שאלה 3 (20 נקודות)

סעיף א (10 נקודות)

נתון **מערך של מחרוזות** []strings באורך N (כש-N מוגדר כ-#define), **ומפתח** key שאף הוא מחרוזת. עליכם לממש פונקציה המחשבת לכל מחרוזת במערך strings כמה תווים משותפים יש לה עם המחרוזת key (כאשר חזרות נספרות, ראו דוגמה בהמשך). על הפונקציה לרשום את התוצאות למערך vals, כש-vals[i] מכיל את מספר התווים המשותפים למפתח עם המחרוזת ה-i. **שימו לב:** אין לשנות את מחרוזות הקלט.

לדוגמה, למחרוזת "Good morning Israel!" ולמפתח "oops" יש 3 תווים משותפים: שני 'o' ו-'s' אחד. לעומת זאת עם המפתח "!ooooops" יש לה 5 תווים משותפים: שלושה 'o', 's' אחד ו-'!' אחד.

דרישות סיבוכיות: סיבוכיות מקום נוסף O(1), וסיבוכיות זמן O(N*m+k) כש-m הוא אורך המחרוזת הארוכה ביותר ו-k הוא אורך המפתח. בשאלה זו ניתן להתייחס למספר התווים האפשריים (256) כקבוע.

```
void keycount(char* strings[N], char* key, int vals[N]) {

  int keycount[256]={0}, i, j;

  for (i=0; key[i]!=0; ++i)
    keycount[key[i]]++;

  for (j=0; j<N; ++j) {
    int strcount[256]={0};
    for (i=0; strings[j][i]!=0; ++i) {
      strcount[strings[j][i]]++;
    }
    vals[j]=0;
    for (i=0; i<256; ++i) {
      vals[j] += min(keycount[i], strcount[i]);
    }
  }
}

int min(int x, int y) {
  return (x<y ? x : y);
}
```

סעיף ב (10 נקודות)

בשאלה זו נתון מערך של מחרוזות ומפתח, כמתואר בסעיף א'. יש לממש פונקציה **שממיינת** את מערך המחרוזות בסדר עולה על פי ערכן של המחרוזות, כאשר ערכה של מחרוזת מוגדר כמספר התווים שמשותפים לה עם המפתח. בסעיף זה ניתן להשתמש בפונקציה מסעיף א' גם אם לא פתרתם אותו.

דרישות סיבוכיות: סיבוכיות מקום נוסף O(N). סיבוכיות זמן $O(N*m+k+N^2)$, כש-N, m ו-k כפי שהוגדרו בסעיף א'.

```
void keysort(char* strings[N], char* key) {

  int vals[N], i, j;

  keycount(strings, key, vals);

  for (i=0; i<N-1; ++i) {
    for (j=0; j<N-i-1; ++j) {
      if (vals[j] > vals[j+1]) {
        swapint(vals+j,  vals+j+1);
        swapstr(strings+j, strings+j+1);
      }
    }
  }
}

void swapint(int* a, int* b) {
  int tmp=*a; *a=*b; *b=tmp;
}

void swapstr(char** a, char** b) {
  char* tmp=*a; *a=*b; *b=tmp;
}
```