שאלה 5 (20 נקודות)

הסיעות השונות במועצה מנסות להרכיב קואליציה חדשה. קואליציה תקפה הינה קבוצה של סיעות, אשר כל אחת מהן מוכנה לשתף פעולה עם כל יתר הסיעות בקבוצה, וכן, סך חברי כל הסיעות בקבוצה עולה על מחצית סך חברי המועצה.

במועצת הארגון יש N סיעות, ו-M חברים סך-הכל (N ו-M הם קבועים המוגדרים באמצעות #define).

נתון מערך parties המכיל את מספר חברי המועצה מכל סיעה – מספר חברי הסיעה ה-i במועצה מאוכסן בתא parties[i] .

כמו-כן נתונה מטריצה בשם relation, המתארת את היחסים בין הקבוצות השונות: התא relation[i][j] מכיל 1 אם ורק אם הסיעה ה-i מוכנה לשתף פעולה עם הסיעה ה-j, ו-0 אחרת. שימו לב כי המטריצה r אינה בהכרח סימטרית (ייתכן שסיעה i מוכנה לשתף-פעולה עם סיעה j , אך j אינה מוכנה לשתף פעולה עם i). ניתן להתעלם מן התאים באלכסון של המערך (כי כל סיעה מוכנה להשתתף בקואליציה עם עצמה).

עליכם לכתוב פונקציה בשם findc, אשר מחזירה **<u>את מספר החברים</u>** בקואליציה התקפה הגדולה ביותר האפשרית במועצה. אם לא קיימת אף קואליציה תקפה, על הפונקציה להחזיר 1-.

יש לפתור את הבעיה בשיטת BackTracking. פתרון רקורסיבי ללא BackTracking יקבל ניקוד חלקי בלבד.

<u>דוגמא</u>: עבור הקבועים הבאים:

```
#define N 4
#define M 120
```

ומטריצה relation הבאה:

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

ומערך parties הבא:

50
20
20
30

על הפונקציה להחזיר 90 (כי מפלגות 0,1,2 מוכנות להקים קואליציה, וסה"כ המנדטים שלהם הוא 90).

int findc (**unsigned int** parties[N], **unsigned int** relation [N][N]) {

```
  int chosen[N] = {0}, size;
  size = findc_aux(parties, relation, chosen, 0);
  return (size > M/2) ? size : -1;
}


int findc_aux(int parties[N], int relation[N][N],
              int chosen[N], int id)
{
  int i, sum, size1=0, size2=0;

  if (id==N) {
    sum = 0;
    for (i=0; i<N; ++i)
      sum += parties[i]*chosen[i];
    return sum;
  }

  if (islegal(chosen, relation, id)) {
    chosen[id] = 1;
    size1 = findc_aux(parties, relation, chosen, id+1);
    chosen[id] = 0;
  }
  size2 = findc_aux(parties, relation, chosen, id+1);
  return (size1 > size2) ? size1 : size2;
}


int islegal(int chosen[N], int relation[N][N], int id)
{
  int i;
  for (i=0; i<id; ++i) {
    if (chosen[i] && (relation[i][id]==0 ||
        relation[id][i]==0))
      return 0;
  }
  return 1;
}
```