

Prima prova di esonero – 7 novembre 2016

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.

Deve essere consegnata solo la “bella copia” del compito scritto; su ciascun foglio deve essere riportato il **nome**, il **cognome** e il **numero di matricola** (o un altro codice identificativo di fantasia) dello studente.

Esercizio n. 1 (max 8 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.

Letti in input tre interi positivi x , y e z , generare una matrice quadrata A di z righe e z colonne, con numeri interi casuali compresi tra il minimo e il massimo tra x e y (estremi inclusi). Stampare la matrice. Stampare la colonna della matrice con il maggior numero di elementi consecutivi disposti in ordine crescente.

Esempio Sia $x = 23$, $y = 5$ e $z = 6$. Si consideri la seguente matrice di numeri casuali:

$$A = \begin{pmatrix} 12 & 17 & 19 & 5 & 14 & 6 \\ 9 & \mathbf{15} & 18 & 11 & 21 & 12 \\ 7 & \mathbf{16} & 15 & 8 & 10 & 13 \\ 5 & \mathbf{19} & 20 & 13 & 7 & 9 \\ 11 & \mathbf{23} & 21 & 6 & 18 & 19 \\ 8 & 8 & 15 & 5 & 18 & 14 \end{pmatrix}$$

La colonna con il maggior numero di elementi consecutivi disposti in ordine crescente è la seconda: 15, 16, 19, 23.

Soluzione

Pseudo-codifica dell’algoritmo

- 1: leggi x, y, z
- 2: se $x < y$ allora
- 3: $min = x, max = y$
- 4: altrimenti
- 5: $min = y, max = x$
- 6: fine-condizione
- 7: per $i = 0, 1, \dots, z - 1$ ripeti:
- 8: per $j = 0, 1, \dots, z - 1$ ripeti:
- 9: $A_{i,j}$ = numero casuale in (min, max) , scrivi $A_{i,j}$
- 10: fine-ciclo
- 11: fine-ciclo
- 12: $max = 0, jmax = 0$

```

13: per  $j = 0, 1, \dots, z - 1$  ripeti:
14:    $c = 0, cmax = 0$ 
15:   per  $i = 0, 1, \dots, z - 2$  ripeti:
16:     se  $A_{i,j} < A_{i+1,j}$  allora
17:        $c = c + 1$ 
18:     altrimenti
19:       se  $c > cmax$  allora
20:          $cmax = c$ 
21:       fine-condizione
22:        $c = 0$ 
23:     fine-condizione
24:   fine-ciclo
25:   se  $cmax > max$  allora
26:      $max = cmax, jmax = j$ 
27:   fine-condizione
28: fine-ciclo
29: per  $i = 0, 1, \dots, z - 1$  ripeti:
30:   scrivi  $A_{i,jmax}$ 
31: fine-ciclo
32: stop

```

Codifica in linguaggio C

```

1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <time.h>
4
5  #define MAX 100
6
7  int minimo(int a, int b) {
8      if (a<b)
9          return(a);
10     else
11         return(b);
12 }
13
14 int massimo(int a, int b) {
15     if (a>b)
16         return(a);
17     else
18         return(b);
19 }
20
21 void generaMatrice(int A[MAX][MAX], int x, int y, int z) {
22     int i, j, min, max;
23     min = minimo(x,y);
24     max = massimo(x,y);
25     srand((unsigned)time(NULL));
26     for (i=0; i<z; i++)
27         for (j=0; j<z; j++)
28             A[i][j] = rand() % (max-min+1) + min;
29     return;
30 }
31

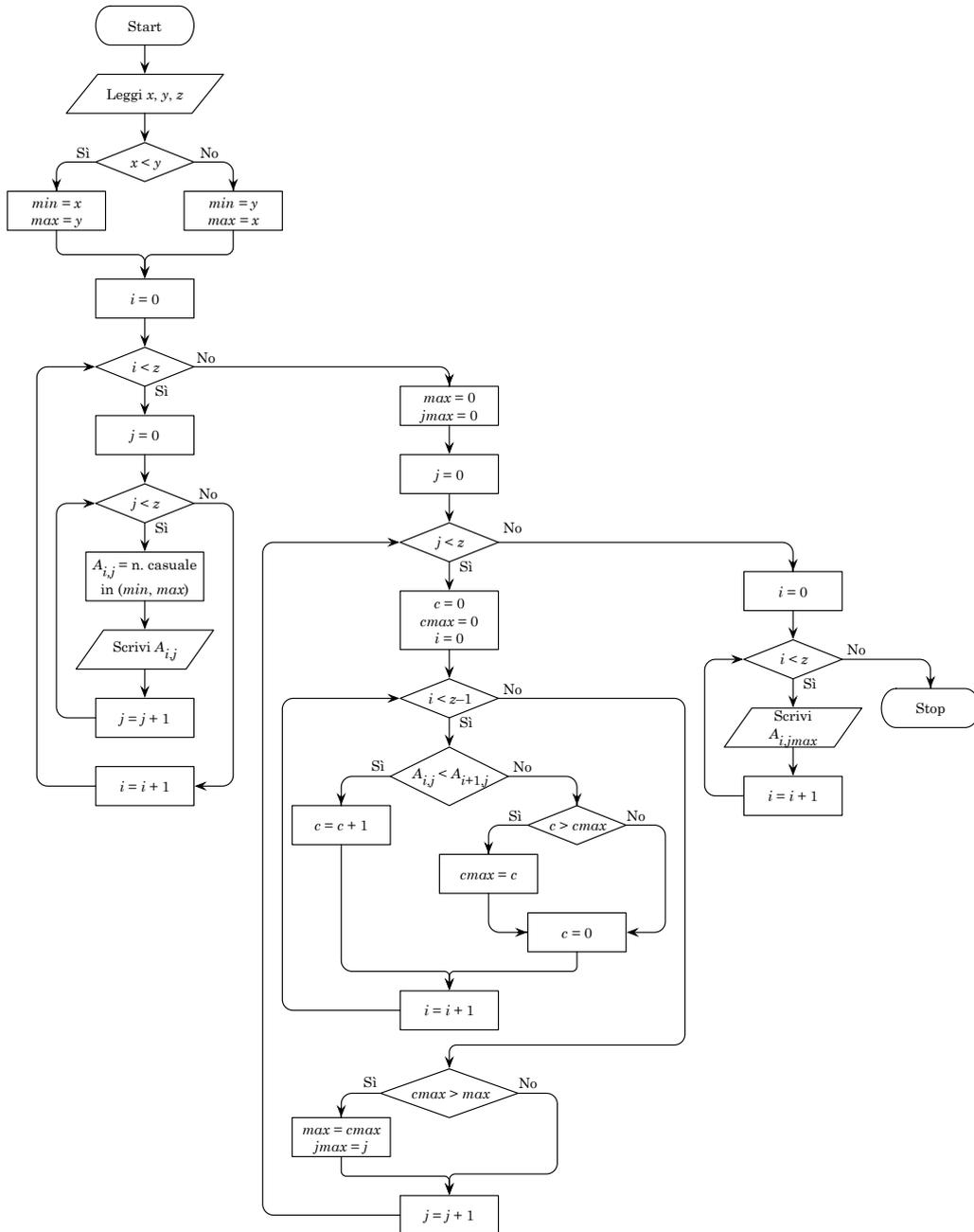
```

```

32 void stampaMatrice(int A[MAX][MAX], int n) {
33     int i, j;
34     for (i=0; i<n; i++) {
35         for (j=0; j<n; j++)
36             printf("%3d ", A[i][j]);
37         printf("\n");
38     }
39     return;
40 }
41
42 void stampaColonna(int A[MAX][MAX], int n) {
43     int i, j, c, cmax, jmax=0, max=0;
44     for (j=0; j<n; j++) {
45         c = 0;
46         cmax = 0;
47         for (i=0; i<n-1; i++) {
48             if (A[i][j]<A[i+1][j]) {
49                 c++;
50             } else {
51                 if (c>cmax)
52                     cmax = c;
53                 c = 0;
54             }
55         }
56         if (cmax > max) {
57             max = cmax;
58             jmax = j;
59         }
60     }
61     for (i=0; i<n; i++)
62         printf("%d ", A[i][jmax]);
63     printf("\n");
64     return;
65 }
66
67 int main(void) {
68     int A[MAX][MAX], x, y, z;
69     printf("Inserisci x, y e z: ");
70     scanf("%d %d %d", &x, &y, &z);
71     generaMatrice(A, x, y, z);
72     stampaMatrice(A, z);
73     stampaColonna(A, z);
74     return(0);
75 }

```

Diagramma di flusso



Esercizio n. 2 (max 8 punti)

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

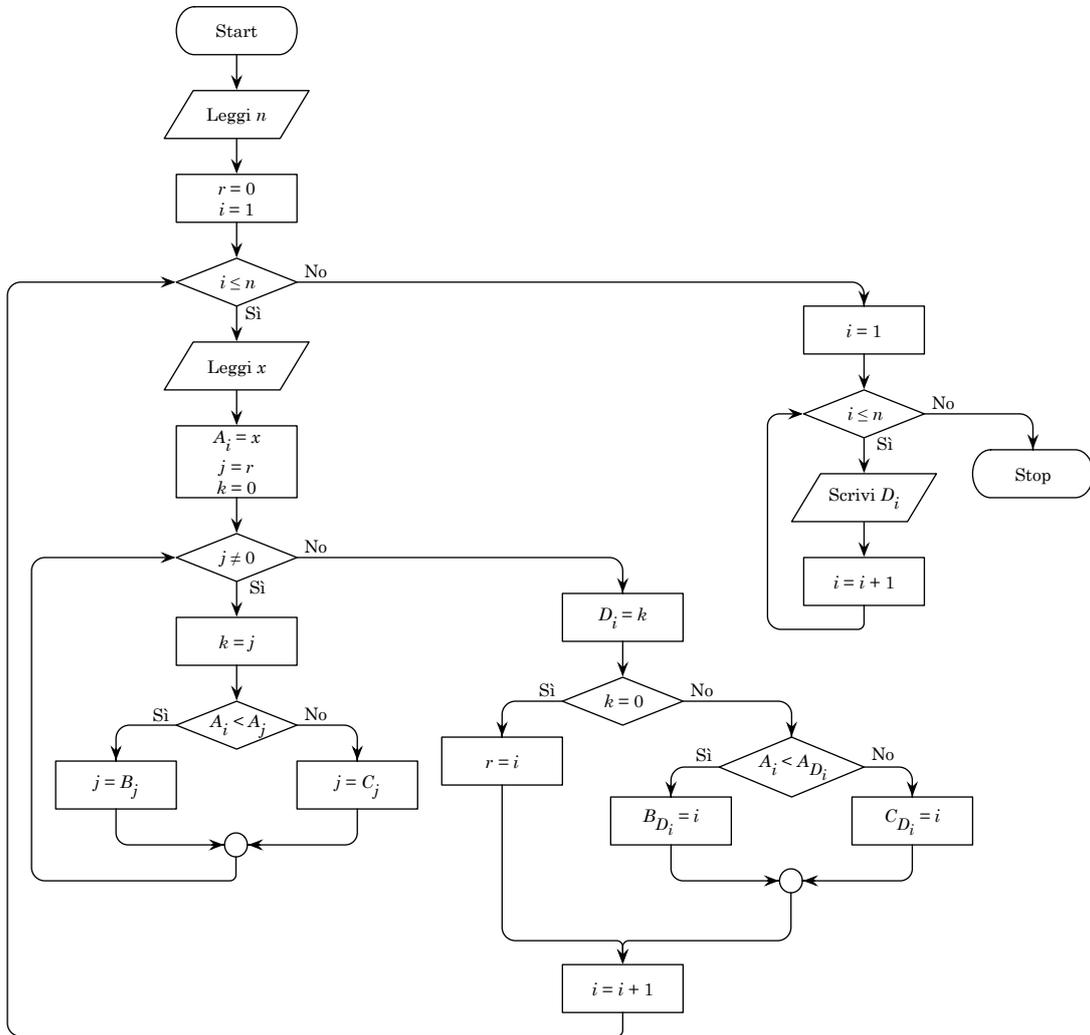
```
1: leggi  $n$ 
2:  $r = 0$ 
3: per  $i = 1, 2, \dots, n$  ripeti:
4:   leggi  $x$ 
5:    $A_i = x$ 
6:    $j = r, k = 0$ 
7:   fintanto che  $j \neq 0$  ripeti:
8:      $k = j$ 
9:     se  $A_i < A_j$  allora
10:       $j = B_j$ 
11:    altrimenti
12:       $j = C_j$ 
13:    fine-condizione
14:  fine-ciclo
15:   $D_i = k$ 
16:  se  $k = 0$  allora
17:     $r = i$ 
18:  altrimenti
19:    se  $A_i < A_{D_i}$  allora
20:       $B_{D_i} = i$ 
21:    altrimenti
22:       $C_{D_i} = i$ 
23:    fine-condizione
24:  fine-condizione
25: fine-ciclo
26: per  $i = 1, 2, \dots, n$  ripeti:
27:   scrivi  $D_i$ 
28: fine-ciclo
```

Soluzione

Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 int main(void) {
6     int n, r=0, x, i, j, k, A[MAX], B[MAX], C[MAX], D[MAX];
7     scanf("%d", &n);
8     for (i=1; i<=n; i++) {
9         scanf("%d", &x);
10        A[i] = x;
11        j = r;
12        k = 0;
13        while (j != 0) {
14            k = j;
15            if (A[i] < A[j])
16                j = B[j];
17            else
18                j = C[j];
19        }
20        D[i] = k;
21        if (k == 0)
22            r = i;
23        else
24            if (A[i] < A[D[i]])
25                B[D[i]] = i;
26            else
27                C[D[i]] = i;
28    }
29    for (i=1; i<=n; i++)
30        printf("%d ", D[i]);
31    return(0);
32 }
```

Diagramma di flusso

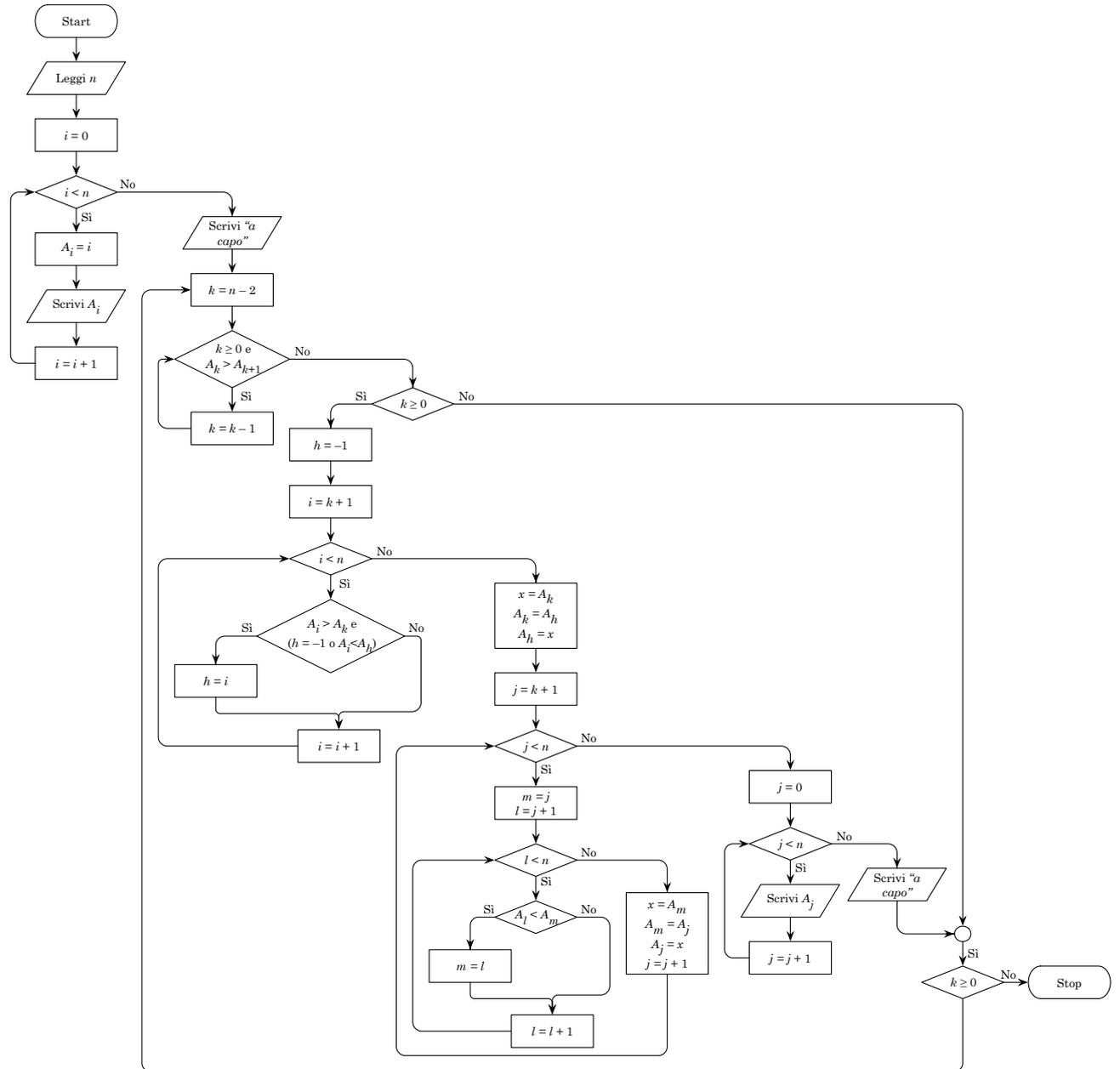


Esercizio n. 3 (max 7 punti)

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

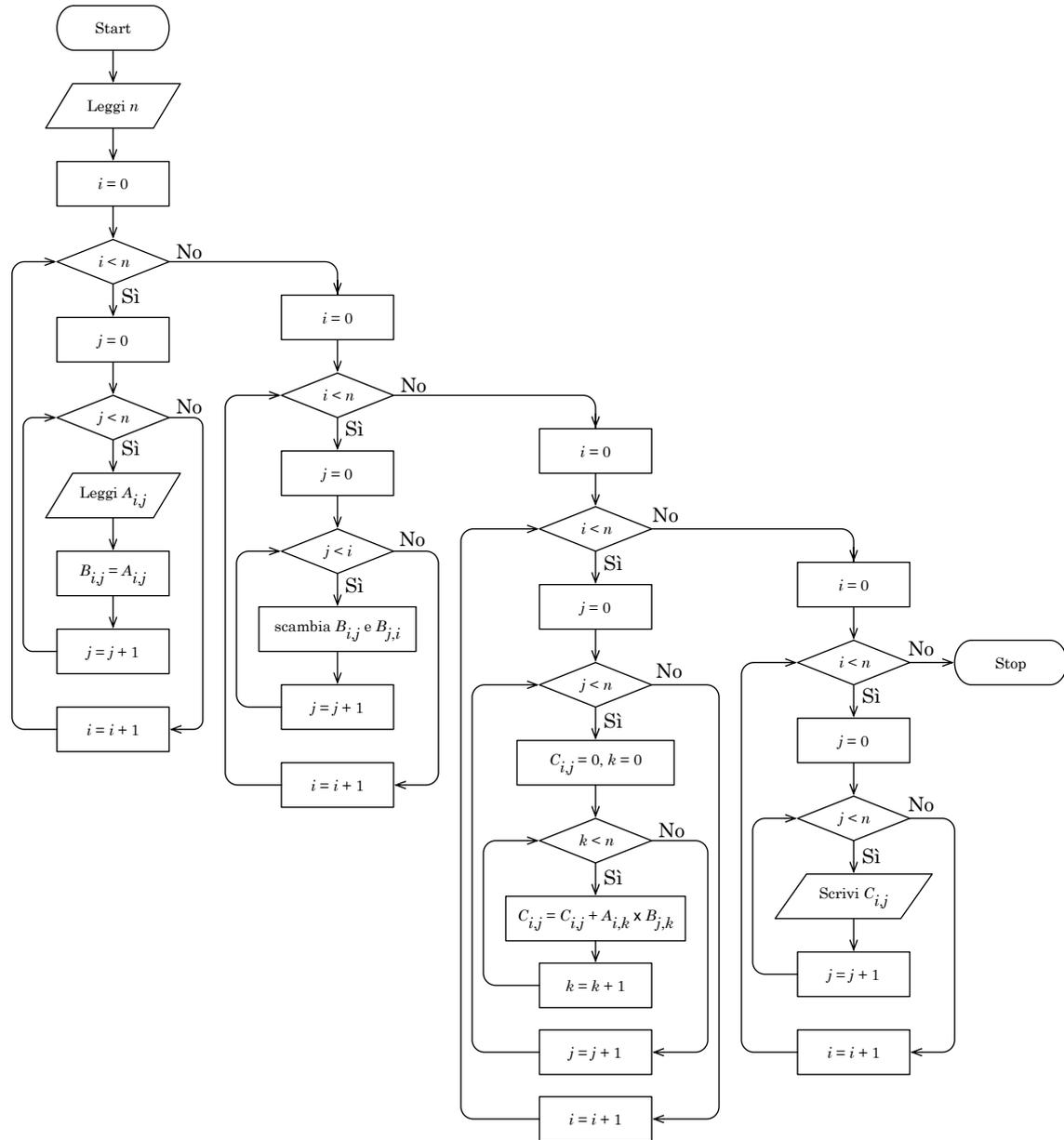
```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4 int main(void) {
5     int i, j, k, h, l, m, n, x, A[MAX];
6     scanf("%d", &n);
7     for (i=0; i<n; i++) {
8         A[i] = i;
9         printf("%3d ", i);
10    }
11    printf("\n");
12    do {
13        k = n-2;
14        while (k>=0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

Soluzione



Esercizio n. 4 (max 7 punti)

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



Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 void scambia(int *x, int *y) {
6     int z;
7     z = *x;
8     *x = *y;
9     *y = z;
10    return;
11 }
12
13 int main(void) {
14     int n, i, j, k, A[MAX][MAX], B[MAX][MAX], C[MAX][MAX];
15     scanf("%d", &n);
16     for (i=0; i<n; i++) {
17         for (j=0; j<n; j++) {
18             scanf("%d", &A[i][j]);
19             B[i][j] = A[i][j];
20         }
21     }
22     for (i=0; i<n; i++)
23         for (j=0; j<i; j++)
24             scambia(&B[i][j], &B[j][i]);
25     for (i=0; i<n; i++) {
26         for (j=0; j<n; j++) {
27             C[i][j] = 0;
28             for (k=0; k<n; k++)
29                 C[i][j] = C[i][j] + A[i][k]*B[j][k];
30         }
31     }
32     for (i=0; i<n; i++) {
33         for (j=0; j<n; j++)
34             printf("%2d ", C[i][j]);
35         printf("\n");
36     }
37     return(0);
38 }
```