

## Prima prova di esonero – 2 novembre 2009

La prova dura tre ore, durante le quali non è possibile allontanarsi dall'aula, se non dopo aver consegnato l'elaborato scritto. Per superare la prova di esonero e poter sostenere la successiva è necessario ottenere almeno 15 punti. È possibile utilizzare libri e appunti personali, senza scambiarli con altri studenti. I compiti che presenteranno evidenti ed anomale "similitudini" saranno annullati.

### Esercizio n. 1 (max 10 punti)

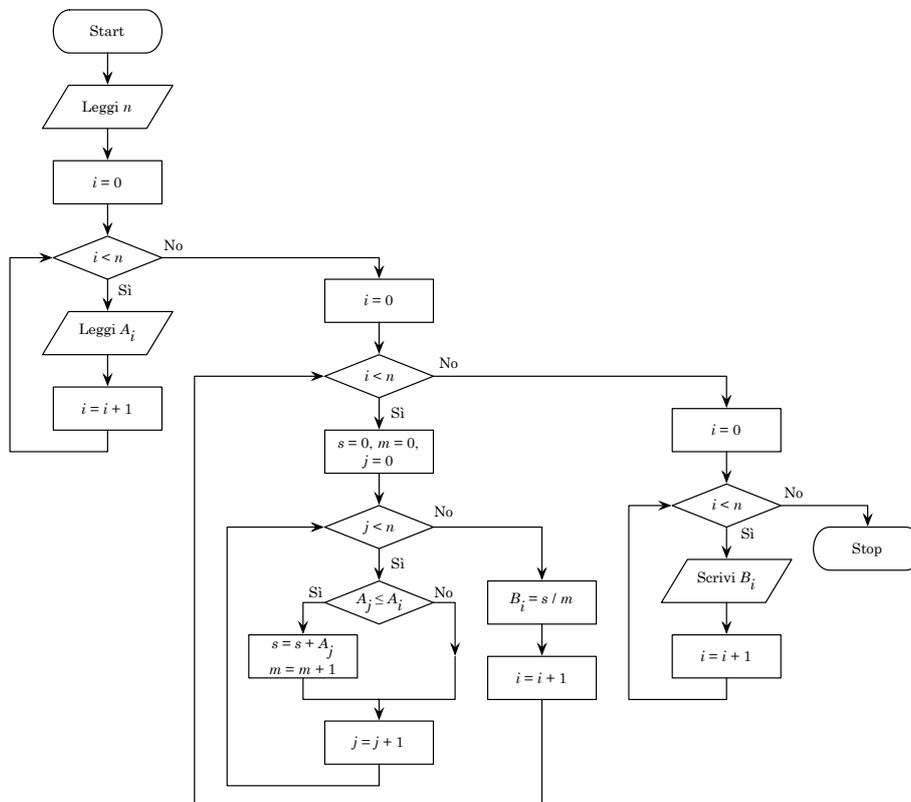
Risolvere il seguente problema proponendo una pseudo-codifica dell'algoritmo, il diagramma di flusso ed infine la codifica in linguaggio C del programma che implementa l'algoritmo stesso.

Leggere in input  $n$  numeri interi e memorizzarli in un vettore  $A$ . Costruire un vettore  $B$  con  $n$  elementi in modo tale che  $B_i$  sia dato dalla media aritmetica degli elementi  $A_j \leq A_i$  ( $j = 0, 1, \dots, n - 1$ ). Stampare il vettore  $B$ .

**Esempio** Supponiamo che vengano acquisiti in input  $n = 5$  numeri memorizzati nel vettore  $A = (3, 7, 12, 6, 2)$ . Allora il vettore  $B$  è il seguente:  $B = (2.5, 4.5, 6.0, 3.666, 2.0)$ .

### Soluzione

#### Diagramma di flusso



### Pseudo-codifica dell'algoritmo

```
1: leggi  $n$ 
2: leggi  $A_0, A_1, \dots, A_{n-1}$ 
3:  $i = 0$ 
4: fintanto che  $i < n$  ripeti
5:    $s = 0, m = 0, j = 0$ 
6:   fintanto che  $j < n$  ripeti
7:     se  $A_j \leq A_i$  allora
8:        $s = s + A_j, m = m + 1$ 
9:     fine-condizione
10:     $j = j + 1$ 
11:   fine-ciclo
12:    $B_i = s/m, i = i + 1$ 
13: fine-ciclo
14: scrivi  $B_0, B_1, \dots, B_{n-1}$ 
```

### Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 int main(void) {
6     int A[MAX], i, j, n, m, s;
7     float B[MAX];
8     printf("Numero di elementi: ");
9     scanf("%d", &n);
10    printf("Inserisci %d numeri interi: ", n);
11    for (i=0; i<n; i++)
12        scanf("%d", &A[i]);
13    for (i=0; i<n; i++) {
14        s = 0;
15        m = 0;
16        for (j=0; j<n; j++)
17            if (A[j]<A[i]) {
18                s = s+A[j];
19                m = m+1;
20            }
21        B[i] = (float)s/m;
22    }
23    for (i=0; i<n; i++)
24        printf("%f ", B[i]);
25    return(0);
26 }
```

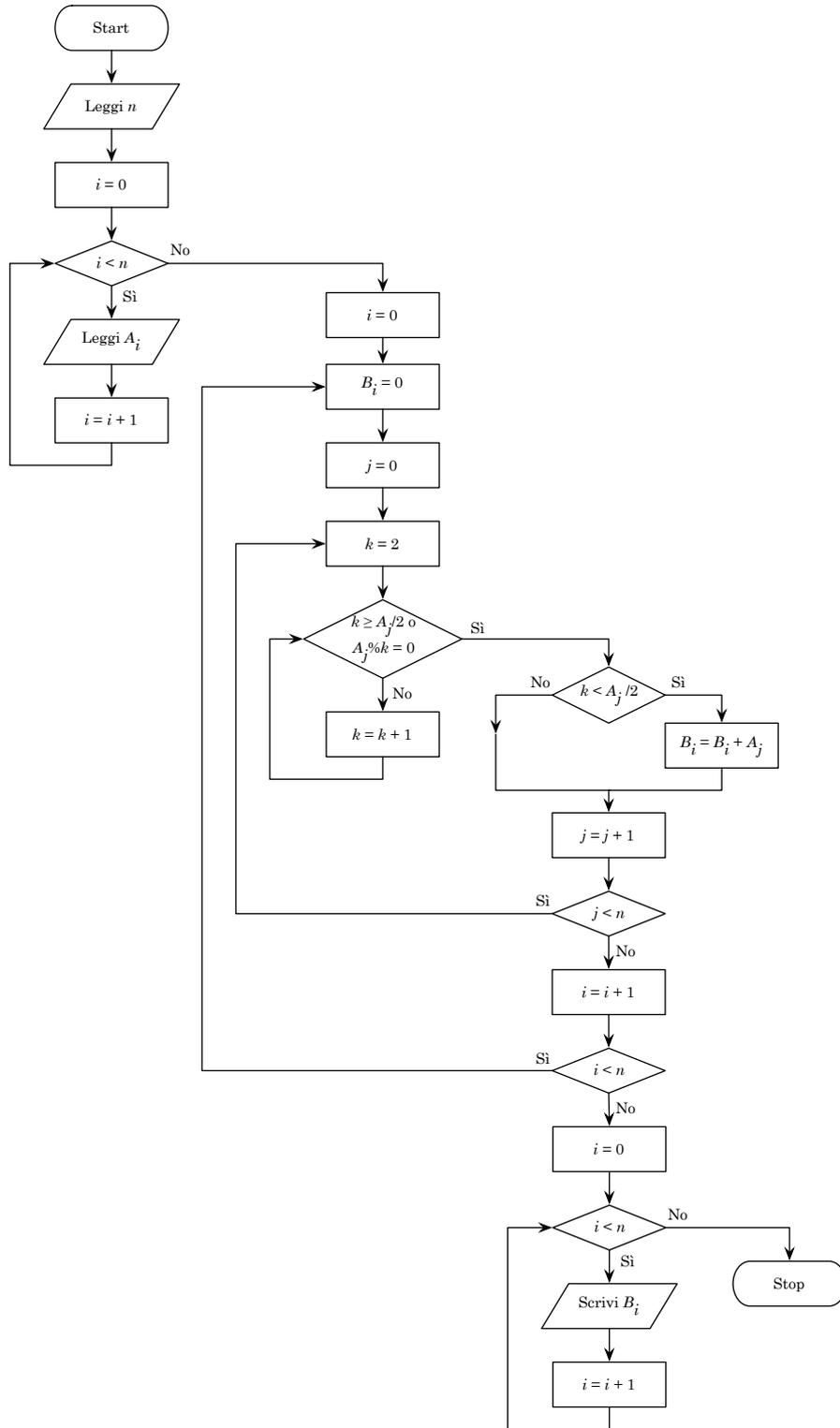
## Esercizio n. 2 (max 8 punti)

Scrivere il diagramma di flusso e la codifica in linguaggio C (un programma completo) del seguente algoritmo:

- 1: leggi  $n$
- 2: leggi  $n$  numeri interi positivi e memorizzali nel vettore  $A$
- 3:  $i = 0$
- 4:  $B_i = 0$
- 5:  $j = 0$
- 6:  $k = 2$
- 7: se  $k \geq A_j/2$  o  $A_j$  è divisibile per  $k$  vai al passo 10
- 8:  $k = k + 1$
- 9: vai al passo 7
- 10: se  $k < A_j/2$  allora  $B_i = B_i + A_j$
- 11:  $j = j + 1$
- 12: se  $j < n$  allora vai al passo 6
- 13:  $i = i + 1$
- 14: se  $i < n$  allora vai al passo 4
- 15:  $i = 0$
- 16: se  $i \geq n$  allora vai al passo 20
- 17: scrivi  $B_i$
- 18:  $i = i + 1$
- 19: vai al passo 16
- 20: fermati

## Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 int leggi_array(int v[]) {
6     int i, n;
7     printf("Numero di elementi: ");
8     scanf("%d", &n);
9     printf("Inserisci %d numeri interi: ", n);
10    for (i=0; i<n; i++)
11        scanf("%d", &v[i]);
12    return(n);
13 }
14
15 void stampa_array(int v[], int n) {
16     int i;
17     for (i=0; i<n; i++)
18         printf("%d ", v[i]);
19     printf("\n");
20     return;
21 }
22
23 int main(void) {
24     int A[MAX], B[MAX], n, i, j, k;
25     n = leggi_array(A);
26     i = 0;
27     do {
28         B[i] = 0;
29         j = 0;
30         do {
31             k = 2;
32             while (k < A[j]/2 && A[j]%k != 0)
33                 k = k+1;
34             if (k<A[j]/2)
35                 B[i] = B[i] + A[j];
36             j = j+1;
37         } while (j<n);
38         i = i+1;
39     } while (i<n);
40     stampa_array(B, n);
41     return(0);
42 }
```

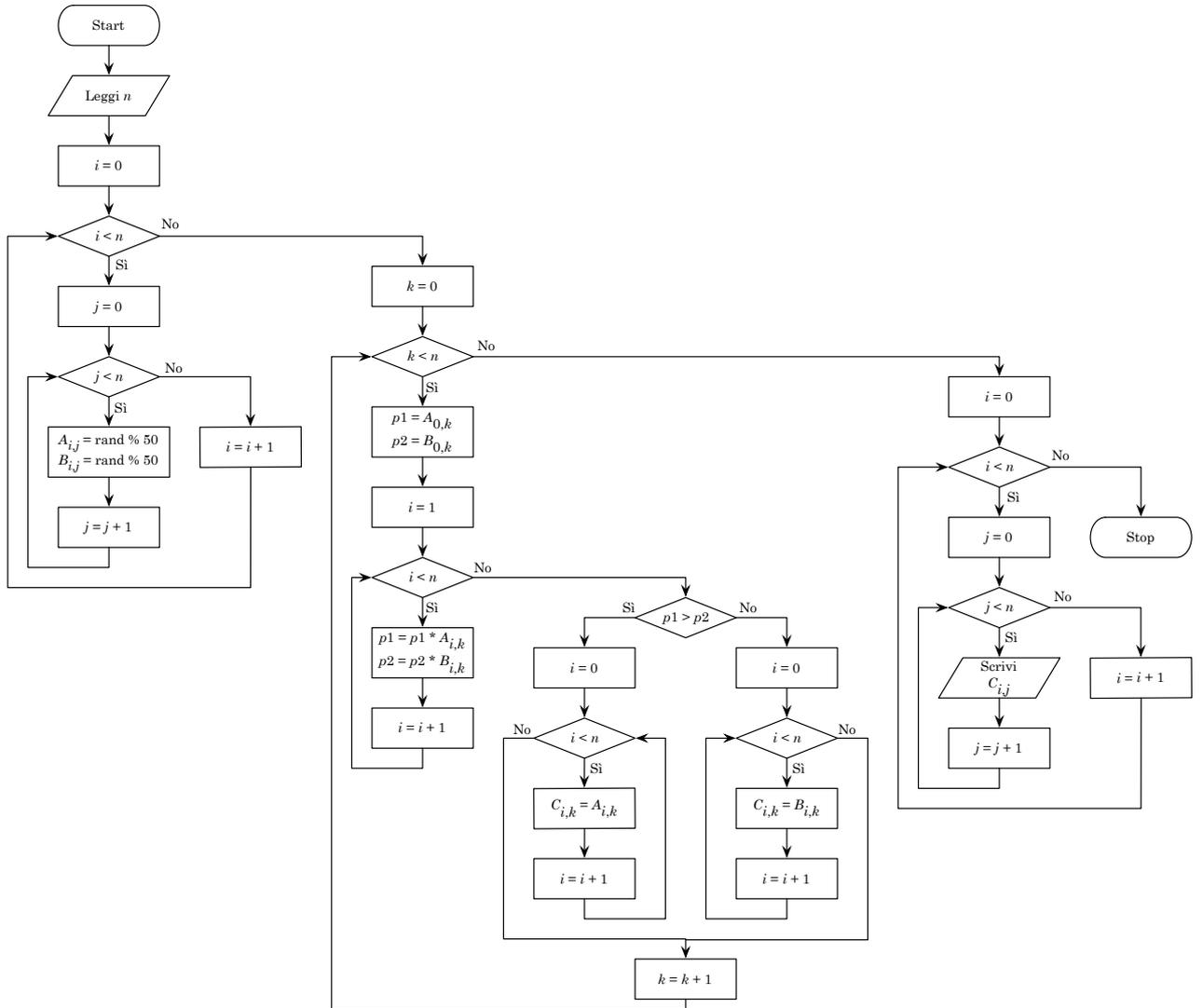


### Esercizio n. 3 (max 6 punti)

Utilizzando le regole della programmazione strutturata, disegnare il diagramma di flusso della seguente funzione codificata in linguaggio C.

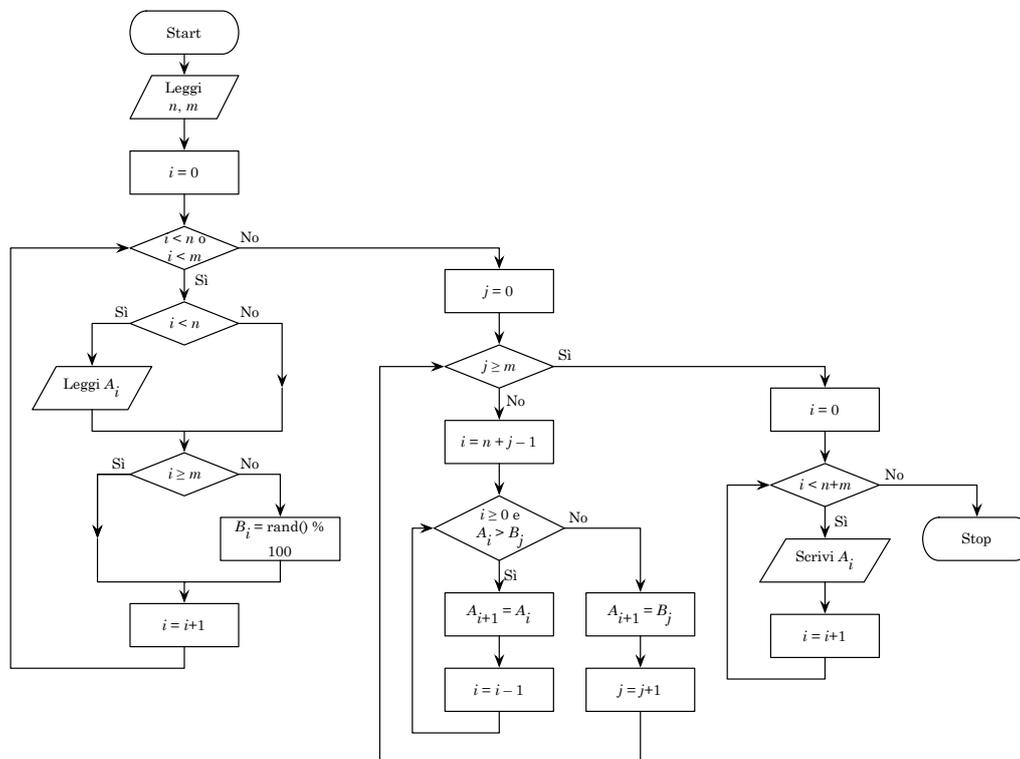
```
1 int main(void) {
2     int A[MAX][MAX], B[MAX][MAX], C[MAX][MAX], i, j, k, n, p1, p2;
3     srand((unsigned)time(NULL));
4     printf("Inserisci l'ordine delle matrici quadrate: ");
5     scanf("%d", &n);
6     for (i=0; i<n; i++)
7         for (j=0; j<n; j++) {
8             A[i][j] = rand() % 50;
9             B[i][j] = rand() % 50;
10        }
11    for (k=0; k<n; k++) {
12        p1 = A[0][k];
13        p2 = B[0][k];
14        for (i=1; i<n; i++) {
15            p1 = p1 * A[i][k];
16            p2 = p2 * B[i][k];
17        }
18        if (p1>p2)
19            for (i=0; i<n; i++)
20                C[i][k] = A[i][k];
21        else
22            for (i=0; i<n; i++)
23                C[i][k] = B[i][k];
24    }
25    for (i=0; i<n; i++) {
26        for (j=0; j<n; j++)
27            printf("%2d ", C[i][j]);
28        printf("\n");
29    }
30    return(0);
31 }
```

# Soluzione



## Esercizio n. 4 (max 6 punti)

Codificare in linguaggio C un programma che implementi l'algoritmo descritto dal seguente diagramma di flusso. Nella codifica porre attenzione anche alla definizione delle variabili.



## Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 int main(void) {
4     int n, m, i, j, A[100], B[100];
5     scanf("%d %d", &n, &m);
6     for (i=0; i<n || i<m; i++) {
7         if (i<n)
8             scanf("%d", &A[i]);
9         if (i<m)
10            B[i] = rand() % 100;
11    }
12    for (j=0; j<m; j++) {
13        for (i=n+j-1; i>=0 && A[i]>B[j]; i--)
14            A[i+1] = A[i];
15        A[i+1] = B[j];
16    }
17    for (i=0; i<n+m; i++)
18        printf("%d ", A[i]);
19    return(0);
20 }
```