.

גודלי המערכים המקוריים לא שווים במקרה הכללי. כאשר המערך בעל הגודל הקטן נגמר, כל איבר שנשאר במערך הגדול מופיע במערך החדש C פעמיים.

יש להקצות מקום בזיכרון שדרוש למערך החדש בתוך הפונקציה. ניתן להניח שיש בזיכרון מספיק מקום להקצאה כזאת. על הפונקציה להחזיר כתובת של המערך החדש ולהעביר את גודלו לתוכנית הראשית.

דוגמא:

A: 8, 6, 12, 9
B: 3, 5, 1, 4, 0, 5, 6
C: 8, 3, 6, 5, 12, 1, 9, 4, 0, 0, 5, 5, 6, 6.

```
int *pair_array (int *a, int *b, int asize, int bsize, int *csize)
 {
   int *c, *temp, min, max, i;

   /* find sizes of shorter and longer arrays */
   if (asize < bsize)
     {
          min = asize;
          max = bsize;
          temp = b;
     }
```

```
    else
     {
          min = bsize;
          max = asize;
          temp = a;
     }

    /* size of array C is a double size of longer array */
          *csize = 2*max;
    c = calloc(*csize, sizeof(int));

    /* fill array C by elements of arrays A and B;
       this process continues till the end of the shorter array */
    for (i=0; i<min; i++)
     {
          c[2*i] = a[i];
          c[2*i+1] = b[i];
     }

    /* each of the remaining elements in the longer
       array is copied into C twice */
     for (i=min; i<max; i++)
          {
            c[2*i] = temp[i];
            c[2*i+1] = temp[i];
          }
    return c;
  }
```

(25 נק')

2. כתוב פונקציה FindFirstVowel המקבלת מחרוזת ומחזירה את הכתובת של ה - Vowel הראשון שהיא נתקלת בו בסריקה משמאל לימין. קבוצת ה - Vowels מכילה את חמשת האותיות: A,a;E,e;I,i;O,o;U,u.

הפונקציה תחזיר NULL אם אין במחרוזת אף Vowel.

```
char *FindFirstVowel (char *str)
{
          for (; *str != '\0'; str++)
                    if (*str == 'A' || *str == 'a' ||
                       *str == 'E' || *str == 'e' ||
                        *str == 'I' || *str == 'i' ||
                        *str == 'O' || *str == 'o' ||
                        *str == 'U' || *str == 'u')
                              return str;
          return NULL;
}
```

(25 נק')

3. כתוב פונקציה שמדפיסה משפט "third time lucky" על המסך כל פעם שלישית, ששית, תשיעית וכו' שנקראת מתוכנית הראשית ולא מבצעת כלום כאשר נקראת בפעם הראשונה, השנייה, הרביעית, החמישית, השביעית וכו'. הפונקציה לא משתמשת באף משתנה מתוכנית הראשית.

```
void third ()
{
        static int count;
        if (count == 2)
        {
                printf ("third time lucky\n");
                count = 0;
        }
        else
                count++;
}
```

(15 נק')

4. איבר של הרשימה המקושרת נתון באמצעות ההגדרה הבאה:

```
typedef struct element
{
     char data;
     struct element *ptr_next;
}    element;
```

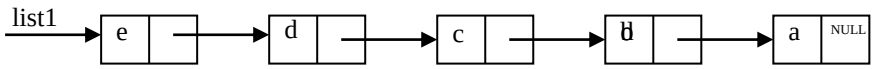
מה מבצעת הפונקציה הבאה עקב הצבת כתובת של מצביע לאיבר ראשון של הרשימה במקום `pl`?

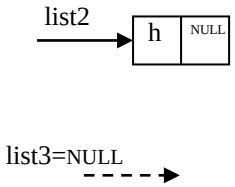
```
void sod (element **pl)
{
     element *curr, *prev, *temp;

     curr = *pl;
     prev = NULL;
     while (curr)
    {
          temp = curr->ptr_next;
          curr->ptr_next = prev;
          prev = curr;
          curr = temp;
     }
     *pl = prev;
}
```

הפונקציה הופכת את הרשימה.

מה יקרה עקב ביצוע **sod** עם כל אחת משלוש הרשימות הבאות? הראה את האפקט.





(בונוס – 10 נק')

5. לפניך פונקציה **func**:

```
int func (char *str)
  {
    int sum=0, i;
    for (i=0; str[i] != '\0'; i++)
      sum += str[i];
    return sum - *str*strlen(str);
  }
```

מה מחזירה פונקציה **func** אם **str** מצביע על המחרוזת **"abcde"**?

10

בהצלחה!