1 Esercizio Programmazione (6 Punti)

Completare le parti mancati del seguente codice C con tutte le istruzioni necessarie per il corretto funzionamento del programma. Inserire dei commenti nel codice per descrivere le operazioni effettuate.

```
 * Output:
 * - X(n,n): soluzione approssimata
 * - num_iter: numero di iterazioni
 * - err_k: differenza tra le ultime due approssimazioni in norma 1
 *
 */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

// Funzioni per la gestione delle matrice (da considerare implementate)
void LETTURA_MATRICE(int righe,int colonne, double A[righe][colonne],FILE *input);
void STAMPA_MATRICE(int righe, int colonne, double A[righe][colonne]);
void ZEROS_MATRICE(int righe, int colonne, double A[righe][colonne]);
void ZEROS_VETTORE(int num, double V[num]);
// Funzione per il calcolo della norma n di un vettore (da considerare implementata)
double NORMA_N_VETTORE(int num, double a[num], double ordine_norma);

int main()
{
  int i=0,j=0;
  int N=0;     /* Numero di equazioni */
  double e=0.;

  /* Lettura parametri di input */
  printf("Inserire il numero di equazioni N: ");
  scanf("%d", &N);
  printf("Inserire la soglia epsilon e: ");
  scanf("%lf", &e);

  /* Allocazione della matrice A e dei vettore B e X del sistema lineare */
    double A[N][N];
    double B[N][1];
    double X0[N];
    double X[N];
    double errore[N];
    /* Inizializzazione di A e B */
    ZEROS_MATRICE(N,N,A);
    ZEROS_MATRICE(N,1,B);
    ZEROS_VETTORE(N,X0);
    ZEROS_VETTORE(N,X);
    ZEROS_VETTORE(N,errore);

    /* Lettura da file dei valori della matrice A e del vettore B */
  FILE *inputA=NULL;
  FILE *inputB=NULL;

  inputA = fopen("matriceA.dat", "r");
  if(inputA == NULL)
  {
    printf("Errore nell'apertura del file matriceA.dat\n");
```

```
53      printf("Riprovare\n");
54      return(1);
55    } /* if */
56
57    inputB = fopen("vettoreB.dat", "r");
58      if(inputB == NULL)
59    {
60      printf("Errore nell'apertura del file vettoreB.dat\n");
61      printf("Riprovare\n");
62      return(1);
63    } /* if */
64
65    LETTURA_MATRICE(N,N,A,inputA);
66    LETTURA_MATRICE(N,1,B,inputB);
67
68    printf("La matrice A:\n");
69    STAMPA_MATRICE(N,N,A);
70      printf("Il vettore B:\n");
71    STAMPA_MATRICE(N,1,B);
72
73    double err_k;
74    int num_iter = 0;
75
76
77    while (err_k > e && num_iter < 100)
78    {
79
80  /*  ----------------PARTE MANCANTE----------------------------
81   *
82   * inserire tutte le operazioni necessarie per calcolare la soluzione
83   * del sistema lineare con il metodo di Gauss-Seidel
84   *
85   */
86
87    err_k = NORMA_N_VETTORE(N,errore,1.0);
88
89    }
90
91
92    printf("Soluzione:\n");
93    for (i=0;i<N;i++) printf("x%d = %12.8lf\n",i,X0[i]);
94    printf("Numero di iterazione effettuate: %d\n",num_iter);
95    printf("La norma uno del vettore errore %lf\n:", err_k);
96
97
98    return 0;
99  }
```

2 Esercizio Programmazione (4 Punti)

Scrivere una funzione per il calcolo della traccia di una matrice quadrata A[num,num]. Rispettare il seguente prototipo:

```
1 double TRACCIA_MATRICE(int num, double A[num][num]);
```