

Programmazione procedurale

Anno di corso 2019/2020

Progetto svolto da:

Giulio Maria Bianchi, matr. 310993, indirizzo mail: g.bianchi12@campus.uniurb.it

SVILUPPO SOFTWARE SECONDO LA METODOLOGIA “IN THE SMALL”

PROGRAMMA: SOMMA E PRODOTTO DI DUE MATRICI

Prima sezione

Specifica del problema:

Scrivere un programma ANSI C che acquisisce da tastiera due matrici di numeri reali allocandole dinamicamente e poi stampa a video la loro somma e il loro prodotto.

Seconda parte

Analisi del problema:

Input:

Prima matrice

Seconda matrice

Output:

Matrice somma

Matrice prodotto

Relazioni:

Somma diretta tra matrici secondo le regole dell'algebra, ciò è possibile se il numero delle righe della prima matrice corrisponde al numero di righe della seconda matrice e il numero di colonne della prima matrice corrisponde al numero di colonne della seconda matrice.

Prodotto diretto tra matrici secondo le regole dell'algebra, cioè il numero di colonne della prima matrice deve corrispondere al numero di righe della seconda matrice

Terza parte

Progettazione dell'algoritmo:

- i numeri reali vengono rappresentati tramite variabili di tipo double
- utilizzo di strutture di dati di tipo array bidimensionali con due variabili di lavoro che fungono da indici per scorrere gli elementi
- allocazione dinamica della memoria

I passi dell'algoritmo sono i seguenti:

- acquisire il numero di righe della prima matrice
- acquisire il numero colonne della prima matrice
- creare due matrici dinamiche- calcolare la somma degli elementi della prima matrice con gli elementi della seconda se è possibile in base alle regole dell'algebra
- calcolare il prodotto degli elementi della prima matrice con gli elementi della seconda matrice se è possibile in base alle regole dell'algebra
- comunicare la matrice prodotto e la matrice somma

Quarta parte

Implementazione dell'algoritmo:

CODICE DEL PROGETTO SOMMA E PRODOTTO TRA MATRICI DINAMICHE

```
/* inclusione delle librerie */

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

/* dichiarazione delle funzioni */

int numero_righe(int numero_matrice);

int numero_colonne(int numero_matrice);

double **crea_matrice(int righe,
                      int colonne);

double **somma_matrici(double **matrice1,
                      double **matrice2,
                      int righe1,
                      int righe2,
                      int colonne1,
                      int colonne2);

double **prodotto_matrici(double **matrice1,
                          double **matrice2,
                          int righe1,
                          int righe2,
                          int colonne1,
                          int colonne2);

void stampa_matrice(double **matrice,
                    int righe,
                    int colonne);

/* definizione della funzione main */
int main()
{
    /* dichiarazione delle variabili locali alla funzione */
    double **matrice1, /*input: prima matrice*/
            **matrice2, /*input: seconda matrice*/
    double **matrice_somma; /*input: numero righe prima matrice*/
            **matrice_prodotto; /*input: numero righe seconda matrice*/
    int righe1, /*input: numero righe prima matrice*/
        colonne1, /*input: numero colonne prima matrice*/
        righe2, /*input: numero righe seconda matrice*/
        colonne2; /*input: numero colonne seconda matrice*/
```

```

/*creare la prima matrice*/
righe1 = numero_righe(1);
colonne1 = numero_colonne(1);
matrice1 = crea_matrice(righe1,
                        colonne1);

/*creare la seconda matrice*/
righe2 = numero_righe(2);
colonne2 = numero_colonne(2);
matrice2 = crea_matrice(righe2,
                        colonne2);

/* stampare la prima matrice */
printf("Matrice 1 \n");
stampa_matrice(matrice1,
                righe1,
                colonne1);

/* stampare la seconda matrice */
printf("Matrice 2 \n");
stampa_matrice(matrice2,
                righe2,
                colonne2);

if(colonne1 != colonne2 || righe1 != righe2)
{
    printf("\n");
    printf("Non Ã possibile effettuare la somma \n");
    printf("\n");
}
else
{
    matrice_somma = somma_matrici(matrice1,
                                matrice2,
                                righe1,
                                righe2,
                                colonne1,
                                colonne2);

/* stamapre la matrice somma */
printf("Matrice Somma \n");
stampa_matrice(matrice_somma,
                righe1,
                colonne1);

free(matrice_somma);
matrice_somma = NULL;
}

if(colonne1 != righe2)
{
    printf("\n");
}

```

```

        printf("Non Ã possibile effettuare il prodotto \n");
        printf("\n");
    }
    else
    {
        matrice_prodotto = prodotto_matrici(matrice1,
                                            matrice2,
                                            righe1,
                                            righe2,
                                            colonne1,
                                            colonne2);

        /* stampare la matrice prodotto */
        printf("Matrice Prodotto \n");
        stampa_matrice(matrice_prodotto,
                        righe1,
                        colonne2);

        free(matrice_prodotto);
        matrice_prodotto = NULL;
    }

    /* libera area di memoria per la prima matrice */
    free(matrice1);
    matrice1 = NULL;

    /* libera area di memoria per la seconda matrice */
    free(matrice2);
    matrice2 = NULL;

    return 0;
}

/* definizione della funzione per l'acquisizione e la validazione stretta dell'input righe*/
int numero_righe(int numero_matrice)
{
    int numero;          /* lavoro: numero righe*/
    int esito_lettura;    /* lavoro: varaibile per effettuare validazione stretta dell'input */

    do
    {
        printf("Inserire il numero di righe della matrice %d \n",
               numero_matrice);

        esito_lettura = scanf("%d",
                              &numero);
        if(esito_lettura != 1 || numero <= 0)

        {
            printf("Errore, Inserire un input valido \n");
        }
        while(getchar() != '\n');
    }

```

```

    }while(esito_lettura !=1 || numero <=0);
    return(numero);
}

```

/ definizione della funzione per l'acquisizione e la validazione dell'input colonne*/*

```

int numero_colonne(int numero_matrice)
{
    int numero;           /* lavoro: numero colonne */
    int esito_lettura;     /* lavoro: varaibile per effettuare validazione stretta dell'input */

    do
    {
        printf("Inserire il numero di colonne della matrice %d \n",
            numero_matrice);

        esito_lettura = scanf("%d",
            &numero);
        if(esito_lettura != 1 || numero <= 0)
        {
            printf("Errore, Inserire un input valido \n");
        }
        while(getchar() !='\n');

    }while(esito_lettura !=1 || numero <=0);
    return(numero);
}

```

/ definizione della funzione per l'allocazione dinamica delle matrici */*

```

double **crea_matrice(int righe,
    int colonne)
{
    int i,           /* lavoro: indice scorrimento array */
        j;          /* lavoro: indice scorrimento array */
    int esito_lettura;
    double **matrice;
    matrice = (double**)calloc(righe,
        sizeof(double*));

    for(i = 0;
        (i < righe);
        i++)
    {
        matrice[i] = (double*)calloc(colonne,
            sizeof(double));
    }
    for(i=0;
        (i < righe);
        i++)
    {
        for(j=0;
            (j < colonne);
            j++)

```



```

    {
        do
        {
            printf("Inserisci il valore nella riga %d colonne %d : \n",
                    i+1,
                    j+1);
            esito_lettura = scanf("%lf",
                                &matrice[i][j]);
            if(esito_lettura !=1 || righe <= 0)
            {
                printf("Errore : Inserire un input valido");
            }
            while (getchar() !='\n');
        }
        while(esito_lettura !=1);
    }
}
return (matrice);
}

```

/ definizione della funzione per la somma di due matrici */*

```

double **somma_matrici(double **matrice1, /* input: prima matrice */
                      double **matrice2, /* input: seconda matrice */
                      int    righe1,    /* input: righe prima matrice */
                      int    righe2,    /* input: righe seconda matrice */
                      int    colonne1,  /* input: colonne prima matrice */
                      int    colonne2)  /* input: colonne seconda matrice */
{
    int    i, /* lavoro: indice scorrimento array */
           j; /* lavoro: indice scorrimento array */
    double **matrice_somma;
    matrice_somma = (double**)calloc(righe1,
                                     sizeof(double*));

    for(i = 0;
        (i < righe1);
        i++)
    {
        matrice_somma[i] = (double*)calloc(colonne1,
                                             sizeof(double*));

        for(j=0;
            (j<colonne1);
            j++)
        {
            matrice_somma[i][j] += matrice1[i][j] + matrice2[i][j];
        }
    }
    return(matrice_somma);
}

```

/ definizione della funzione per la moltiplicazione di due matrici */*

```

double **prodotto_matrici(double **matrice1, /* input: prima matrice */

```

```

        double **matrice2, /* input: seconda matrice */
        int     righe1,    /* input: righe prima matrice */
        int     righe2,    /* input: righe seconda matrice */
        int     colonne1,  /* input: colonne prima matrice */
        int     colonne2) /* input: colonne seconda matrice */
{
    int     i,             /* lavoro: indice scorrimento array */
           j,             /* lavoro: indice scorrimento array */
           c;             /* lavoro: indice scorrimento array */
    double **matrice_prodotto = 0;
    matrice_prodotto = (double**)calloc(righe1,
                                         sizeof(double)* 10);

    for(i = 0;
        (i < righe1);
        i++)
    {
        matrice_prodotto[i] = (double*)calloc(colonne2,
                                                sizeof(double) * 10);
    }

    for(i=0;
        (i<righe1);
        i++)
    {
        for(j=0;
            (j<colonne2);
            j++)
        {
            matrice_prodotto[i][j] = 0;

            for(c=0;
                (c<righe2);
                c++)
            {
                matrice_prodotto[i][j]+=(matrice1[i][c]*matrice2[c][j]);
            }
        }
    }
    return(matrice_prodotto);
}

```

/* definizione della funzione per stampare le matrici somma e prodotto */

```

void stampa_matrice(double **matrice, /* output: prima matrice */
                    int righe,        /* output: righe prima matrice */
                    int colonne)      /* output: colonne prima matrice */
{
    int i,           /* lavoro: indice scorrimento array */
        j;          /* lavoro: indice scorrimento array */

    for (i = 0;

```

```
(i< righe);
    i++)
{
    for (j = 0;
        (j< colonne);
        j++)
    {
        printf("{%lf}",
            matrice[i][j]);
    }

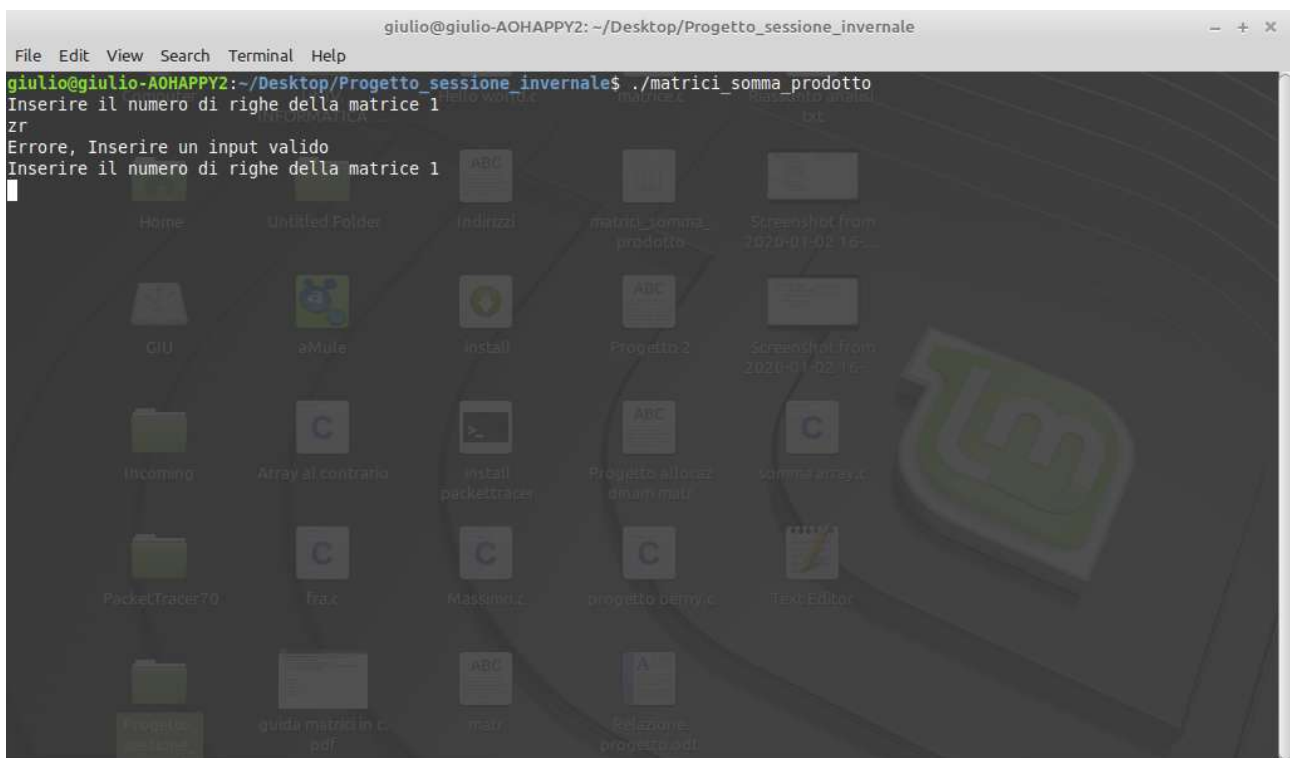
    printf("\n");
}
}
```

MAKEFILE

#Progetto matrici_somma_prodotto

matrici_somma_prodotto: matrici_somma_prodotto.c

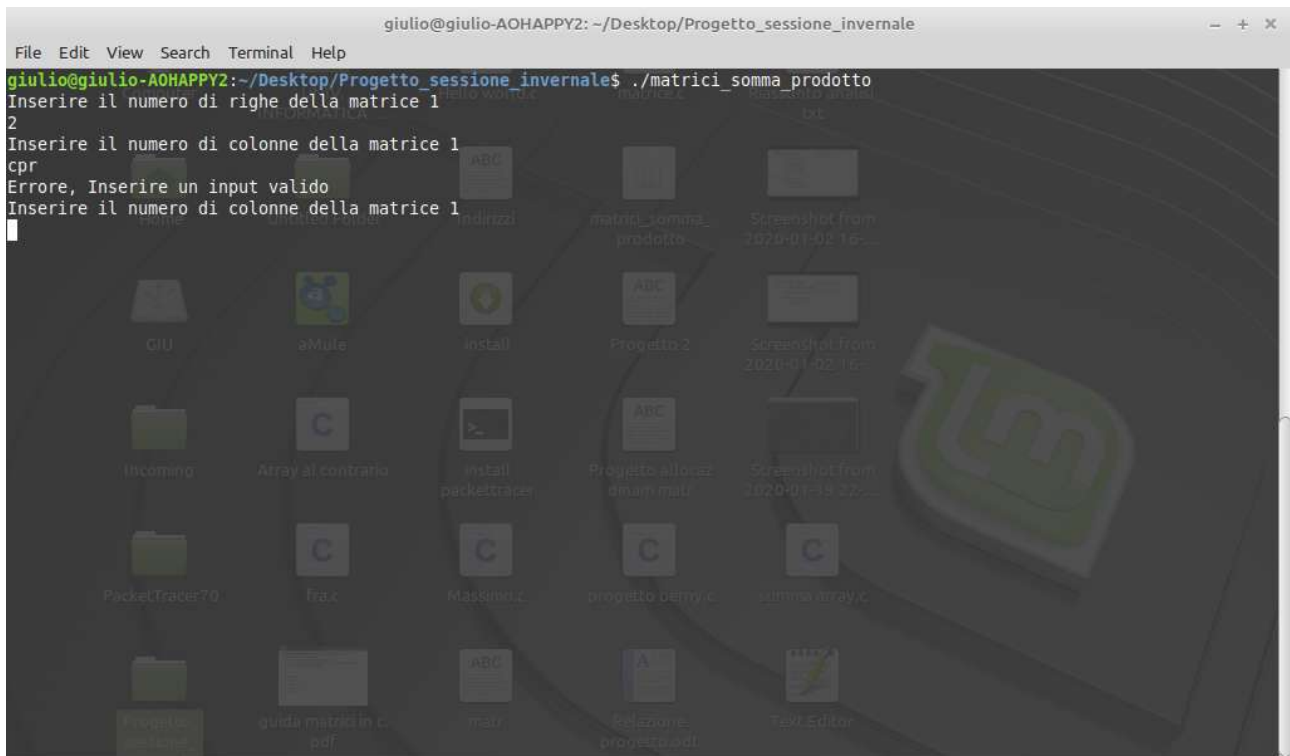
gcc -ansi -Wall -O matrici_somma_prodotto.c -o matrici_somma_prodotto



Test 2

Dati di ingresso: inserimento di caratteri quando il programma chiede all'utente di scegliere il numero di righe della seconda matrice.

Risultato prodotto:



```
giulio@giulio-AOHAPPY2: ~/Desktop/Progetto_sessione_invernale
File Edit View Search Terminal Help
giulio@giulio-AOHAPPY2:~/Desktop/Progetto_sessione_invernale$ ./matrici_somma_prodotto
Inserire il numero di righe della matrice 1
2
Inserire il numero di colonne della matrice 1
cpr
Errore, Inserire un input valido
Inserire il numero di colonne della matrice 1
█
```

Test 3

Dati di ingresso: inserimento di una matrice 2 X 2.

Risultato prodotto:

```
giulio@giulio-AOHAPPY2: ~/Desktop/Progetto_sessione_invernale
File Edit View Search Terminal Help
giulio@giulio-AOHAPPY2:~/Desktop/Progetto_sessione_invernale$ ./matrici_somma_prodotto
Inserire il numero di righe della matrice 1
2
Inserire il numero di colonne della matrice 1
2
Inserisci il valore nella riga 0 colonne 0 :
2
Inserisci il valore nella riga 0 colonne 1 :
2
Inserisci il valore nella riga 1 colonne 0 :
2
Inserisci il valore nella riga 1 colonne 1 :
2
Inserire il numero di righe della matrice 2
2
Inserire il numero di colonne della matrice 2
2
Inserisci il valore nella riga 0 colonne 0 :
2
Inserisci il valore nella riga 0 colonne 1 :
2
Inserisci il valore nella riga 1 colonne 0 :
2
Inserisci il valore nella riga 1 colonne 1 :
2
Matrice 1
|2.000000|2.000000|
|2.000000|2.000000|
Matrice 2
|2.000000|2.000000|
|2.000000|2.000000|
Matrice somma
|4.000000|0.000000|
|4.000000|7.999971|
Matrice prodotto
|8.000000|8.000000|
|8.000000|8.000000|
giulio@giulio-AOHAPPY2:~/Desktop/Progetto_sessione_invernale$
```

Test 4

Dati di ingresso: inserimento di una matrice 2 X 2 e 1 X 1.

Risultato prodotto:

```
giulio@giulio-AOHAPPY2: ~/Desktop/Progetto_sessione_invernale
File Edit View Search Terminal Help
giulio@giulio-AOHAPPY2:~/Desktop/Progetto_sessione_invernale$ ./matrici_somma_prodotto
Inserire il numero di righe della matrice 1
2
Inserire il numero di colonne della matrice 1
2
Inserisci il valore nella riga 0 colonne 0 :
2
Inserisci il valore nella riga 0 colonne 1 :
2
Inserisci il valore nella riga 1 colonne 0 :
2
Inserisci il valore nella riga 1 colonne 1 :
2
Inserire il numero di righe della matrice 2
1
Inserire il numero di colonne della matrice 2
1
Inserisci il valore nella riga 0 colonne 0 :
1
Matrice 1
|2.000000|2.000000|
|2.000000|2.000000|
Matrice 2
|1.000000|
Non è possibile effettuare la somma
Non è possibile effettuare il prodotto
giulio@giulio-AOHAPPY2:~/Desktop/Progetto_sessione_invernale$
```

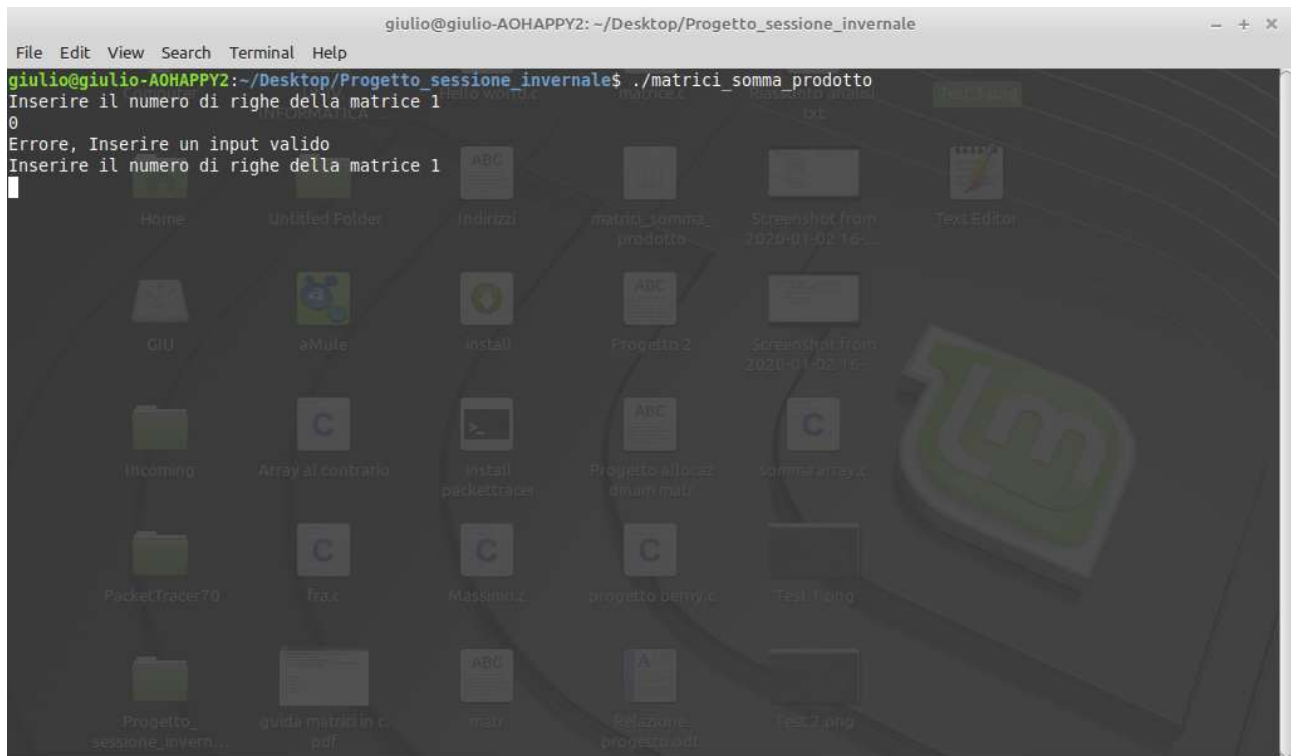
```
giulio@giulio-AOHAPPY2: ~/Desktop/Progetto_sessione_invernale
File Edit View Search Terminal Help
giulio@giulio-AOHAPPY2:~/Desktop/Progetto_sessione_invernale$ ./matrici_somma_prodotto
Inserire il numero di righe della matrice 1
0
Errore, Inserire un input valido
Inserire il numero di righe della matrice 1

```


Test 5

Dati di ingresso: inserimento del numero zero quando il programma chiede all'utente di scegliere il numero di righe della prima matrice.

Risultato prodotto:



```
giulio@giulio-AOHAPPY2: ~/Desktop/Progetto_sessione_invernale
File Edit View Search Terminal Help
giulio@giulio-AOHAPPY2:~/Desktop/Progetto_sessione_invernale$ ./matrici_somma_prodotto
Inserire il numero di righe della matrice 1
0
Errore, Inserire un input valido
Inserire il numero di righe della matrice 1
█
```

Test 6

Dati in ingresso: inserimento di un carattere quando il programma chiede all'utente di inserire un valore all'interno della prima matrice.

Risultato prodotto:

```
giulio@giulio-AOHAPPY2: ~/Desktop/Progetto_sessione_invernale
File Edit View Search Terminal Help
giulio@giulio-AOHAPPY2:~/Desktop/Progetto_sessione_invernale$ ./matrici_somma prodotto
Inserire il numero di righe della matrice 1
2
Inserire il numero di colonne della matrice 1
2
Inserisci il valore nella riga 0 colonne 0 :
1
Inserisci il valore nella riga 0 colonne 1 :
2
Inserisci il valore nella riga 1 colonne 0 :
3
Inserisci il valore nella riga 1 colonne 1 :
4
Inserire il numero di righe della matrice 2
2
Inserire il numero di colonne della matrice 2
2
Inserisci il valore nella riga 0 colonne 0 :
a
Errore : Inserire un input validoInserisci il valore nella riga 0 colonne 0 :
```

Test 7

Dati in ingresso: inserimento di una matrice 2 X 3 ed una 5 X 2.

Risultato prodotto:

```
giulio@giulio-AOHAPPY2: ~/Desktop/Progetto_sessione_invernale
File Edit View Search Terminal Help
giulio@giulio-AOHAPPY2:~/Desktop/Progetto_sessione_invernale$ ./matrici_somma_prodotto
Inserire il numero di righe della matrice 1
2
Inserire il numero di colonne della matrice 1
3
Inserisci il valore nella riga 0 colonne 0 : hello world.c
1
Inserisci il valore nella riga 0 colonne 1 :
2
Inserisci il valore nella riga 1 colonne 0 : ABC
3
Inserisci il valore nella riga 1 colonne 1 : Indirizzo
4
Inserire il numero di righe della matrice 2
5
Inserire il numero di colonne della matrice 2
2
Inserisci il valore nella riga 0 colonne 0 : install
1
Inserisci il valore nella riga 0 colonne 1 :
2
Inserisci il valore nella riga 0 colonne 2 :
3
Inserisci il valore nella riga 0 colonne 3 : install packettracer
4
Inserisci il valore nella riga 0 colonne 4 :
5
Inserisci il valore nella riga 1 colonne 0 :
6
Inserisci il valore nella riga 1 colonne 1 :
7
```

```
giulio@giulio-AOHAPPY2: ~/Desktop/Progetto_sessione_invernale
File Edit View Search Terminal Help
Inserisci il valore nella riga 1 colonne 2 :
8
Inserisci il valore nella riga 1 colonne 3 :
9
Inserisci il valore nella riga 1 colonne 4 :
1
Inserisci il valore nella riga 2 colonne 0 :
2
Inserisci il valore nella riga 2 colonne 1 :
3
Inserisci il valore nella riga 2 colonne 2 :
4
Inserisci il valore nella riga 2 colonne 3 :
5
Inserisci il valore nella riga 2 colonne 4 :
6
Inserisci il valore nella riga 3 colonne 0 :
7
Inserisci il valore nella riga 3 colonne 1 :
8
Inserisci il valore nella riga 3 colonne 2 :
1
Inserisci il valore nella riga 3 colonne 3 :
2
Inserisci il valore nella riga 3 colonne 4 :
3
Inserisci il valore nella riga 4 colonne 0 :
4
Inserisci il valore nella riga 4 colonne 1 :
5
Inserisci il valore nella riga 4 colonne 2 :
6
```

```
giulio@giulio-AOHAPPY2: ~/Desktop/Progetto_sessione_invernale
File Edit View Search Terminal Help
Inserisci il valore nella riga 3 colonne 2 :
1
Inserisci il valore nella riga 3 colonne 3 :
2
Inserisci il valore nella riga 3 colonne 4 :
3
Inserisci il valore nella riga 4 colonne 0 :
4
Inserisci il valore nella riga 4 colonne 1 :
5
Inserisci il valore nella riga 4 colonne 2 :
6
Inserisci il valore nella riga 4 colonne 3 :
7
Inserisci il valore nella riga 4 colonne 4 :
8
Matrice 1
|1.000000|2.000000|
|3.000000|4.000000|
Matrice 2
|1.000000|2.000000|3.000000|4.000000|6.000000|
|6.000000|7.000000|8.000000|9.000000|2.000000|
|2.000000|3.000000|4.000000|5.000000|7.000000|
|7.000000|8.000000|1.000000|2.000000|4.000000|
|4.000000|5.000000|6.000000|7.000000|8.000000|
Non è possibile effettuare la somma
Non è possibile effettuare il prodotto
giulio@giulio-AOHAPPY2:~/Desktop/Progetto_sessione_invernale$ ./matrici_somma_prodotto
```

Test 8

Dati di ingresso: inserimento di un numero negativo quando il programma chiede all'utente di inserire il numero di righe di una matrice.

Risultato ricevuto:

```
giulio@giulio-AOHAPPY2: ~/Desktop/Progetto_sessione_invernale
File Edit View Search Terminal Help
giulio@giulio-AOHAPPY2:~/Desktop/Progetto_sessione_invernale$ ./matrici_somma_prodotto
Inserire il numero di righe della matrice 1
2
Inserire il numero di colonne della matrice 1
2
Inserisci il valore nella riga 0 colonne 0 :
1
Inserisci il valore nella riga 0 colonne 1 :
2
Inserisci il valore nella riga 1 colonne 0 :
-2
Inserisci il valore nella riga 1 colonne 1 :
-2
Inserire il numero di righe della matrice 2
-2
Errore, Inserire un input valido
Inserire il numero di righe della matrice 2
```

Test 9

Dati in ingresso: inserimento di numeri negativi all'interno delle due matrici

Risultato prodotto:

```
giulio@giulio-AOHAPPY2: ~/Desktop/Progetto_sessione_invernale
File Edit View Search Terminal Help

giulio@giulio-AOHAPPY2:~/Desktop/Progetto_sessione_invernale$ ./matrici_somma_prodotto
Inserire il numero di righe della matrice 1
2
Inserire il numero di colonne della matrice 1
2
Inserisci il valore nella riga 0 colonne 0 :
1
Inserisci il valore nella riga 0 colonne 1 :
2
Inserisci il valore nella riga 1 colonne 0 :
3
Inserisci il valore nella riga 1 colonne 1 :
4
Inserire il numero di righe della matrice 2
2
Inserire il numero di colonne della matrice 2
2
Inserisci il valore nella riga 0 colonne 0 :
-1
Inserisci il valore nella riga 0 colonne 1 :
-2
Inserisci il valore nella riga 1 colonne 0 :
-3
Inserisci il valore nella riga 1 colonne 1 :
-4
Matrice 1
|1.000000|2.000000|
|3.000000|4.000000|
Matrice 2
|-1.000000|-2.000000|
|-3.000000|-4.000000|
Matrice somma
|0.000000|0.000000|
|0.000000|0.000000|
Matrice prodotto
|-7.000000|-10.000000|
|-15.000000|-22.000000|
giulio@giulio-AOHAPPY2:~/Desktop/Progetto_sessione_invernale$ clear
```


Test 10

Dati di ingresso: inserimento di una matrice 3 X 2 e 2 X 4.

Risultato prodotto:

```
giulio@giulio-AOHAPPY2: ~/Desktop/Progetto_sessione_invernale
File Edit View Search Terminal Help
giulio@giulio-AOHAPPY2:~/Desktop/Progetto_sessione_invernale$ ./matrici_somma_prodotto
Inserire il numero di righe della matrice 1
3
Inserire il numero di colonne della matrice 1
2
Inserisci il valore nella riga 0 colonne 0 :
3
Inserisci il valore nella riga 0 colonne 1 :
2
Inserisci il valore nella riga 0 colonne 2 :
1
Inserisci il valore nella riga 1 colonne 0 :
3
Inserisci il valore nella riga 1 colonne 1 :
2
Inserisci il valore nella riga 1 colonne 2 :
1
Inserisci il valore nella riga 2 colonne 0 :
3
Inserisci il valore nella riga 2 colonne 1 :
2
Inserisci il valore nella riga 2 colonne 2 :
1
Inserire il numero di righe della matrice 2
2
Inserire il numero di colonne della matrice 2
4
Inserisci il valore nella riga 0 colonne 0 :
1
Inserisci il valore nella riga 0 colonne 1 :
2
```

```
giulio@giulio-AOHAPPY2: ~/Desktop/Progetto_sessione_invernale
File Edit View Search Terminal Help
Inserisci il valore nella riga 2 colonne 0 :
3
Inserisci il valore nella riga 2 colonne 1 :
2
Inserisci il valore nella riga 2 colonne 2 :
1
Inserire il numero di righe della matrice 2
2
Inserire il numero di colonne della matrice 2
4
Inserisci il valore nella riga 0 colonne 0 :
1
Inserisci il valore nella riga 0 colonne 1 :
2
Inserisci il valore nella riga 1 colonne 0 :
3
Inserisci il valore nella riga 1 colonne 1 :
4
Matrice 1
|3.000000|2.000000|1.000000|
|3.000000|2.000000|1.000000|
|3.000000|2.000000|1.000000|
Matrice 2
|1.000000|2.000000|
|3.000000|4.000000|
Non è possibile effettuare la somma
Matrice prodotto
|9.000000|14.000000|
|9.000000|14.000000|
giulio@giulio-AOHAPPY2:~/Desktop/Progetto_sessione_invernale$
```

Parte sesta

Verifica della correttezza del programma

```
for(i=0;
    (i<righe1);
    i++)
{
    for(j=0;
        (j<colonne2);
        j++)
    {
        matrice_prodotto[i][j] = 0;

        for(c=0;
            (c<righe2);
            c++)
        {
            matrice_prodotto[i][j]+=(matrice1[i][c]*matrice2[c][j]);
        }
    }
}
```

- Formalizzo la POSTCONDIZIONE $R = (\text{matrice_prodotto}[i][j] = \sum_{(i=0)(j=0)}^{\text{righe}, \text{colonne}} a[(i)(j)] \quad)$

- posso rendere la tripla vera ponendo come PRECONDIZIONE vero in quanto:

$$P = [0 \leq i \leq \text{righe}] \quad (0 \leq j \leq \text{colonne}) \quad \wedge \quad (\text{matrice_prodotto}[i][j] = \sum_{(i=0)(j=0)}^{\text{righe}, \text{colonne}} a[(i)(j)])$$

- e la FUNZIONE: $\text{tr}(i) = [(\text{colonne} - i) \quad (\text{righe} - j)]$

soddisfano le ipotesi del teorema dell'invariante di ciclo in quanto:

1) INVARIANZA:

$\{ P \wedge (i \leq \text{colonne}; j \leq \text{righe}) \} , \text{matrice_prodotto}[i][j]+=(\text{matrice1}[i][c]*\text{matrice2}[c][j]); \{ P \}$

$$\begin{aligned} P_{i,i+1;j,j+1} &= (0 \leq i + 1 \leq \text{righe}; \\ &\quad 0 \leq j + 1 \leq \text{colonne}) \quad \wedge \quad \text{matrice_prodotto}[i][j]+=(\text{matrice1}[i][c]*\text{matrice2}[c][j]) \\ &\equiv (0 \leq i + 1 \leq \text{righe}; \\ &\quad 0 \leq j + 1 \leq \text{colonne}) \quad \wedge \quad \sum_{(i=0)(j=0)}^{\text{righe}, \text{colonne}} a[(i)(j)] \end{aligned}$$

e, denotato con P^1 quest'ultimo predicato, da:

$$P^1_{\text{righe}, \text{righe}+1; \text{colonne}, \text{colonne}+1} = (0 \leq i + 1 \leq \text{righe};$$

$$0 \leq j + 1 \leq \text{colonne}) \quad \wedge \quad a[k] \left(\sum_{(i=0)(j=0)}^{\text{righe}, \text{colonne}} a[(i)(j)] \right)$$

2) PROGRESSO è garantito dal fatto che la funzione tr(i) decresce di un'unità ad ogni iterazione di entrambi i cicli in quanto ambedue vengono incrementati di un'unità fino al raggiungimento della condizione di terminazione

3) LIMITATEZZA segue da:

$$(P \wedge \text{tr}(i) \leq 0) = [(0 \leq i \leq \text{righe}; \quad [\text{righe} - i \leq 0; \\ 0 \leq j \leq \text{colonne}]] \wedge \left(\sum_{(i=0)(j=0)}^{\text{righe}-1, \text{colonne}-1} a[(i)(j)] \right) \wedge; \text{colonne} - j \leq 0$$

$$\equiv [i = \text{righe}; \\ j = \text{colonne}] \wedge \sum_{(i=0)(j=0)}^{\text{righe}-1, \text{colonne}-1} a[(i)(j)]$$