

## Prima prova di esonero – 10 novembre 2023

*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.

Letti in input due interi  $n > 0$  e  $k > 0$ , costruire un array  $A$  di  $n$  numeri interi casuali compresi in  $\{1, 2, \dots, k-2, k-1, k\}$ . Calcolare l'indice  $m$  tale che la somma dei primi  $m+1$  elementi di  $A$  sia più vicina alla somma dei restanti elementi di  $A$ . In altri termini si chiede di calcolare l'indice  $m$  tale che  $|a_0 + a_1 + \dots + a_m - a_{m+1} - \dots - a_{n-1}|$  sia minimo. Stampare in output l'indice  $m$ , la parte di array  $a_0, \dots, a_m$  e la parte di array  $a_{m+1}, \dots, a_{n-1}$ . Ricordiamo che in linguaggio C la funzione `abs(...)` restituisce il valore assoluto dell'argomento.

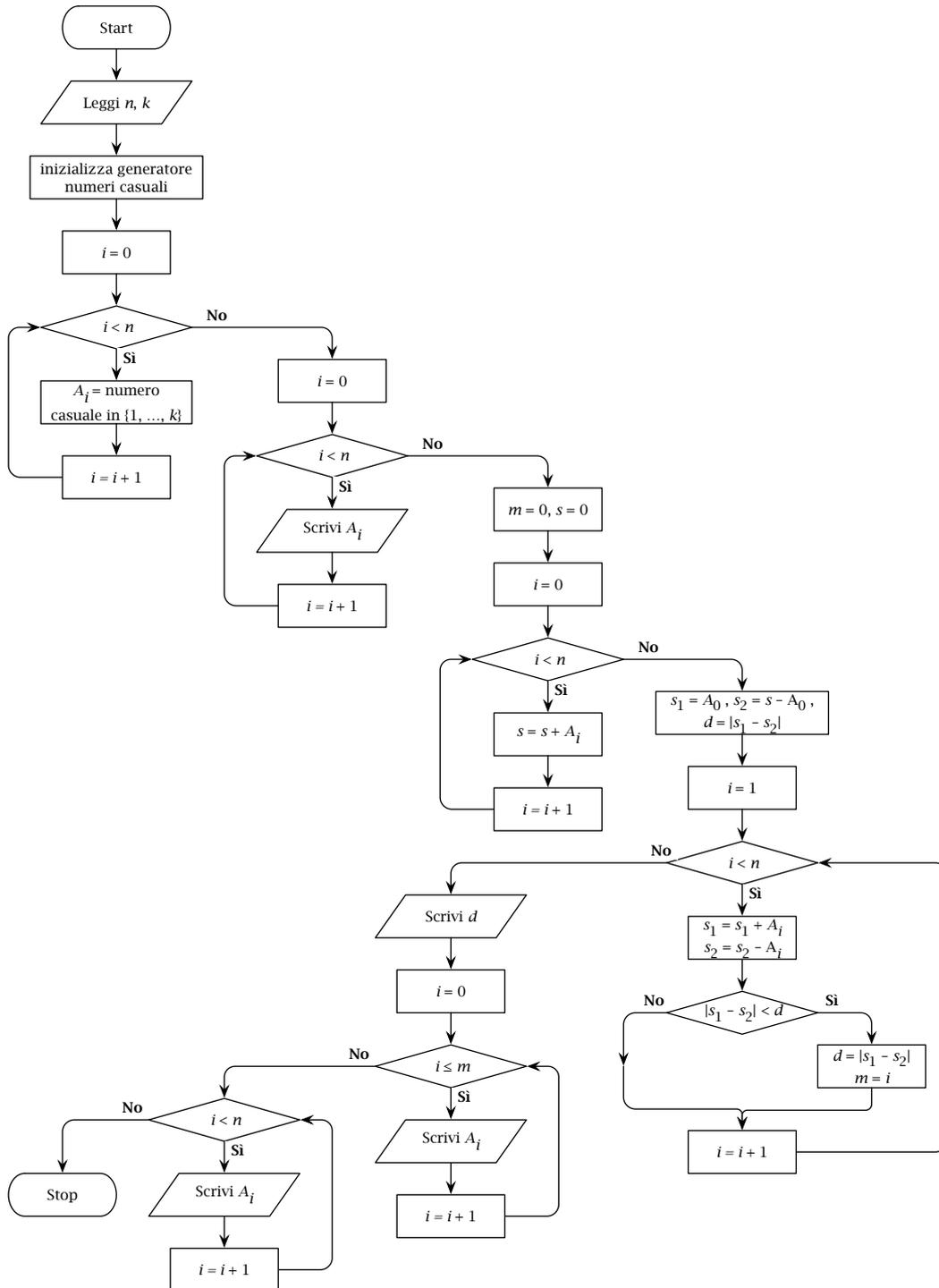
**Esempio** Siano  $n = 10$  e  $k = 5$ . Sia inoltre  $A = (5, 2, 1, 2, 4, 2, 1, 4, 5, 5)$ . Allora risulta  $m = 5$  e l'array  $A$  viene suddiviso nelle due parti  $A' = (5, 2, 1, 2, 4, 2)$  e  $A'' = (1, 4, 5, 5)$ , rispettivamente con somma 16 e 15 e differenza  $16 - 15 = 1$ .

## Soluzione

### Pseudo-codifica dell'algoritmo

- 1: leggi  $n$  e  $k$
- 2: inizializza il generatore dei numeri casuali
- 3: per  $i = 0, 1, \dots, n - 1$  ripeti:
  - 4:  $A_i =$  numero casuale in  $\{1, \dots, k\}$
  - 5: fine-ciclo
- 6: per  $i = 0, 1, \dots, n - 1$  ripeti:
  - 7: scrivi  $A_i$
  - 8: fine-ciclo
- 9:  $m = 0, s = 0$
- 10: per  $i = 0, 1, \dots, n - 1$  ripeti:
  - 11:  $s = s + A_i$
  - 12: fine-ciclo
  - 13:  $s_1 = A_0, s_2 = s - A_0, d = |s_1 - s_2|$
  - 14: per  $i = 1, 2, \dots, n - 1$  ripeti:
    - 15:  $s_1 = s_1 + A_i, s_2 = s_2 - A_i$
    - 16: se  $|s_1 - s_2| < d$  allora
    - 17:  $d = |s_1 - s_2|, m = i$
    - 18: fine-condizione
  - 19: fine-ciclo
- 20: scrivi  $d$
- 21: scrivi "Prima parte:"
- 22: per  $i = 0, \dots, m$  ripeti:
  - 23: scrivi  $A_i$
  - 24: fine-ciclo
- 25: scrivi "Seconda parte:"
- 26: per  $i = m + 1, \dots, n - 1$  ripeti:
  - 27: scrivi  $A_i$
  - 28: fine-ciclo

### Diagramma di flusso



## Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 1000
5
6 int arrayCasuale(int x[]) {
7     int n, i, k;
8     srand((unsigned)time(NULL));
9     printf("Numero di elementi: ");
10    scanf("%d", &n);
11    printf("Valore massimo: ");
12    scanf("%d", &k);
13    for (i=0; i<n; i++) {
14        x[i] = rand() % k + 1;
15    }
16    return(n);
17 }
18
19 void scriviArray(int x[], int n) {
20     int i;
21     for (i=0; i<n; i++)
22         printf("%d ", x[i]);
23     printf("\n");
24     return;
25 }
26
27 int dividiArray(int x[], int n) {
28     int i, s=0, s1, s2, m = 0, d;
29     for (i=0; i<n; i++) {
30         s = s + x[i];
31     }
32     s1 = x[0];
33     s2 = s - x[0];
34     d = abs(s1-s2);
35     for (i=1; i<n; i++) {
36         s1 = s1 + x[i];
37         s2 = s2 - x[i];
38         if (abs(s1-s2) < d) {
39             d = abs(s1-s2);
40             m = i;
41         }
42     }
43     return(m);
44 }
45
```

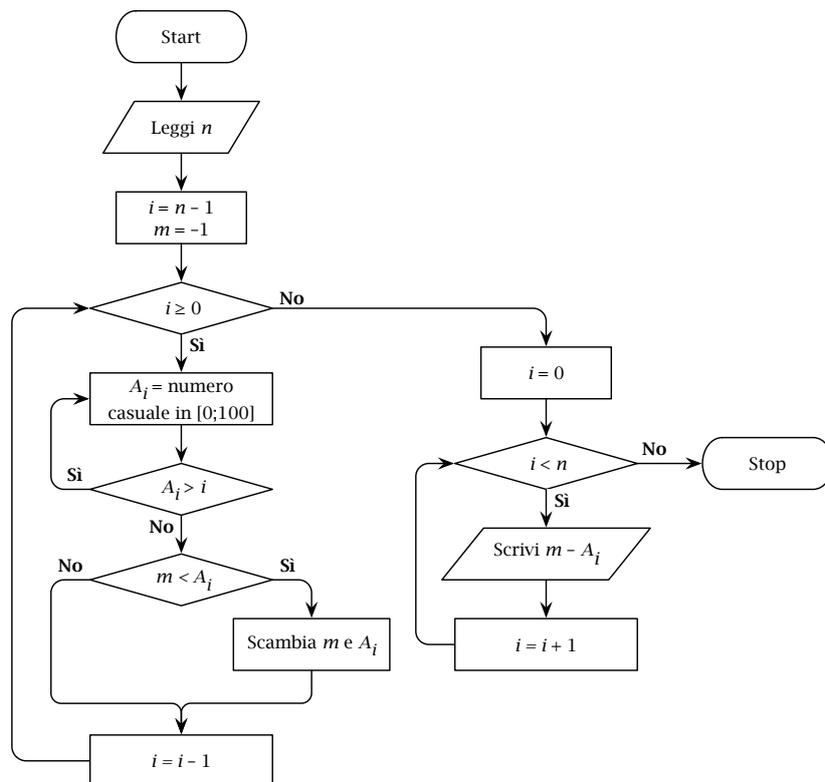
```
46 int main(void) {  
47     int a[MAX], n, m;  
48     n = arrayCasuale(a);  
49     scriviArray(a, n);  
50     m = dividiArray(a, n);  
51     scriviArray(a, m+1);  
52     scriviArray(&a[m+1], n-m-1);  
53     return(0);  
54 }
```

## 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:  $i = n - 1$
- 3:  $m = -1$
- 4: fintanto che  $i \geq 0$  ripeti:
- 5:   ripeti:
- 6:      $a_i =$  numero casuale in  $[0, 100]$
- 7:     fintanto che  $a_i > i$
- 8:     se  $m < a_i$  allora
- 9:       scambia  $m$  con  $a_i$
- 10:    fine-condizione
- 11:     $i = i - 1$
- 12: fine-ciclo
- 13: per  $i = 0, \dots, n - 1$  ripeti:
- 14:   scrivi  $m - a_i$
- 15: fine-ciclo

### 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 A[MAX], i, n, m = -1, x;
8     srand((unsigned)time(NULL));
9     scanf("%d", &n);
10    i = n-1;
11    while (i >= 0) {
12        do {
13            A[i] = rand() % 101;
14        } while (A[i] > i);
15        if (m < A[i]) {
16            x = m;
17            m = A[i];
18            A[i] = x;
19        }
20        i = i-1;
21    }
22    for (i=0; i<n; i++) {
23        printf("%d ", m - A[i]);
24    }
25    return(0);
26 }
```

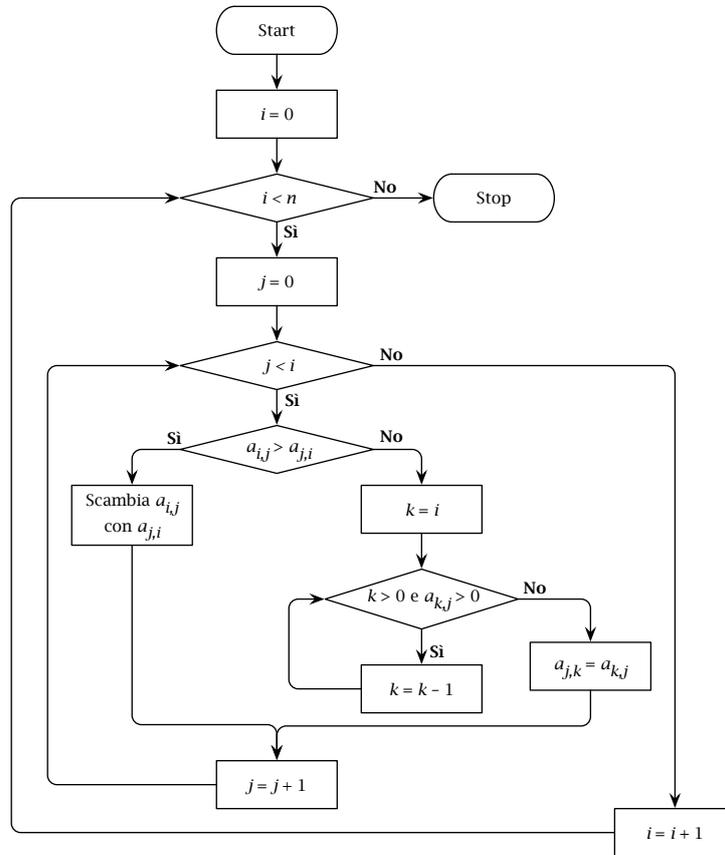
### Esercizio n. 3 (max 6 punti)

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(float a[100][100], int n) {
2     int i, j, k;
3     for (i=0; i<n; i++) {
4         for (j=0; j<i; j++) {
5             if (a[i][j] > a[j][i])
6                 scambia(&a[i][j], &a[j][i]);
7             else {
8                 k = i;
9                 while (k>0 && a[k][j] > 0) {
10                    k--;
11                }
12                a[j][k] = a[k][j];
13            }
14        }
15    }
16    return;
17 }
```

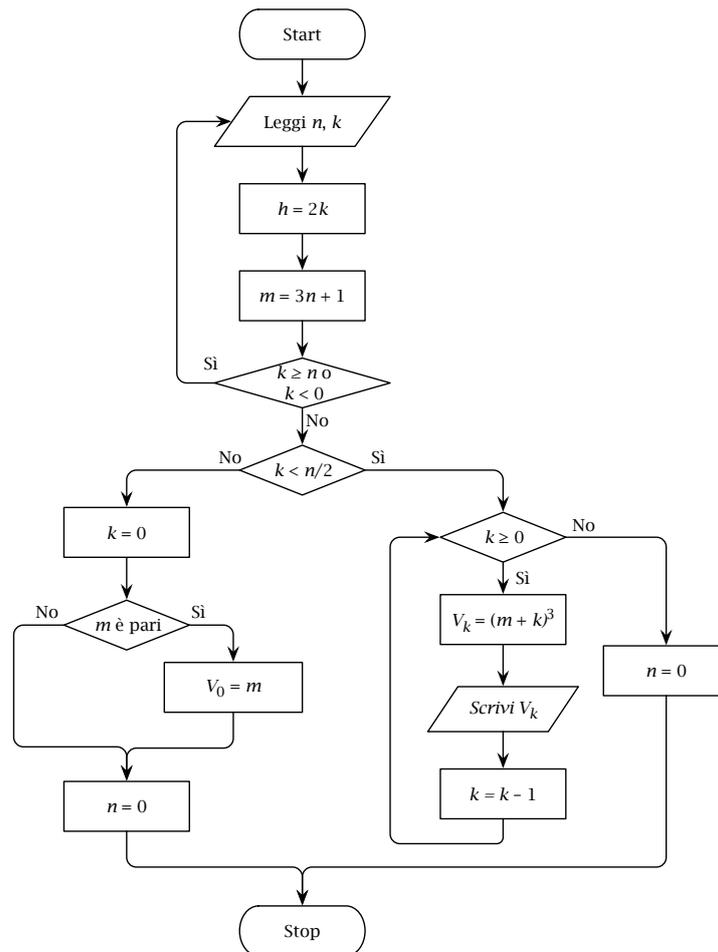
## Soluzione



### 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, h, k, V[MAX];
7     do {
8         scanf("%d %d", &n, &k);
9         h = 2*k;
10        m = 3*n + 1;
11    } while (k >= n || k < 0);
12    if (k < n/2) {
13        while (k > 0) {
14            V[k] = (m+k)*(m+k)*(m+k);
15            printf("%d ", V[k]);
16            k = k-1;
17        }
18        n = 0;
19    } else {
20        k = 0;
21        if (m % 2 == 0)
22            V[0] = m;
23        n = 0;
24    }
25    return(0);
26 }
```