שאלה 1 (25 נקודות)

סעיף א

נתונה תוכנית ה-C הבאה, כאשר הפרמטר N הוא קבוע #define שלם וחיובי כלשהו:

```
int i = 2*N;


void strange(int *p)
{
   int j = *p;
   while (j) {
     printf("Backtracking is strange");
     j--;
   }
   *p = *p-1;
}


void charm()
{
   static int k = 8;
   int i;

   while (k) {
      printf("Arrays are charming");
      k = k/2;
   }
   i = 2*k;
}


int main ()
{
  while (i>0) {
    strange(&i);
    charm();
    i--;
  }
  return 0;
}
```

כתבו כמה פעמים תודפס כל אחת מן המחרוזות "Backtracking is strange" ו- "Arrays are charming!" כפונקציה של N (הכוונה למספר מדויק, לא לקירוב אסימפטוטי):

"Backtracking is strange": N^2+N "Arrays are charming": 4

הסבר:

Backtracking is strange: באיטרציה ה-i השורה מודפסת i פעמים. i מתחיל מ-2N ומופחת כל פעם בשתיים (פעם בתוך הלולאה ופעם בפונקציה strange, דרך המצביע p) ולכן נקבל כסך הפעמים שהשורה מודפסת סדרה חשבונית:

$$2N + (2N-2) + (2N-4) + \ldots 2 = N^2+N$$

הערה: במבחן ניתנו מלוא הנקודות גם למי שכתב את התוצאה כסכום ללא שימוש בנוסחת סכום סדרה חשבונית.

Arrays are charming: המשתנה k סטטי ולכן הוא מאותחל רק פעם אחת בתחילת ריצת התוכנית, וערכו נשמר בין הקריאות לפונקציה. יוצא שהשורה מודפסת רק 4 פעמים בקריאה הראשונה לפונקציה charm, ולאחר מכן כל קריאה נוספת אינה גורמת להדפסות נוספות.

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  i = 0;
int *j = *i;
```

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

הסבר: הפעלת האופרטור* על משתנה שאיננו מצביע.

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

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

3.
```
int *j;
int *i = j;
*i = 5;
```

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

הסבר: כתיבה לזיכרון (`*i`) שלא הוקצה.

4.
```
int *i, j, k;
i = &(j+k);
```

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

הסבר: הפעלת האופרטור & על ביטוי ולא על משתנה.

5.
```
char* s = "Hello World!";
while (*s) {
  *s = *s + 1;
}
```

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

הסבר: כתיבה לזיכרון שמוגדר כקריאה בלבד (אזור קבועים).