

Corso di Informatica 1 (IN110) – Prof. Marco Liverani – a.a. 2014/2015

Prima prova di esonero – 19 novembre 2014

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.

Leggere in input una sequenza di n numeri interi e memorizzarla in un vettore $A = (a_0, \dots, a_{n-1})$. Costruire un array $B = (b_0, b_1, \dots, b_{n-1})$ tale che, per ogni $k = 0, \dots, n-1$, b_k sia la media aritmetica degli elementi $a_k, a_{k+1}, \dots, a_{n-1}$. Stampare l'array B .

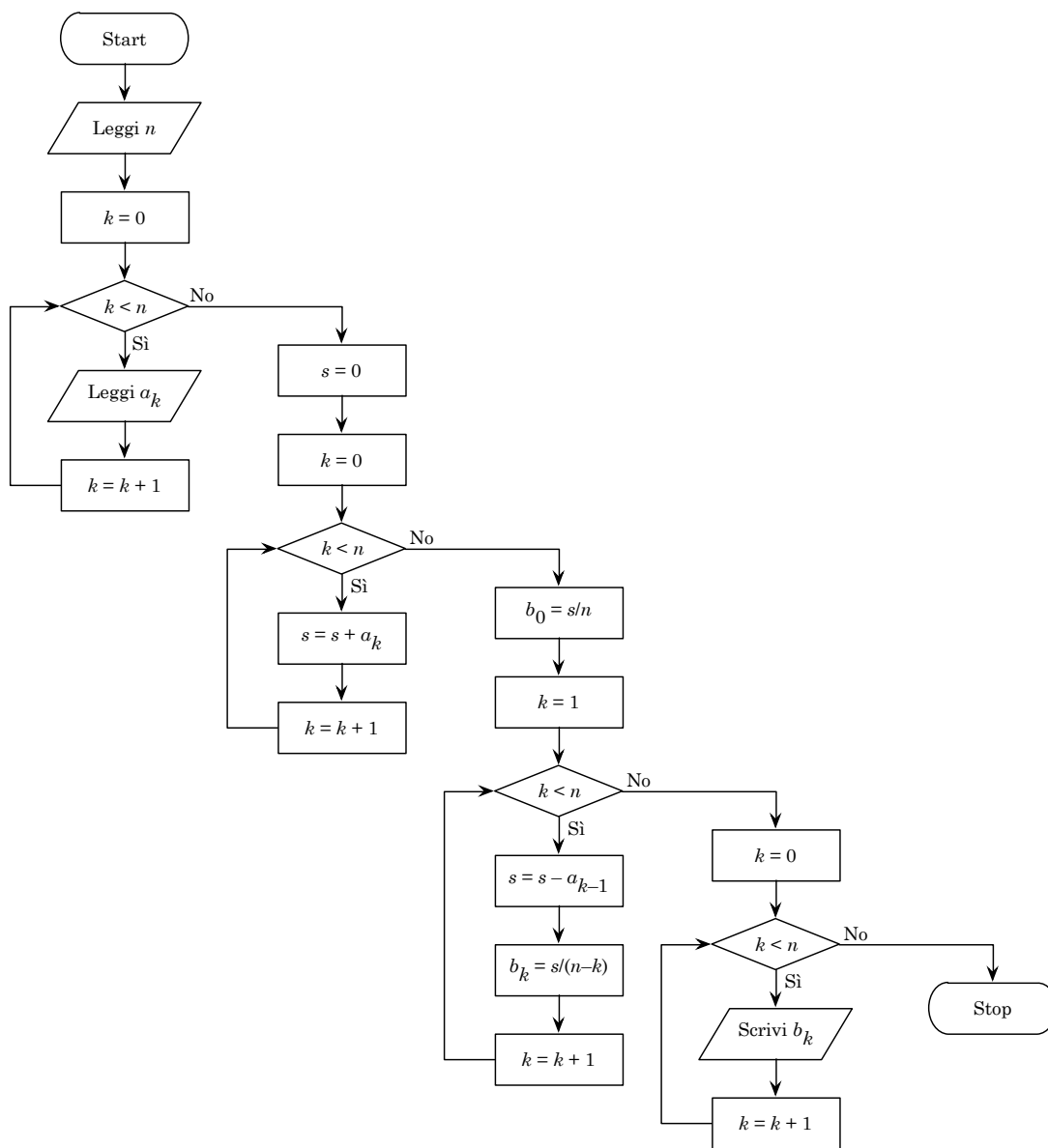
Esempio Supponiamo che $n = 8$ e $A = (12, 43, 11, 5, 2, 7, 25, 37)$; Quindi il vettore B sarà costruito come segue: $B = (17.75, 18.57, 14.5, 15.2, 17.75, 23.0, 31.0, 37.0)$.

Soluzione

Pseudo-codifica dell'algoritmo

- 1: leggi n
- 2: per $k = 0, 1, \dots, n-1$ ripeti:
- 3: leggi a_k
- 4: fine-ciclo
- 5: $s = 0$
- 6: per $k = 0, 1, \dots, n-1$ ripeti:
- 7: $s = s + a_k$
- 8: fine-ciclo
- 9: $b_0 = s/n$
- 10: per $k = 1, 2, \dots, n-1$ ripeti:
- 11: $s = s - a_{k-1}$
- 12: $b_k = s/(n-k)$
- 13: fine-ciclo
- 14: per $k = 0, 1, \dots, n-1$ ripeti:
- 15: scrivi b_k
- 16: fine-ciclo

Diagramma di flusso



Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 int leggiArray(int A[]) {
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", &A[i]);
12    }
13    return(n);
14 }
15
16 void stampaArray(float A[], int n) {
17     int i;
18     for (i=0; i<n; i++) {
19         printf("%.2f\n", A[i]);
20     }
21     return;
22 }
23
24 void costruisciArray(int A[], float B[], int n) {
25     int k, s=0;
26     for (k=0; k<n; k++) {
27         s = s + A[k];
28     }
29     B[0] = (float)s/n;
30     for (k=1; k<n; k++) {
31         s = s - A[k-1];
32         B[k] = (float)s/(n-k);
33     }
34     return;
35 }
36
37 int main(void) {
38     int A[MAX], n;
39     float B[MAX];
40     n = leggiArray(A);
41     costruisciArray(A, B, n);
42     stampaArray(B, n);
43     return(0);
44 }
```

Esercizio n. 2 (max 8 punti)

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

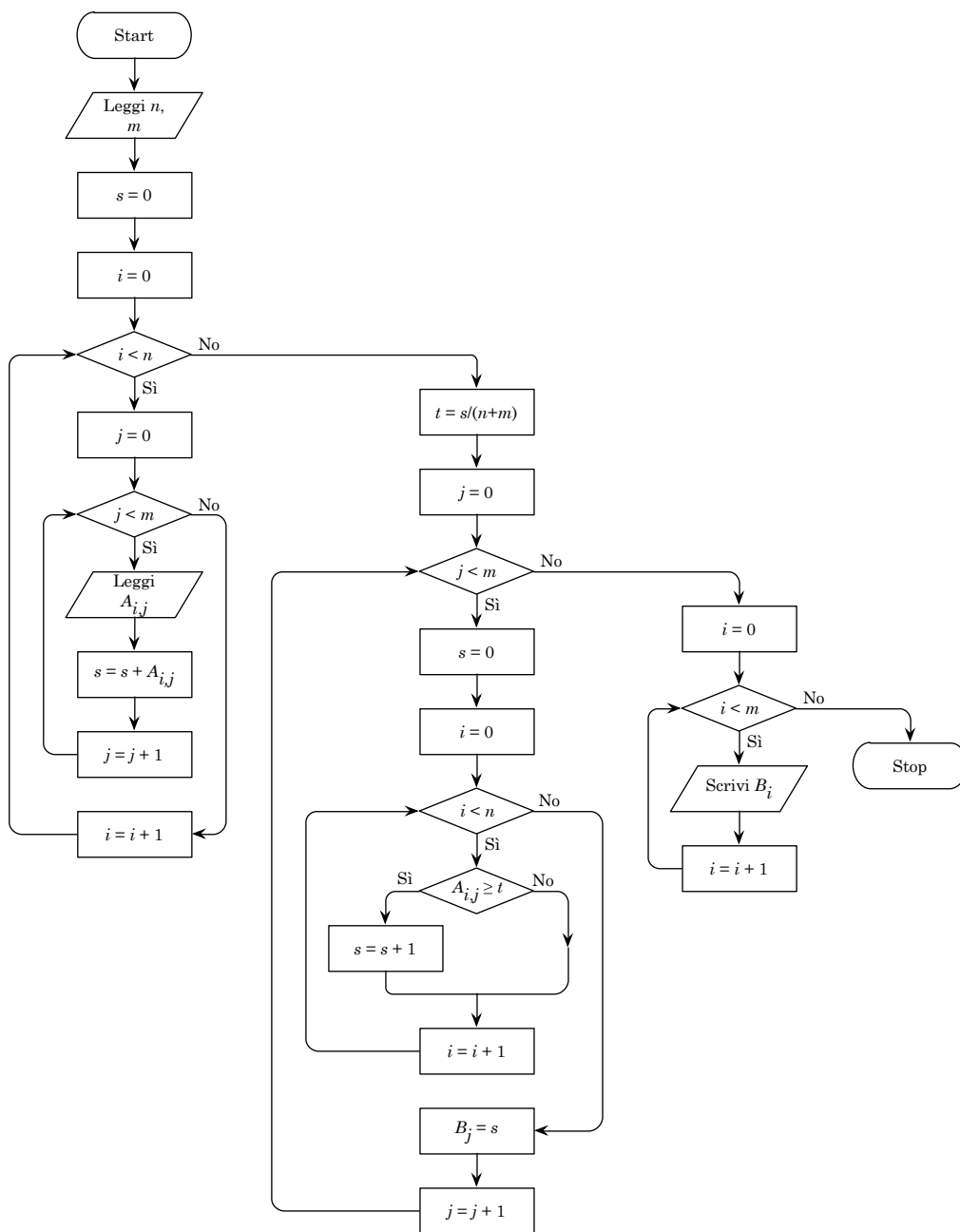
```
1: leggi  $n, m$ 
2:  $s = 0$ 
3: per  $i = 0, 1, \dots, n - 1$  ripeti:
4:   per  $j = 0, 1, \dots, m - 1$  ripeti:
5:     leggi  $A_{i,j}$ 
6:      $s = s + A_{i,j}$ 
7:   fine-ciclo
8: fine-ciclo
9:  $t = \frac{s}{n+m}$ 
10: per  $j = 0, 1, \dots, m - 1$  ripeti:
11:    $s = 0, i = 0$ 
12:   fintanto che  $i < n$  ripeti:
13:     se  $A_{i,j} \geq t$  allora
14:        $s = s + 1$ 
15:     fine-condizione
16:      $i = i + 1$ 
17:   fine-ciclo
18:    $B_j = s$ 
19: fine-ciclo
20: per  $i = 0, 1, 2, \dots, m - 1$  ripeti:
21:   scrivi  $B_i$ 
22: 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 A[MAX][MAX], B[MAX], i, j, n, m, s=0;
7     float t;
8     scanf("%d %d", &n, &m);
9     for (i=0; i<n; i++) {
10        for (j=0; j<m; j++) {
11            scanf("%d", &A[i][j]);
12            s = s+A[i][j];
13        }
14    }
15    t = (float)s/(n+m);
16    for (j=0; j<m; j++) {
17        s = 0;
18        for (i=0; i<n; i++) {
19            if (A[i][j] >= t) {
20                s = s+1;
21            }
22        }
23        B[j] = s;
24    }
25    for (i=0; i<m; i++) {
26        printf("%d\n", B[i]);
27    }
28    return(0);
29 }
```

Diagramma di flusso

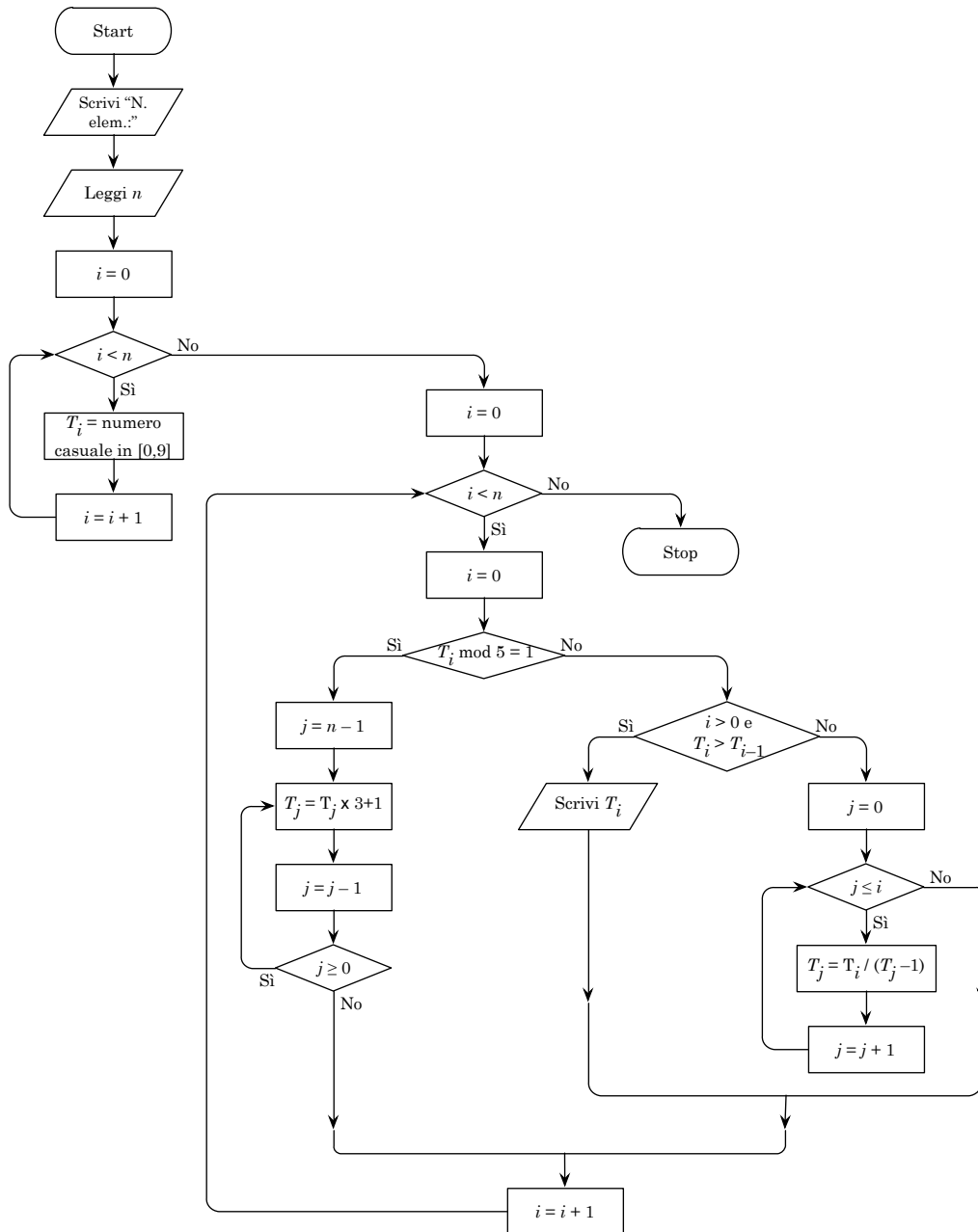


Esercizio n. 3 (max 6 punti)

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

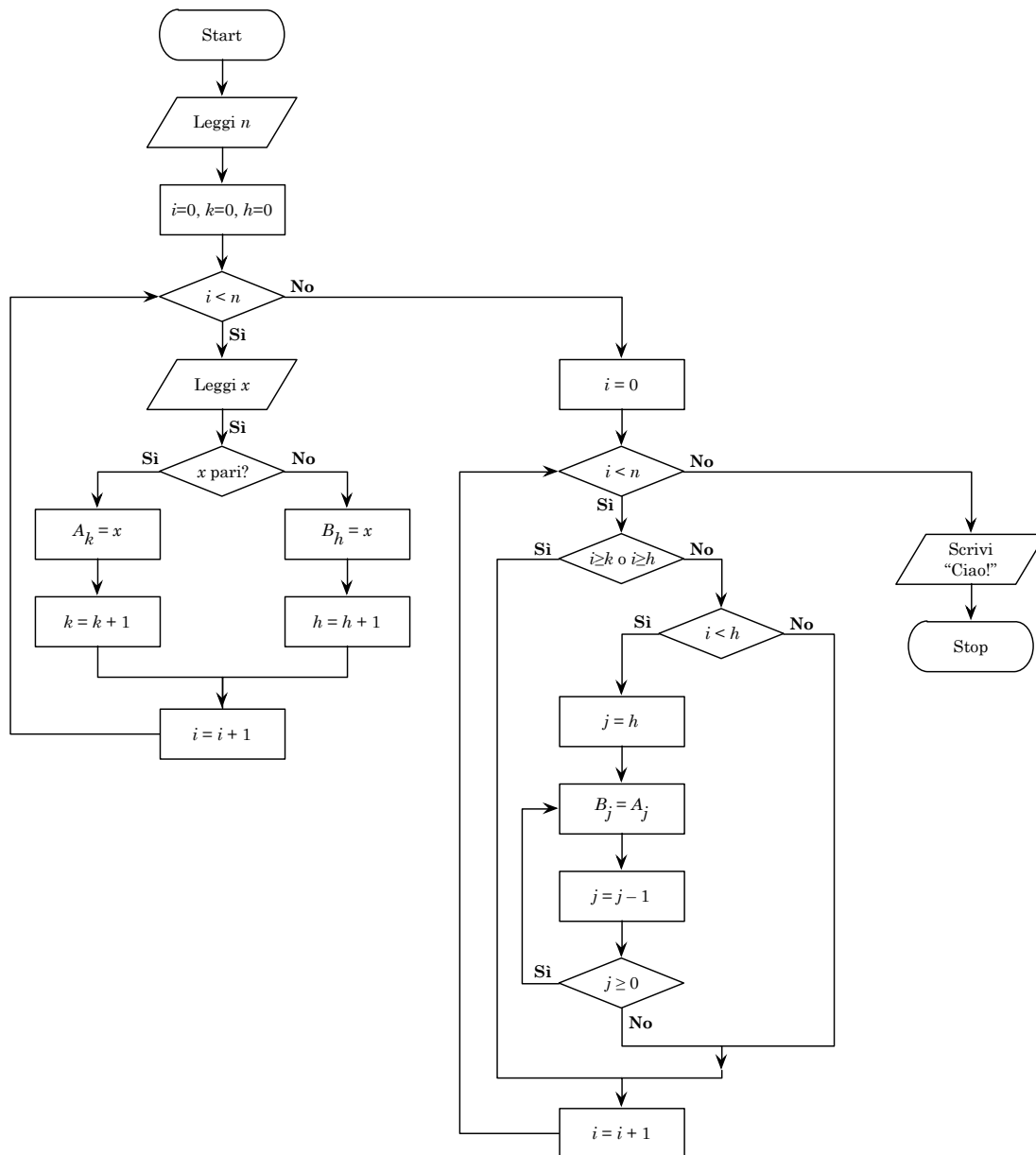
```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 100
5
6 int main(void) {
7     int T[MAX], i, j, n;
8     printf("Numero di elementi: ");
9     scanf("%d", &n);
10    for (i=0; i<n; i++)
11        T[i] = rand() % 10;
12    i = 0;
13    while (i < n) {
14        i = 0;
15        if (T[i] % 5 == 1) {
16            j = n-1;
17            do {
18                T[j] = T[j]*3+1;
19                j--;
20            } while (j>=0);
21        } else {
22            if (i>0 && T[i] > T[i-1])
23                printf("%d\n", T[i]);
24            else {
25                for (j=0; j<=i; j++)
26                    T[j] = T[i]/(T[j]-1);
27            }
28        }
29        i = i+1;
30    }
31    return(0);
32 }
```

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.



Soluzione

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