הנדסאים באריאל
קמפוס אוניברסיטת אריאל

בחינות גמר ממלכתיות להנדסאים ולטכנאים
מועד הבחינה: מועד ב 5.2.20
סמסטר א

משרד העבודה הרווחה והשירותים החברתיים
חוסן חברתי לישראל

מספר ת"ז: _____
מספר מחברת: _____

C

נבחן תחת שם קורס אריאל שפת C ב
מספרי קבוצות: 7424332 1442001

הנדסאים וטכנאים – הנדסת תוכנה

שם מרצה: אפי פרופוס

הנחיות לנבחן

א. משך הבחינה: ארבע שעות

ב. מבנה השאלון ומפתח ההערכה:

בשאלון זה **שמונה** שאלות.
יש לענות על **שש** שאלות בלבד בהתאם לפירוט הבא:

חלק א' – שאלה 1 (חובה)	15 נקודות
חלק ב' – שאלות 2-4 (יש לענות על **2** שאלות בלבד)	40 נקודות
חלק ג' – שאלות 5-8 (יש לענות על **3** שאלות בלבד)	45 נקודות
סך הכול	**100 נקודות**

ג. חומר עזר מותר לשימוש:

1. מחשבון (אין להשתמש במחשב כף יד או במחשבון עם תקשורת חיצונית).
2. קלסר **אחד בלבד** עם **חומר ההרצאות**. אין להוציא דפים מהקלסר.

אין לצרף ספרים או חוברות עם פתרונות.

ד. הוראות כלליות:

1. יש לקרוא בעיון את ההנחיות בדף השער ואת כל שאלות הבחינה, ולוודא שהן מובנות.
2. את התשובות יש לכתוב בצורה מסודרת, בכתב יד ברור ונקי. (גם בכך תלויה הערכת הבחינה).
3. יש לכתוב **בעט בלבד**.
4. יש להתחיל כל תשובה בעמוד חדש ולציין את מספר השאלה ואת הסעיף. אין צורך להעתיק את השאלה עצמה.
5. טיוטה יש לכתוב במחברת הבחינה בלבד. יש לרשום את המילה "טיוטה" בראש העמוד ולהעביר עליו קו כדי שלא ייבדק.
6. יש להציג פתרון מלא ומנומק, כולל חישובים לפי הצורך. הצגת תשובה סופית ללא שלבי הפתרון לא תזכה בניקוד.
7. יש להסביר בפירוט כל תוכנית שנכתבה, **תוכנית ללא הסבר מפורט לא תזכה בניקוד.**
8. אם לדעתך חסר בשאלה נתון, יש לציין זאת ולהוסיף נתון מתאים שיאפשר לך להמשיך בפתרון השאלה. נמק את בחירתך.

חל איסור מוחלט להוציא שאלון או מחברת בחינה מחדר הבחינה!
ההנחיות בשאלון זה מנוסחות בלשון זכר, אך מכוונות לנבחנות ולנבחנים כאחד.

בהצלחה!

בשאלון זה ישנם 9 עמודים כולל עמוד זה.

תרגיל 1 – שאלת חובה.

מה יודפס?

```
#include <stdio.h>

typedef struct NODE
{
    int num;
    struct NODE *next;

}NODE;


int main()
{
   int x=6 , y = 4,i,j, mat[3][3]={0};

   printf("%d\n" , x = x*y>>2^3);

   if( x^0x2 > 1)
   {
      printf("not right\n");
   }
   else
   {
       printf("go left\n");
   }

   NODE *head = (NODE *) malloc (sizeof(NODE));
   head->num=8;
    NODE *temp = (NODE *) malloc (sizeof(NODE));
    temp->num=6;
  head->next = temp;
    NODE *temp1 = (NODE *) malloc (sizeof(NODE));
    temp1->num=3;

  temp->next = temp1;
  head->next->next = temp;
   i=4;


   mat[head->num%2][head->next->num%2] =3;
   *((*mat + head->num/4) + 1)= 2;
   *(mat[(head->next->next->next->next->num)/4] +1) =4;

   for(i=0;i<2 ; i++)
   {
       for(j=0 ; j<2 ; j++)
       {
           printf("%d ",mat[i][j] );
       }
   }

    return 0;
}
```

חלק ב, יש לפתור 2 מתוך שאלות 2-4:

תרגיל 2:

חלק א:

כתבו פונקציה רקורסיבית המקבלת מספר שלם num ומחזירה את המספר num בסדר הפוך. להלן חתימת הפונקציה:

int reverse(int num)

לדוגמה,
הפונקציה תקבל את המספר 54321 ותחזיר 12345.

דוגמה נוספת,
הפונקציה תקבל את המספר 1819823 ותחזיר 3289181.

הערות:
1. מותר להשתמש בפונקציות עזר.
2. אסור להשתמש בלולאות בפתרון (לא בפונקציה עצמה ולא בפונקציות עזר).
3. אסור לשנות את חתימת הפונקציה.

חלק ב:

כתבו פונקציה רקורסיבית המקבלת 3 מספרים. הפונקציה תחזיר true אם סכומם של כל שתי ספרות צמודות במספר הראשון שוות למספר השני או השלישי שהפונקציה קיבלה.

<u>דוגמאות:</u>

- עבור המספר 172636 והמספרים 8 ו- 9 הפונקציה תחזיר true מאחר והסכום של כל שתי ספרות צמודות הוא 8 או 9.
- עבור המספר 173544 והמספרים 8 ו- 9 הפונקציה תחזיר false מאחר וסכום הספרות הצמודות 3 ו- 7 אינו 8 או 9.

תרגיל 3:

להלן חתימה של פונקציה:

```
void putValues(int mat[][MAX_SIZE], int val, int size,
                                    int rowIndex, int colIndex)
```

הפונקציה מקבלת את הפרמטרים הבאים:

- mat - מטריצה ריבועית של מספרים שלמים שממדיה מוגדרים ע"י קבוע MAX_SIZE
- val - מספר שלם שיש לשים במטריצה
- size - מספר שלם המייצג גודל של תת-מטריצה ריבועית
- rowIndex - מספר שלם המייצג אינדקס של שורה במטריצה
- colIndex - מספר שלם מייצג אינדקס של עמודה במטריצה

הפונקציה תשים את הערך val בתת-מטריצה ריבועית שממדיה size, החל מהאיבר שמיקומו מתחיל בשורה שהאינדקס שלה הוא rowIndex ובעמודה שהאינדקס שלה הוא colIndex.

<u>שימו לב</u>: במידה ותת-המטריצה המתחילה במיקום [rowIndex][colIndex] אינה יכולה להכיל תת מטריצה ריבועית בגודל size, יש לשים לב לא לחרוג מגבולות המטריצה, ולמלא רק את האיברים שבגבולותיה.

<u>דוגמאות</u>:

עבור קריאה לפונקציה עם הערכים הבאים:

- matrix – מטריצה שגודלה 8x8
- val=0
- size=8
- rowIndex=0
- colIndex=0

המטריצה תתעדכן להיות:

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

עבור קריאה נוספת לפונקציה עם הערכים הבאים:

- matrix – המטריצה מהדוגמה הקודמת
- val =1
- size=5
- rowIndex=3
- colIndex=4

המטריצה תתעדכן להיות:

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1

עבור קריאה נוספת לפונקציה עם הערכים הבאים:

- matrix – המטריצה מהדוגמה הקודמת
- val =7
- size=4
- rowIndex=2
- colIndex=1

המטריצה תתעדכן להיות:

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	7	7	7	7	0	0	0
0	7	7	7	7	1	1	1
0	7	7	7	7	1	1	1
0	7	7	7	7	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1

תרגיל 4:

נתונה רשימה מקושרת אשר כל צומת בה מוגדר כך:

```
struct node{
        Int data;
        Node* next;
};
```

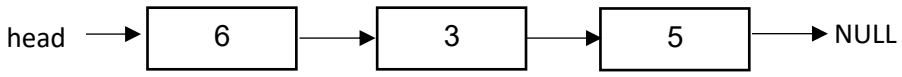
ידוע כי ברשימה לפחות איבר אחד וכי האיבר האחרון ברשימה מצביע ל-NULL. עליכם לכתוב פונקציה שחתימתה:

```
void  printHalf(Node* head)
```

הפונקציה מקבלת מצביע לתחילת רשימה מקושרת **ומדפיסה** בסדר הפוך את האיברים הבאים:

- אם מספר האיברים ברשימה זוגי, יודפסו האיברים במקומות הזוגיים ברשימה.
- אם מספר האיברים ברשימה אי-זוגי, יודפסו האיברים במקומות האי-זוגיים ברשימה.

לדוגמה,



לאחר הקריאה לפונקציה, יודפס:

5 6

חלק ג, יש לענות על 3 שאלות מתוך השאלות 5-8:

תרגיל 5:

א. מה הקוד מדפיס?
ב. מה הקוד מבצע?

```
#include <stdio.h>

int func(int arr[], int l, int r, int x)
{
if (r >= l)
{
        int m = l + (r - l)/2;


        if (arr[m] == x) return m;


        if (arr[m] > x) return func(arr, l, m-1, x);


        return func(arr, m+1, r, x);
}

return -1;
}

int main(void)
{
int arr[] = {2, 3, 4, 10, 40};
int n = sizeof(arr)/ sizeof(arr[0]);
int x = 10;
int result = func(arr, 0, n-1, x);
(result == -1)? printf(" no") : printf("yes %d", result);
return 0;
}
```

תרגיל 6:

מה יודפס במידה והתקבל base = 7 ?

```
void print (int base)
{
	int i, j, k;

	for (i = 1; i <= base; i++)
	{
		for (k = 0; k < base; k++)
		{
			for (j = 1; j <= i; j++)
				printf("*");

			for (; j <= base; j++)
				printf(" ");

			printf(" ");
		}
		printf("\n");
	}
}
```

תרגיל 7:

בהינתן המערך הבא:

	0	1	2	3	4
left	1	3	1	6	3
right	2	5	5	2	4

וגודל 5

מה יבצעו הפונקציות ואיך יראה המערך בסוף התהליך אם נפעיל את הפונקציה dMDV:

```
int find (Domino dominos[], int size)
{
	int maxSum = 0, i;
	for (i = 0; i < size; i++)
	{
		int sum = dominos[i].left + dominos[i].right;
		if (sum > maxSum)
			maxSum = sum;
	}
	return maxSum;
}
int dMDV (Domino dominos[], int size)
{
	int maxSum = find (dominos, size);
	int i;
	int newSize = size;
	for (i = 0 ; i < newSize; i++)
	{
		if (dominos[i].left + dominos[i].right == maxSum)
		{
			dominos[i] = dominos[newSize - 1];
			i--;
			newSize--;
		}
	}
	return newSize;
}
```

תרגיל 8:

עבור המטריצה הבאה:

10	28	19	41	3	0
-7	5	27	20	18	-3
19	33	58	-4	36	15
55	38	27	22	19	65
13	-3	10	17	0	10
6	-1	13	13	2	12

וקוד הבא, מה יחזור ב SUM ומה הפונקציה מבצעת?

```
void sum (int arr[][M])
{
	int sum = 0;

	for (int i = 0; i <= M; i++)
		for (int j = 0; j <= M; j++)
			if (j >= i && j < M - i)
				sum += arr[i][j];

	return sum;
}
```

בהצלחה!