

Corso di Algoritmi e Strutture Dati (IN110) – Prof. Marco Liverani – a.a. 2025/2026

Prima prova di esonero – 12 novembre 2025

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 un numero $n > 0$ costruire un array A di n numeri casuali scelti nell'insieme $\{3, 4, 5, 6, 7\}$. Costruire un secondo array B con gli elementi di $A = \{a_0, a_1, \dots, a_{n-1}\}$ ripetuti a_i volte. Stampare in output gli array A e B .

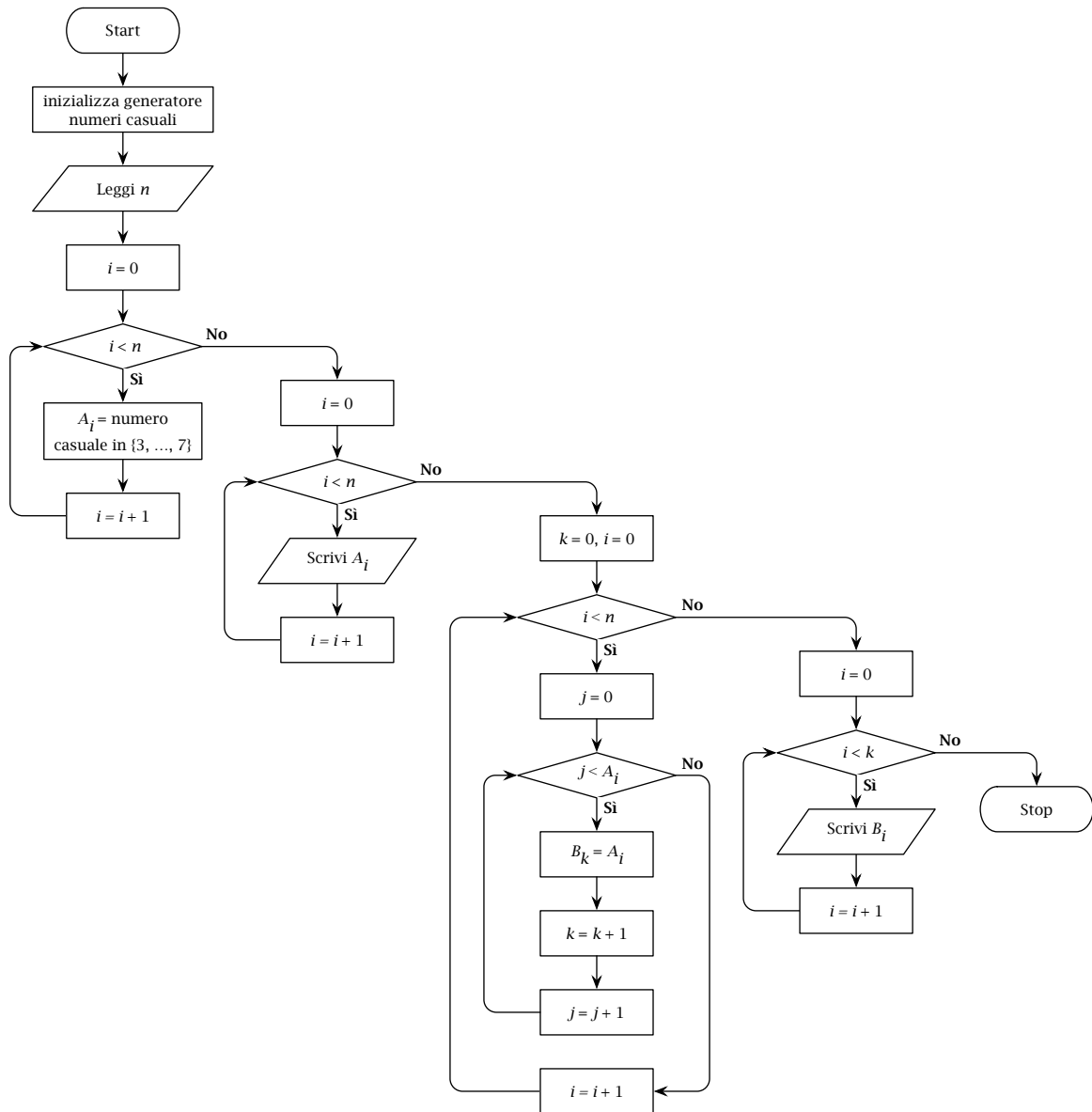
Esempio Siano $n = 4$ e $A = (3, 5, 3, 4)$. Allora $B = (3, 3, 3, 5, 5, 5, 5, 3, 3, 3, 4, 4, 4, 4)$.

Soluzione

Pseudo-codifica dell'algoritmo

- 1: inizializza il generatore dei numeri casuali
- 2: leggi n
- 3: per $i = 0, 1, \dots, n-1$ ripeti:
- 4: $A_i =$ numero casuale in $\{3, \dots, 7\}$
- 5: fine-ciclo
- 6: per $i = 0, 1, \dots, n-1$ ripeti:
- 7: scrivi A_i
- 8: fine-ciclo
- 9: $k = 0$
- 10: per $i = 0, \dots, n-1$ ripeti:
- 11: per $j = 0, \dots, A_i$ ripeti:
- 12: $B_k = A_i$
- 13: $k = k + 1$
- 14: fine-ciclo
- 15: fine-ciclo
- 16: per $i = 0, 1, \dots, k-1$ ripeti:
- 17: scrivi B_i
- 18: 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, h, k;
8     srand((unsigned)time(NULL));
9     printf("Numero di elementi: ");
10    scanf("%d", &n);
11    for (i=0; i<n; i++)
12        x[i] = rand() % 5 + 3;
13    return n;
14 }
15
16 void scriviArray(int x[], int n) {
17     int i;
18     for (i=0; i<n; i++)
19         printf("%d ", x[i]);
20     printf("\n");
21     return;
22 }
23
24 int arrayConRipetizioni(int A[], int B[], int n) {
25     int i, k = 0;
26     for (i=0; i<n; i++)
27         for (j=0; j<A[i]; j++)
28             B[k++] = A[i];
29     return k;
30 }
31
32 int main(void) {
33     int A[MAX], B[MAX], n, m;
34     n = arrayCasuale(A);
35     scriviArray(A, n);
36     m = arrayConRipetizioni(A, B, n);
37     scriviArray(B, m);
38     return 0;
39 }
```

Esercizio n. 2 (max 8 punti)

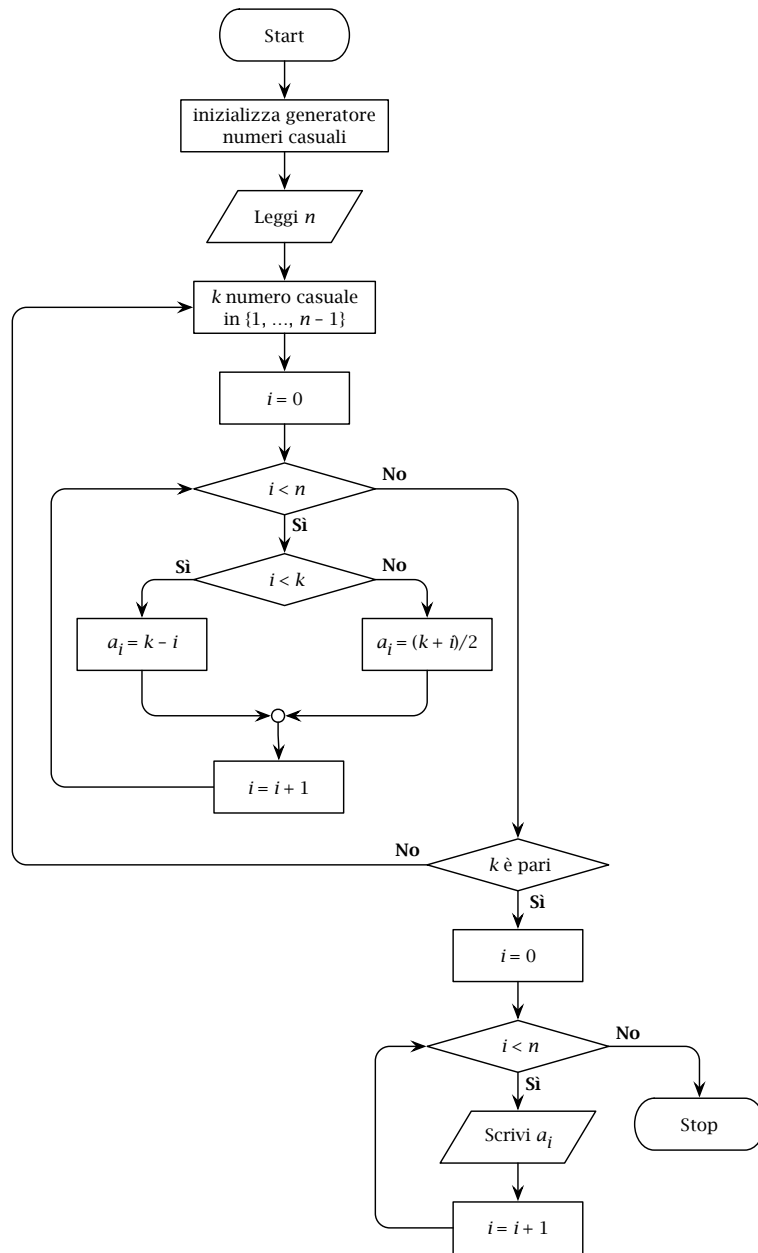
Scrivere il **diagramma di flusso** e la **codifica in linguaggio C** (un programma completo) del seguente algoritmo:

```
1:  inizializza il generatore di numeri casuali
2:  leggi  $n$ 
3:  ripeti:
4:    sia  $k$  un intero casuale tale che  $0 < k < n$ 
5:    per  $i = 0, \dots, n-1$  ripeti:
6:      se  $i < k$  allora
7:         $A_i = k - i$ 
8:      altrimenti
9:         $A_i = \frac{k+i}{2}$ 
10:     fine-condizione
11:  fine-ciclo
12:  fintanto che  $k$  non è pari
13:  per  $i = 0, \dots, n-1$  ripeti:
14:    scrivi  $A_i$ 
15:  fine-ciclo
```

Codifica in linguaggio C

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <time.h>
4
5  int main(void) {
6      int i, k, n A[100];
7      srand((unsigned)time(NULL));
8      scanf("%d", &n);
9      do {
10         k = rand() % (n-1) + 1;
11         for (i=0; i<n; i++) {
12             if (i < k)
13                 A[i] = k - i;
14             else
15                 A[i] = (k + i)/2;
16         }
17     } while (k % 2 == 1);
18     for (i=0; i<n; i++)
19         printf("%d ", A[i]);
20     return 0;
21 }
```

Diagramma di flusso



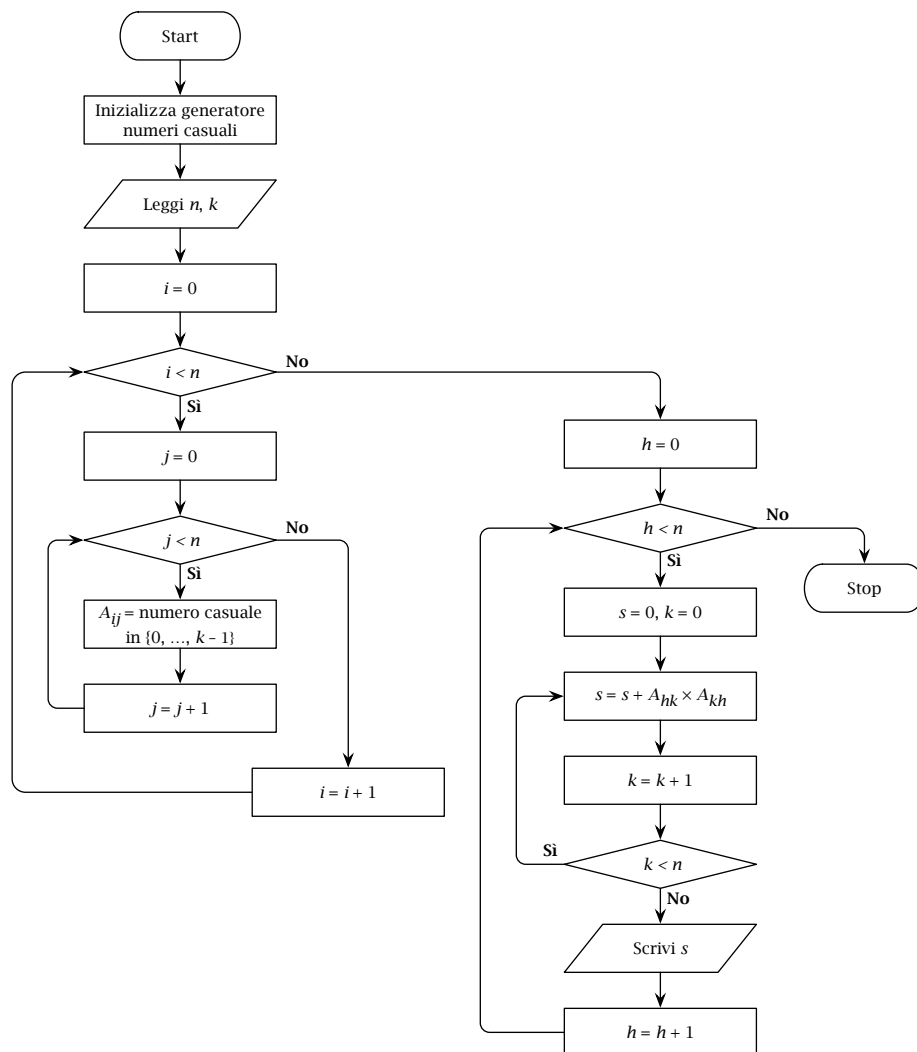
Esercizio n. 3 (max 6 punti)

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

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

```
1 void funzione(int A[100][100]) {
2     int i, j, h, k, n, s;
3     srand((unsigned)time(NULL));
4     scanf("%d %d", &n, &k);
5     for (i=0; i<n; i++)
6         for (j=0; j<n; j++)
7             A[i][j] = rand() % k;
8     h = 0;
9     while (h < n) {
10         s = 0;
11         k = 0;
12         do {
13             s = s + A[h][k] * A[k][h];
14             k++;
15         } while (k < n);
16         printf("s = %d\n", s);
17         h++;
18     }
19     return;
20 }
```

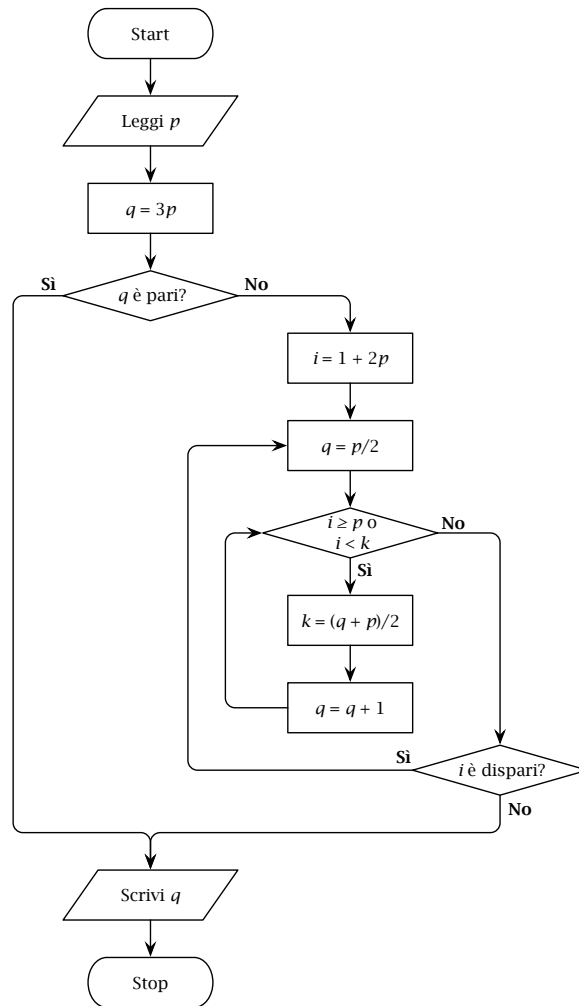
Soluzione



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
4 int main(void) {
5     int i, k, p, q;
6     scanf("%d", &p);
7     q = 3*p;
8     if (q % 2 != 0) {
9         i = 2*p + 1;
10        do {
11            q = p/2;
12            while (i >= p || i < k) {
13                k = (q+p)/2;
14                q = q + 1;
15            }
16        } while (i % 2 != 0);
17    }
18    printf("%d\n", q);
19    return 0;
20 }
```