

Corso di Informatica 1 (IN110) – Prof. Marco Liverani – a.a. 2019/2020

## Prima prova di esonero – 14 novembre 2019

*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 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.

Letto in input i numeri  $n > 0$  ed  $m > 0$ , generare una matrice  $A$  di ordine  $n \times m$  ( $n$  righe ed  $m$  colonne), contenenti numeri interi compresi tra  $-50$  e  $50$ , estremi inclusi. Visualizzare in output la matrice  $A$ . Costruire un vettore  $B$  di  $m$  elementi tale che per  $k = 0, 1, \dots, m - 1$  risulti  $B_k = 1$  se il numero di elementi positivi sulla colonna  $k$  della matrice  $A$  è maggiore del numero di elementi negativi; altrimenti  $B_k = -1$ . Visualizzare in output il vettore  $B$ .

**Esempio** Sia  $n = 5$ ,  $m = 4$  e si consideri la seguente matrice di numeri casuali:

$$A = \begin{pmatrix} -15 & 29 & 17 & 2 \\ -4 & -50 & 48 & -3 \\ 12 & 3 & 0 & -43 \\ -9 & 28 & -2 & 50 \\ 36 & -24 & 22 & -7 \end{pmatrix}$$

Il vettore  $B$  è il seguente:  $B = (-1, 1, 1, -1)$

## Soluzione

### Pseudo-codifica dell'algoritmo

```
1: leggi  $n$  e  $m$ 
2: per  $i = 0, \dots, n - 1$  ripeti:
3:   per  $j = 0, \dots, m - 1$  ripeti:
4:      $A_{i,j}$  = numero casuale in  $\{-50, \dots, 50\}$ 
5:   fine-ciclo
6: fine-ciclo
7: per  $i = 0, \dots, n - 1$  ripeti:
8:   per  $j = 0, \dots, m - 1$  ripeti:
9:     scrivi  $A_{i,j}$ 
10:  fine-ciclo
11: fine-ciclo
12: per  $j = 0, \dots, m - 1$  ripeti:
13:    $pos = 0, neg = 0$ 
14:   per  $i = 0, \dots, n - 1$  ripeti:
15:     se  $A_{i,j} < 0$  allora
16:        $neg = neg + 1$ 
17:     fine-condizione
18:     se  $A_{i,j} > 0$  allora
19:        $pos = pos + 1$ 
20:     fine-condizione
21:   fine-ciclo
22:   se  $pos > neg$  allora
23:      $B_j = 1$ 
24:   altrimenti
25:      $B_j = -1$ 
26:   fine-condizione
27: fine-ciclo
28: per  $i = 0, \dots, m - 1$  ripeti:
29:   scrivi  $B_i$ 
30: fine-ciclo
31: stop
```

### Codifica in linguaggio C

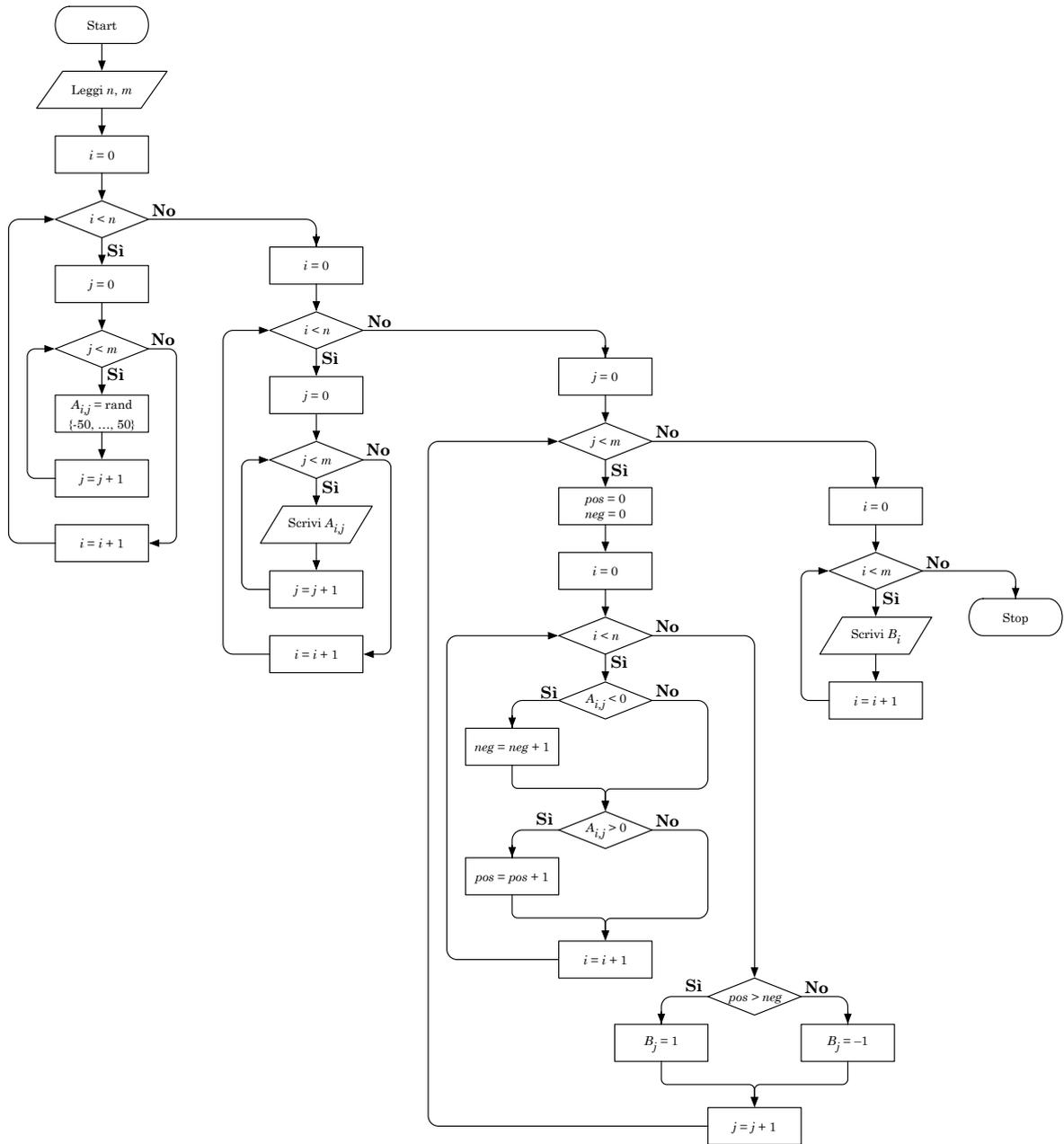
```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4
5 #define MAX 100
6
7 void generaMatrice(int A[MAX][MAX], int *n, int *m) {
8     int i, j;
9     printf("Numero di righe e di colonne: ");
10    scanf("%d %d", n, m);
11    srand((unsigned)time(NULL));
12    for (i=0; i<*n; i++)
13        for (j=0; j<*m; j++)
14            A[i][j] = rand() % (101) - 50;
15    return;
16 }
17
```

```

18 void stampaMatrice(int A[MAX][MAX], int n, int m) {
19     int i, j;
20     for (i=0; i<n; i++) {
21         for (j=0; j<m; j++)
22             printf("%3d ", A[i][j]);
23         printf("\n");
24     }
25     return;
26 }
27
28 void stampaVettore(int B[], int n) {
29     int i;
30     for (i=0; i<n; i++)
31         printf("%d ", B[i]);
32     return;
33 }
34
35 int main(void) {
36     int A[MAX][MAX], B[MAX], n, m, i, j, pos, neg;
37     generaMatrice(A, &n, &m);
38     stampaMatrice(A, n, m);
39     for (j=0; j<m; j++) {
40         pos = 0;
41         neg = 0;
42         for (i=0; i<n; i++) {
43             if (A[i][j] < 0)
44                 neg++;
45             if (A[i][j] > 0)
46                 pos++;
47         }
48         if (pos > neg)
49             B[j] = 1;
50         else
51             B[j] = -1;
52     }
53     stampaVettore(B, m);
54     return(0);
55 }

```

## Diagramma di flusso



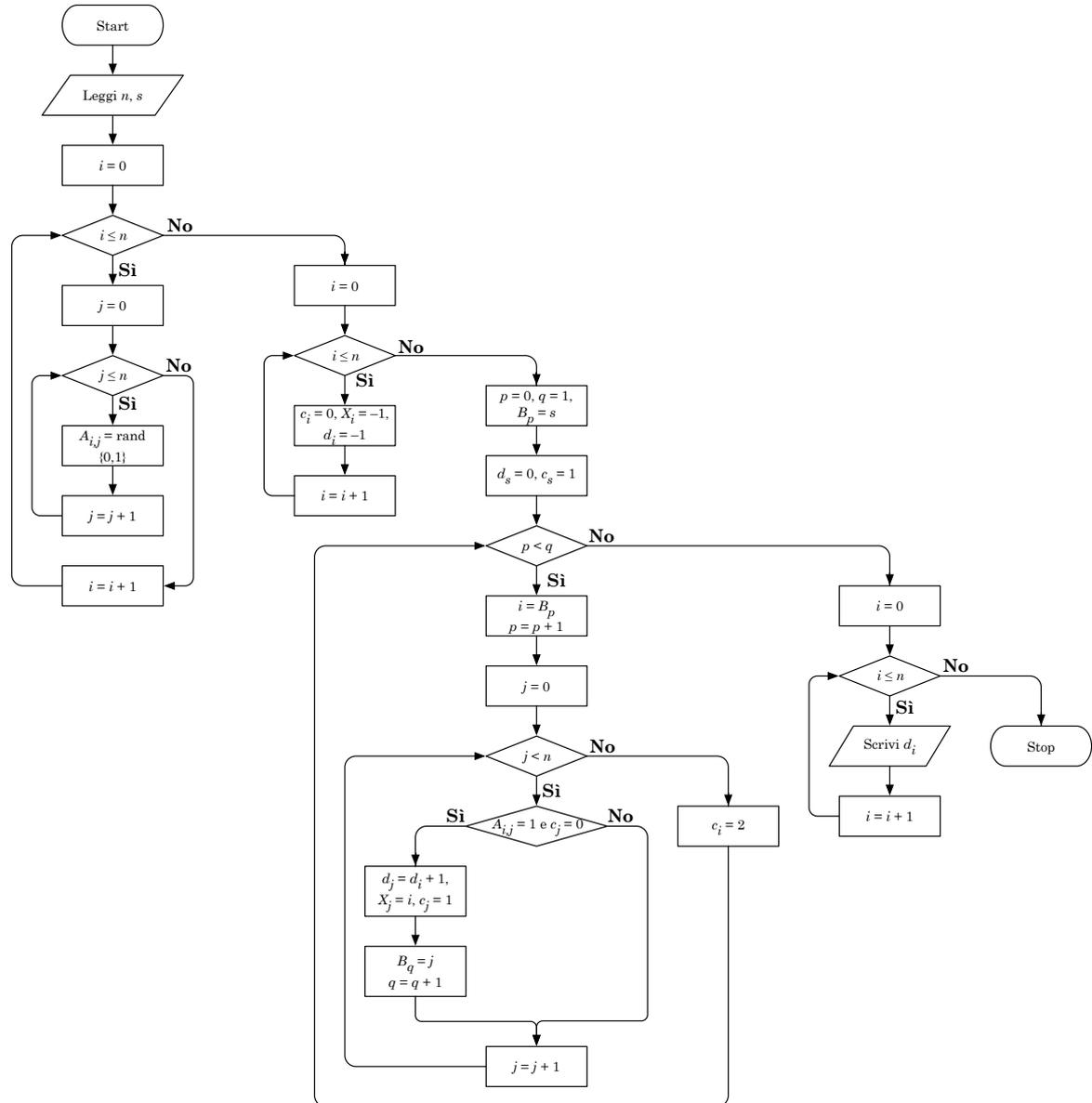
## Esercizio n. 2 (max 8 punti)

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

```
1: leggi  $n$  e  $s$ 
2: per  $i = 0, 1, \dots, n$  ripeti:
3:   per  $j = 0, 1, \dots, n$  ripeti:
4:      $A_{i,j} = \text{rand}(\{0, 1\})$ 
5:   fine-ciclo
6: fine-ciclo
7: per  $i = 0, 1, \dots, n$  ripeti:
8:    $c_i = 0, X_i = -1, d_i = -1$ 
9: fine-ciclo
10:  $p = 0, q = 1, B_p = s, d_s = 0, c_s = 1$ 
11: fintanto che  $p < q$  ripeti:
12:    $i = B_p, p = p + 1$ 
13:   per  $j = 0, \dots, n - 1$  ripeti:
14:     se  $A_{i,j} = 1$  e  $c_j = 0$  allora
15:        $d_j = d_i + 1, X_j = i, c_j = 1$ 
16:        $B_q = j, q = q + 1$ 
17:     fine-condizione
18:   fine-ciclo
19:    $c_i = 2$ 
20: fine-ciclo
21: per  $i = 0, 1, \dots, n$  ripeti:
22:   scrivi  $d_i$ 
23: fine-ciclo
```

## Soluzione

### Diagramma di flusso



## Codifica in linguaggio C

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

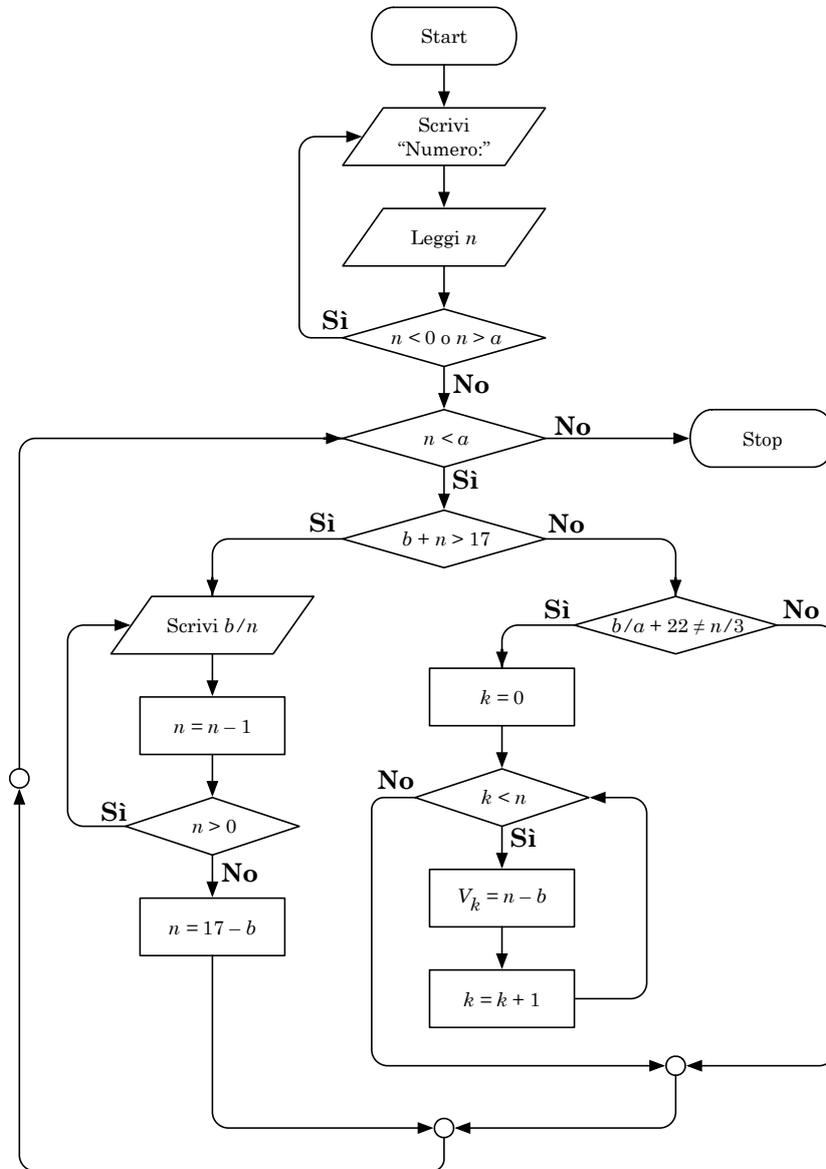
### Esercizio n. 3 (max 6 punti)

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

*Nota: la funzione riporta istruzioni sintatticamente corrette, ma complessivamente prive di significato.*

```
1 void funzione(int a, float b) {
2     int k, n;
3     float V[MAX];
4     do {
5         printf("Inserisci un numero: ");
6         scanf("%d", &n);
7     } while (n<0 || n>a);
8     while (n < a) {
9         if (b+n > 17) {
10            do {
11                printf("%f\n", b/n);
12                n--;
13            } while (n>0);
14            n = 17-b;
15        } else {
16            if (b/a+22 != n/3) {
17                for (k=0; k<n; k++) {
18                    V[k] = n-b;
19                }
20            }
21        }
22    }
23    return;
24 }
```

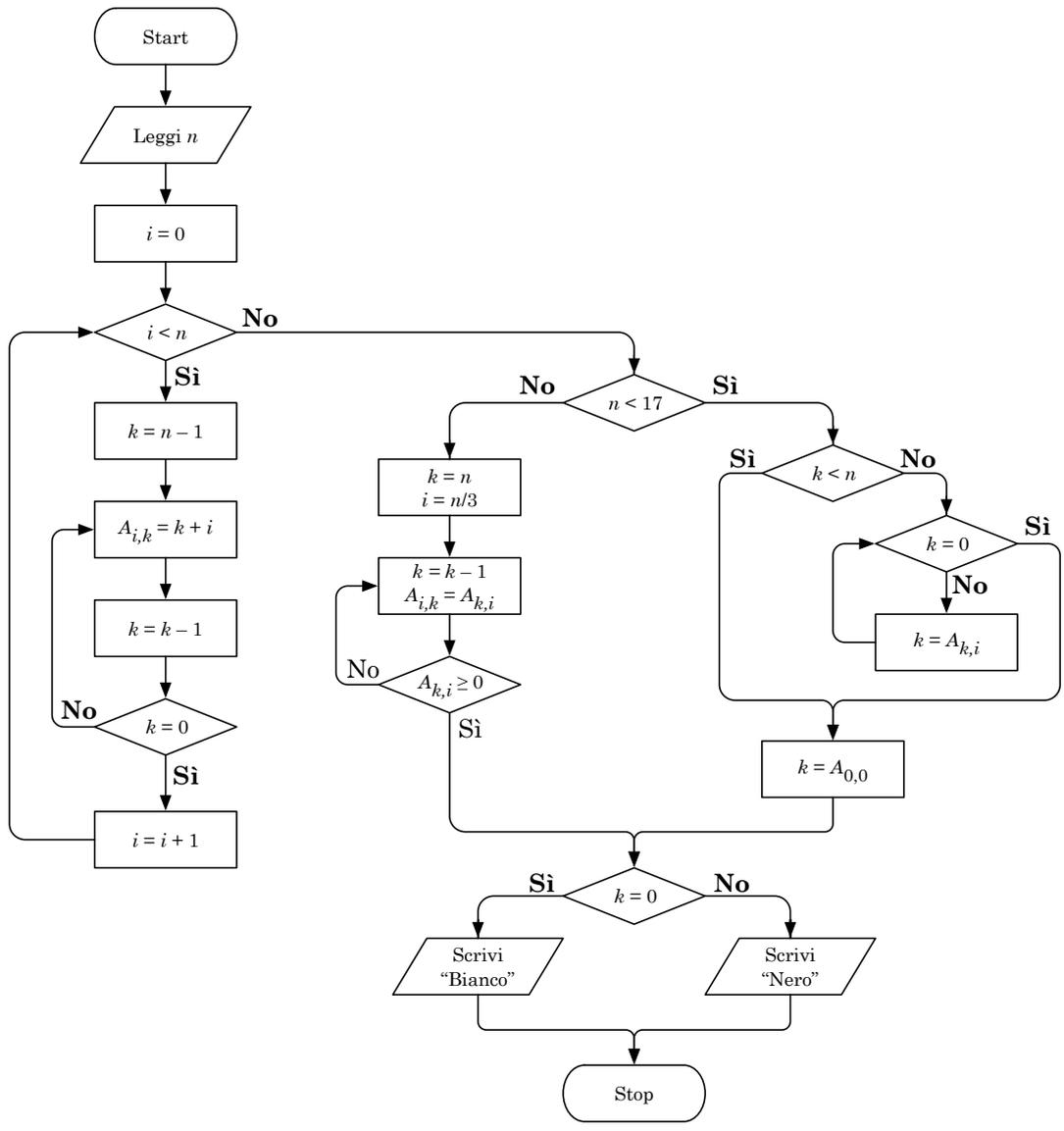
# Soluzione



### Esercizio n. 4 (max 6 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.

Nota: il diagramma di flusso è formalmente corretto, anche se rappresenta un algoritmo privo di significato.



## Soluzione

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