שאלה 1 (25 נקודות)

סעיף א

נתונה תוכנית ה-C הבאה:

```
#define N 5

void mish(char** arr1, char** arr2, int n)
{
    char *p;
    if((n<=0) || (arr1 == arr2)) return;
    p=*arr1;
    *arr1=*arr2;
    *arr2 = p;
    mish(arr1+1, arr2-1, n-1);
}

void mash(char* arr[][N], int n)
{
    if(n>N/2) return;

    mish(&arr[n][0], &arr[N-n-1][N-1], N);
    mash(arr, n+1);
}

int main()
{
    int i, j;
    char* arr[][N] = {{"It", "is", "the", "end", "of"},
      {"the", "world", "as", "we"}, {"know", "it", "and"},
      {"I", "feel"}, {"fine"}};

    mash(arr, 0);
    for(i=0; i<N; i++)
        for(j=0; j<N; j++)
            printf("%s%c", arr[i][j]? arr[i][j] : "",
                           (j==N-1)? '\n' : '\t');

    return 0;
}
```

1. מה הם הפרמטרים אותם מקבלת הפונקציה `mish()` ומה מבצעת הפונקציה?
2. מה תדפיס התכנית?

1. הפונקציה מקבלת מצביע לתחילת שורה של מצביעים למחרוזות, מצביע נוסף לסוף שורה שניה של מצביעים למחרוזות וכן את אורך השורה ומחליפה את האיברים בשתי השורות כך שהאיבר הראשון בשורה הראשונה יהיה האחרון בשניה (ולהיפך), השני בשורה הראשונה יהיה שני מהסוף בשורה השנייה (ולהיפך) וכו.

2.הפונקציה תדפיס:

				fine
			feel	I
		and	it	know
	we	as	world	the
of	end	the	is	it

סעיף ב

בכל אחד מהסעיפים הבאים מופיעות מספר שורות קוד. לכל קטע קוד, הקיפו בעיגול את התיאור המתאים:

א. **ללא שגיאות** – הקוד יתקמפל ללא כל שגיאה וירוץ ללא תקלות.
ב. **שגיאת זמן ריצה** – הקוד יתקמפל ללא שגיאות, אולם הוא **עלול** לבצע שגיאה בזמן הריצה.
ג. **שגיאת קומפילציה** – הקוד לא יעבור קומפילציה.

1.

```
int arr[]={1,2,3,4,0};
while(*arr)
  printf("%d ", *(arr++));
```

א. ללא שגיאות
ב. שגיאת זמן ריצה
ג. שגיאת קומפילציה

2.

```
int i[8] = {0,1,2,3,4,5}, *p = i+4;
*(p + 4) = 8;
```

א. ללא שגיאות
ב. שגיאת זמן ריצה
ג. שגיאת קומפילציה

3.

```
int i=1, *j=&i+1;
(double(i/2))? *j=2 : *(j-1)=2;
```

א. ללא שגיאות
ב. שגיאת זמן ריצה
ג. שגיאת קומפילציה

4.

```
void f(int x) {
  --x;
}
void h(int n) {
  if(!n) return;
  h(f(n));
}
int main() {
  h(3);
  return(0);
}
```

א. ללא שגיאות
ב. שגיאת זמן ריצה
ג. שגיאת קומפילציה

5.

```
char s[5]={'H','e','l','l','o'},*p=s;
do {
  putchar(++(*p));
} while (*(++p));
```

א. ללא שגיאות
ב. שגיאת זמן ריצה
ג. שגיאת קומפילציה

הסברים:

1. בשורה השלישית – הניסיון לקדם את המשתנה arr ע"י ++arr אינו חוקי, מכיוון ש-arr הוא מטיפוס int[5] ועל כן לא ניתן לשנות את הכתובת עליה הוא מצביע.

2. במערך i יש 8 מקומות (i[0 .. 7]), המצביע p מאותחל לכתובת של i[4]. בשורה השנייה מנסים לכתוב לכתובת p+4, כלומר ל-i[8] שנמצאת מעבר לגבולות המערך. כתיבה למקום זה עלולה להסתיים בשגיאת זמן ריצה, מכיוון שהזכרון הזה אינו בהכרח מוקצה לתכנית.

3. בקטע הקוד הזה התקבלו שתי תשובות – גם א' וגם ג'. בעיקרון – לפי הסטנדרט של C אותו למדנו ההמרה double(i/2) אינה חוקית (צ"ל (double)(i/2)), ולכן התקבלה תשובה א. מצד שני – לא התכוונו להכשיל על דברים כאלו, ולכן קיבלנו גם את תשובה ג, מכיוון ש-(i/2) נותן 0 (חלוקה בשלמים של 1 ב-2), ולכן יתבצע i=(j-1)* שישנה את ערכו של i.

4. השורה ;h(f(n)) תגרום לשגיאת קומפילציה – הפונקציה f() אינה מחזירה דבר (void) ולכן לא ניתן להשתמש בערך ההחזרה שלה.

5. המערך s מאותחל לאותיות של המלה hello, אבל לא בצורה של מחרוזת. אי לכך לא יתווסף ה-'0\' בסוף המלה, ולולאת ה-while לא תעצור בסוף המערך. השורה ;putchar(++(*p)) משנה את הערך עליו מצביע p, ולכן כאשר נגלוש מתחומי המערך נתחיל לשנות זכרון שאינו מוקצה לתכנית שלנו מה שעלול לגרום לשגיאת זמן ריצה.