שאלה 4 (25 נקודות)

חברת התעופה Fly-PC פועלת ב-N ערים ברחבי עולם. החברה מפעילה קווי טיסה סדירים בין **חלק** מן הערים הללו, כאשר לכל טיסה נתון מחיר חיובי ממש. ערים שאין טיסה ישירה ביניהן, ניתן לעבור ביניהן באמצעות ביצוע מספר טיסות בזו אחר זו.

המטרה בשאלה זו הינה לכתוב פונקציה שבהינתן עיר מוצא ועיר יעד, מחשבות את המחיר המינימאלי הדרוש על מנת להגיע מעיר המוצא לעיר היעד. במידה ולא קיים מסלול מעיר המוצא לעיר היעד, על הפונקציה להחזיר 1-.

מחירי הטיסות נתונים במטריצה דו-ממדית E, כאשר התא E[i][j] מכיל את מחיר הטיסה מהעיר i לעיר j. אם אין טיסה בין i ל-j אז במטריצה יופיע 1- בתא המתאים. שימו לב שהמטריצה אינה בהכרח סימטרית: במילים אחרות, אם יש טיסה מ-i ל-j זה לא אומר שיש גם טיסה מ-j ל-i, וכן גם שאם יש טיסה כזאת, מחירה עשוי להיות שונה.

שימו לב:
- אנו מחפשים את המסלול **הזול ביותר** בין שתי הערים, ולא הקצר ביותר במחינת מספר הטיסות.
- בשאלה זו אין דרישות סיבוכיות.
- אין טיסות מעיר כלשהיא לעצמה, כלומר E[i][i] שווה 1- לכל i.

```
double cheapest_rate(double E[N][N], int from, int to) {
  return cheapest_rate_aux(E, from, to, N-1);
}

double cheapest_rate_aux(double E[N][N], int from, int to,
                         int maxlen) {

  double minprice = -1, route_price;
  int i;

  if (from == to) return 0;
  if (length == 0) return -1;

  for (i=0; i<N; i++) {

    if (E[from][i]<0)
      continue;  // no direct flight to i

    route_price = cheapest_rate_aux(E, i, to, maxlen-1);
    if (route_price<0)
      continue;  // can't continue from i to destination

    if (minprice < 0 || minprice > E[from][i]+route_price)
      minprice = E[from][i]+route_price;
  }
  return minprice;
}
```

המסלול הזול ביותר יוצא מ- from בטיסה ישירה לעיר ביניים כלשהי (בלתי ידועה) p, וממשיך ממנה ל-to במסלול הזול ביותר. על מנת למצוא את p, אנו עוברים על כל האפשרויות לטוס מ-from לעיר כלשהי i, מחשבים רקורסיבית את המחיר הזול ביותר מ-i ל-to, וסוכמים את שני המחירים. העיר p היא זו שמביאה את מחיר המסע הכולל למינימום. שימו לב שייתכן שהמסלול הזול ביותר הוא פשוט טיסה ישירה מ-from ל-to, ומקרה זה מתואר על ידי הבחירה p==to.

יש לזכור שרקורסיה רק עובדת כאשר הבעיה הרקורסיבית קטנה ממש מהבעיה המקורית – אחרת יש לנו מצב של רקורסיה אינסופית ללא הקטנת הבעיה. לפיכך, הוסף לפונקציה פרמטר maxlen שקובע את מספר הטיסות המקסימאלי שאנו מרשים על מנת להגיע מ-from ל-to. בכל קריאה רקורסיבית אנו מקטינים מספר זה באחת כיוון שהשתמשנו בטיסה אחת על מנת להגיע מ-from ל-i, ובכך הבעיה הוקטנה. אם הגענו ל-0, אין פתרון למצב זה ולכן מחזירים 1-.