שאלה 1 (15 נקודות)

בכל סעיף, עליכם לכתוב אם האמור "**תמיד נכון**", בשפת ANSI-C, "**לפעמים נכון ולפעמים אינו נכון**" או "**תמיד אינו נכון**". עליכם לנמק את תשובתכם, תשובה לא מנומקת, גם אם היא נכונה, לא תזכה בנקודות.

(5 נק') א. אם הפונקציה func מופיעה באב טיפוס (prototype) באופן הבא:

```
double func(double);
```

אז בקריאה לפונקציה:

```
x = func(3);
```

המספר 3 (שהוא integer) יהפוך לערך 3.0 (שהוא double) ללא צורך בהמרה (cast).

(5 נק') ב. כאשר רוצים לחסוך בזיכרון, ניתן להשתמש ב-bit-field. באופן זה אורזים כמה אובייקטים במילת זיכרון אחת. כך גם ניתן להתייחס לכתובת כל ביט, על יד שימוש באופרטור הכתובת '&'.

(5 נק') ג. כדי לקרוא בתים ניתן להשתמש ב-system call (קריאת מערכת) "read". מומלץ לקרוא בית אחד בכל קריאה.

שאלה 2 (18 נקודות)

(10 נק') א. עליכם לכתוב מקרו, המשמש לחיפוש איבר במערך **מספרי כלשהו**. על המקרו למצוא את האיבר ולספק את מיקומו במערך. על המקרו לקבל פרמטרים על פי הצורך, ולטפל בשגיאות במידת הצורך.

<u>דוגמא</u>: מערך מכיל את האיברים: 2, 5, 7, 88, 9, 11

המערך הנ"ל מכיל את המספר 7 במקום השלישי.

כאשר הקלט למקרו הוא 7, אזי התוצאה המבוקשת היא 3 (המקום השלישי).

(8 נק') ב. עליכם לכתוב תכנית **מלאה** (כלומר כוללת תכנית ראשית, וניתנת להרצה, ללא תוספת קוד), המשתמשת במקרו מהסעיף הקודם, ומחפשת איבר כלשהו במערך בן 12 איברים.

שאלה 3 (27 נקודות)

בסעיפים הבאים נתונים קטעי תוכניות בשפת ANSI-C. הקטעים יכולים שלא לבצע את המצופה מהם. עבור כל קטע, עליכם לכתוב **אם** הוא שגוי ומדוע, **מה** השגיאה/שגיאות, **וכיצד** יש לתקנו.
אם קטע התכנית אינו שגוי, עליכם להסביר כיצד הוא מבצע את המוטל עליו. יש לכתוב עד 4 שורות של הסבר.

(9 נק') א. קטע התכנית הבא אמור להגדיל את גודלו של מערך בעשרה איברים.

```
int **p1;
int **p2;
p1 = (int **)malloc(100*sizeof(int *));
.....
p2=p1;
p1 = (int **)malloc(110*sizeof(int *));

if (p1 == NULL)
  {
    puts("Fail: allocation error");
    exit (1);
  }


for  (i=0; i<100; i++)
    p1[i] = p2[i];

free(p2);
```

(9 נק') ב. התכנית הבאה אמורה להחליף תוכן שני משתנים : i - אשר ערכו ההתחלתי 5, array[5] - אשר ערכו ההתחלתי נקבע באתחול המערך.

```
#define swap(A, B) {int temp = (A); (A) = (B); (B) = temp;}

void main (void)
{
    int array[] = {2,5,79,81,0,4};
    int i;
    i=5;
    swap(i, array[i]);
}
```

temp = 5
i = 4
array[i] = 5

(המשך השאלה בעמוד הבא)

85.79.412

(9 נק') ג. התכנית הבאה אמורה להדפיס אילו מנות מזון להגיש, על פי העדפת הסועדים.

```
1  #include <stdio.h>
2  #include <string.h>
3  enum type { H=1 /* herbivore */, C=2 /* carnivore */,
4              S=4 /* sweet tooth */, D=8 /* Drinker */ };
5
6  char *dish( int x )
7    {
8    char buf[100]; char *p = buf; *p = 0;
9    if( x & D ) strcat( p, "rum-soaked " );
10   if( x & S ) strcat( p, "sugared " );
11   if( x & C ) strcat( p, "turkey " );
12   if( x & H ) strcat( p, "broccoli " );
13   return p;
14   }
15 int main()
16   {
17   printf( "please pass the %s and the %s.",
18       dish(C|S), dish(H|D) );
19   return 0;
20   }
```

H = 1
C = 2
S = 4
D = 8

0001	1	H
0010	2	C
0011	3	
0100	4	S
0101	5	
0110	6	
0111	7	
1000	8	D

המשך הבחינה בעמוד הבא

85.79.412

שאלה 4 (40 נקודות)

(5 נק') א. עליכם להגדיר טיפוס מבנה נתונים בשם sculpture לאחסון מידע של פסלים, המועמדים לתצוגה בתערוכה במוזיאון. על פרטי הפסלים להופיע **ברשימה מקושרת** כאשר השליפה וההכנסה נעשים על-פי מבנה נתונים של **מחסנית**.
הנתונים הדרושים הם:

1. שם פסל
2. חומר (עץ, מתכת, אבן, פלסטיק)
3. משקל/שטח
4. אמן/גלריה

פרטי אמן/גלריה הדרושים הם:

1. שם אמן/שם גלריה
2. מספר טלפון
3. סוג ביטוח נדרש (מקיף, חלקי, ללא ביטוח)

(35 נק') ב. תוך שימוש במבנה הנתונים שהגדרתם בסעיף א', עליכם לכתוב תכנית מלאה המטפלת במחסנית של פסלים **(אין להניח שהרשימה נתונה, יש לבנותה)**.
עליכם לממש push, pop של הפסלים ברשימה המקושרת.
על התכנית למצוא את כל הפסלים ברשימה, אשר הוגדרו "ללא שם" (כלומר הפסלים בשם "no name"), להדפיס כמה פסלים קיימים, תחת שם זה, ומה נתוניהם.

הבהרה:
יש לכתוב תכנית **מלאה**, כלומר כוללת תכנית ראשית, וניתנת להרצה, ללא תוספת קוד. יש לממש כל פונקציה שאתם משתמשים בה. לא ניתן להניח קיום פונקציות עזר חיצוניות, שאינן מהספרייה הסטנדרטית. יש להגדיר כל מבנה נתונים שהשתמשתם בו.

בהצלחה!