Corso di Algoritmi e Strutture Dati (IN110) – Prof. Marco Liverani – a.a. 2026/2027

Prima prova di esonero – 20 novembre 2026

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*** *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 intero $n > 0$, costruire un array A di n numeri interi casuali compresi nell'insieme $\{30, 31, 32, \ldots, 50\}$. Visualizzare in output l'array A. Costruire un array B con n elementi tale che per $i = 0, 1, \ldots, n-1$ gli elementi b_i di B siano dati dal numero di elementi in $(a_0, a_1, \ldots, a_i)$ minori o uguali alla media tra il massimo e il minimo in $(a_0, a_1, \ldots, a_i)$. Visualizzare in output l'array B.

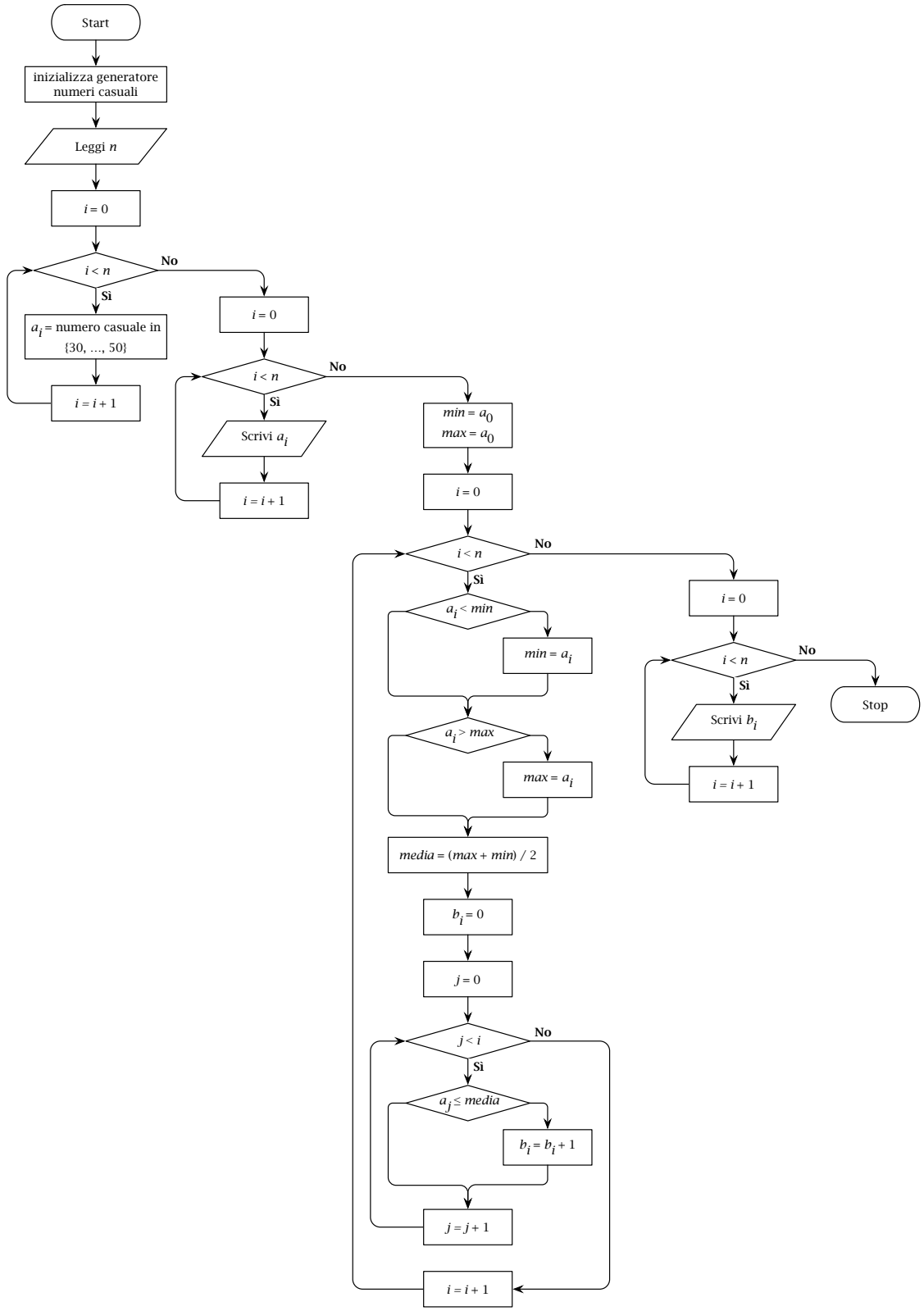
Esempio Siano $n = 6$ e $A = (38, 45, 31, 33, 47, 30)$. Allora $B = (1, 1, 2, 3, 3, 4)$.

Soluzione

Pseudo-codifica dell'algoritmo

inizializza il generatore dei numeri casuali
leggi n
per $i = 0, 1, \ldots, n-1$ ripeti:
    $a_i =$ numero casuale in $\{30, \ldots, 50\}$
fine-ciclo
per $i = 0, 1, \ldots, n-1$ ripeti:
    scrivi a_i
fine-ciclo
$min = a_0$, $max = a_0$
per $i = 0, 1, \ldots, n-1$ ripeti:
    se $a_i < min$ allora $min = a_i$
    se $a_i > max$ allora $max = a_i$
    $media = (min + max)/2$
    $b_i = 0$
    per $j = 0, 1, \ldots, i$ ripeti:
        se $a_j \leq media$ allora $b_j = b_j + 1$
    fine-ciclo
fine-ciclo
per $i = 0, 1, \ldots, k-1$ ripeti:
    scrivi B_i
fine-ciclo

Diagramma di flusso



Codifica in linguaggio C

```c
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#define MAX 1000

int arrayCasuale(int x[]) {
  int n, i;
  srand((unsigned)time(NULL));
  printf("Numero di elementi: ");
  scanf("%d", &n);
  for (i=0; i<n; i++)
    x[i] = rand() % 21 + 30;
  return n;
}

void scriviArray(int x[], int n) {
  int i;
  for (i=0; i<n; i++)
    printf("%d ", x[i]);
  printf("\n");
  return;
}

int main(void) {
  int A[MAX], B[MAX], n, i, j, min, max;
  float media;
  n = arrayCasuale(A);
  scriviArray(A, n);
  min = A[0];
  max = A[0];
  for (i=0; i<n; i++) {
    if (A[i] < min)
      min = A[i];
    if (A[i] > max)
      max = A[i];
    media = (min + max)/2.0;
    B[i] = 0;
    for (j=0; j<=i; j++)
      if (A[j] <= media)
        B[i] = B[i] + 1;
  }
  scriviArray(B, n);
  return 0;
}
```

Esercizio n. 2 **(max 8 punti)**

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

1: ripeti:
2: leggi n
3: fintanto che $n \leq 0$
4: ripeti:
5: leggi m
6: fintanto che $m \leq 0$
7: $s = 0$, $c = 0$
8: per $i = 0, \ldots, n-1$ ripeti:
9: $t = 0$, $j = -1$
10: fintanto che $j < m$ ripeti:
11: $x = i \times j$
12: scrivi x
13: $t = t + x$
14: $s = s + x$
15: se x è divisibile per 3 allora
16: $c = c + 1$
17: fine-condizione
18: $j = j + 1$
19: fine-ciclo
20: scrivi t
21: fine-ciclo
22: scrivi s, c

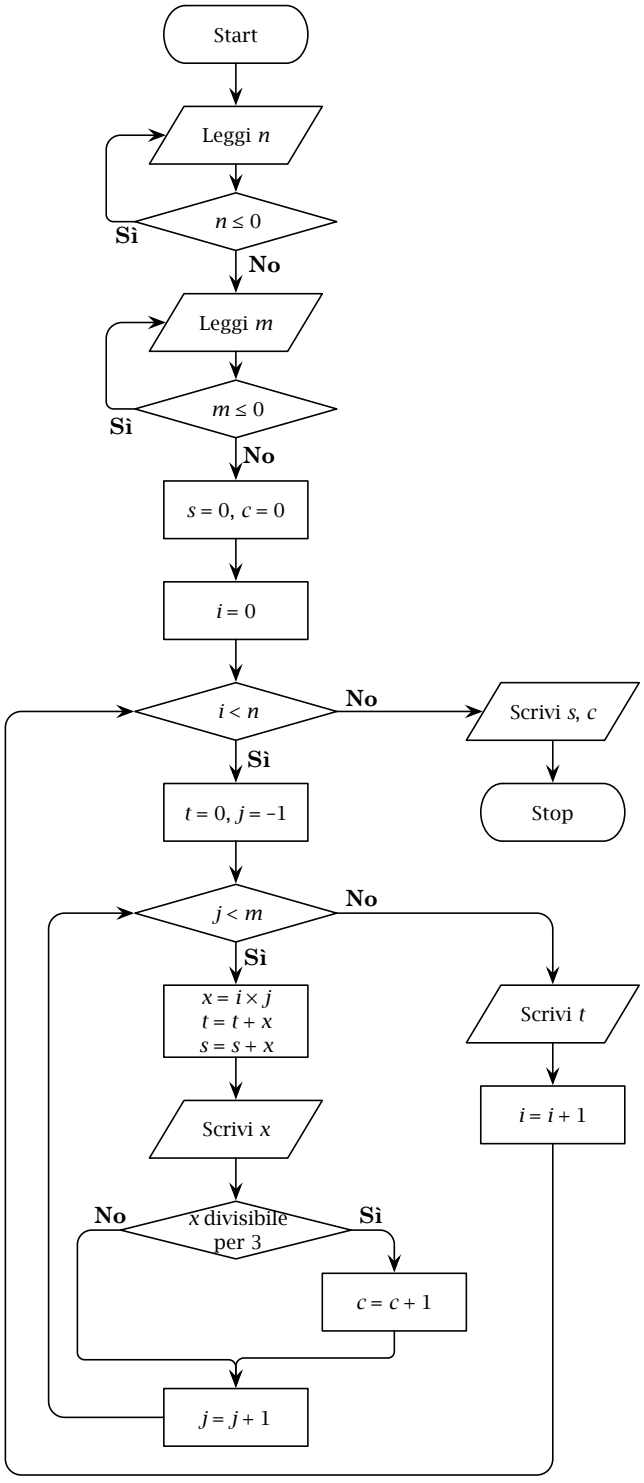
Codifica in linguaggio C

```c
#include <stdlib.h>
#include <stdio.h>

int main(void) {
  int i, j, n, m, s = 0, c = 0, t, x;
  do {
    scanf("%d", &n);
  } while (n <= 0);
  do {
    scanf("%d", &m);
  } while (m <= 0);
  for (i=0; i<n; i++) {
    t = 0;
    j = -1;
    while (j < m) {
      x = i*j;
      printf("%d ", x);
      t = t + x;
      s = s + x;
      if (x % 3 == 0)
        c = c + 1;
      j = j + 1;
    }
    printf("%d ", t);
  }
```

```
26    printf("%d %d\n", s, c);
27    return 0;
28  }
```

Diagramma di flusso



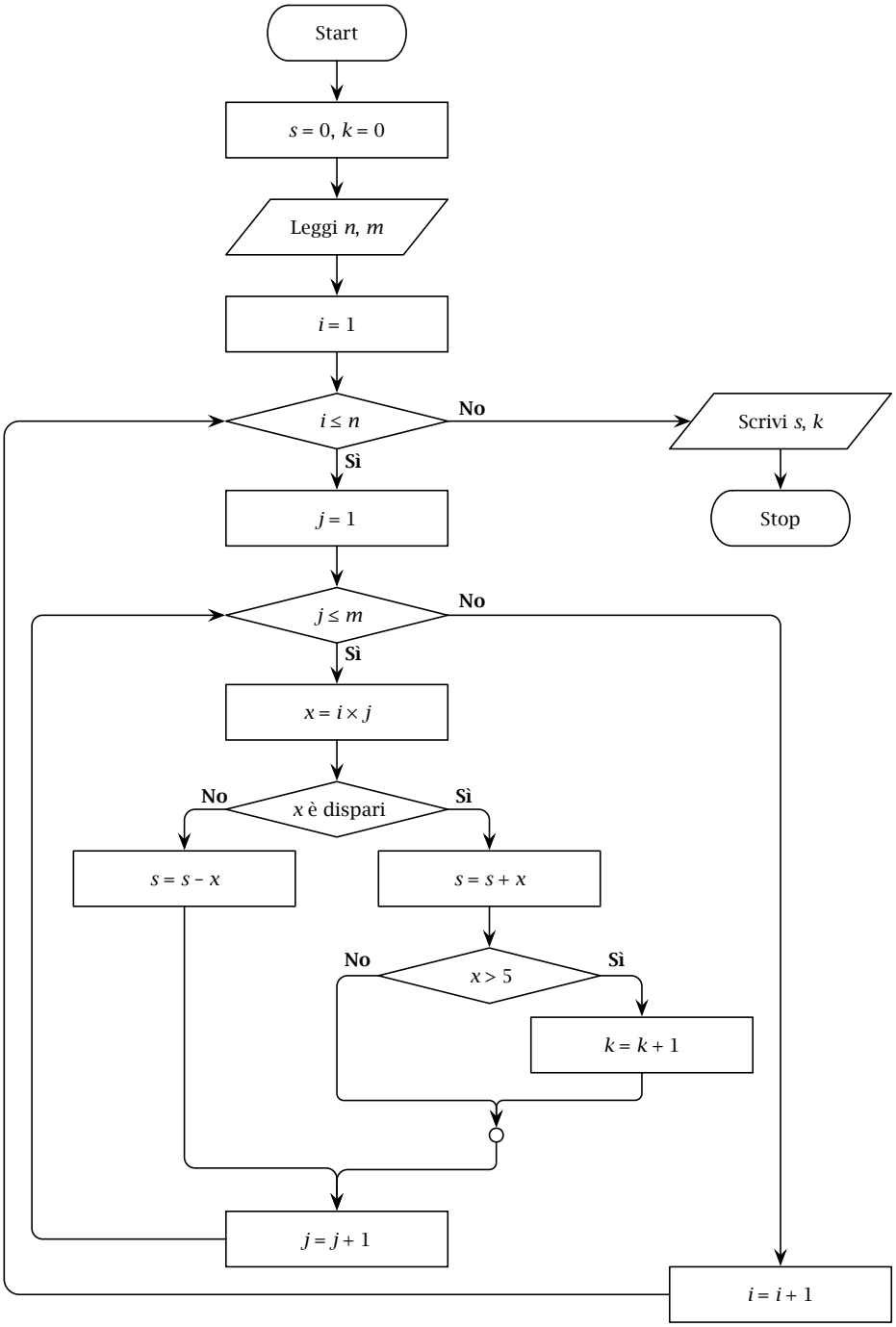
Esercizio n. 3 (max 6 punti)

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

```c
#include <stdio.h>
#include <stdlib.h>

int main(void) {
  int i, j, n, m, x, s = 0, k = 0;
  scanf("%d %d", &n, &m);
  for (i = 1; i <= n; i++) {
    j = 1;
    while (j <= m) {
      x = i * j;
      if (x % 2 == 1) {
        s = s + x;
        if (x > 5)
          k++;
      } else {
        s = s - x;
      }
      j++;
    }
  }
  printf("s = %d, k = %d\n", s, k);
  return 0;
}
```

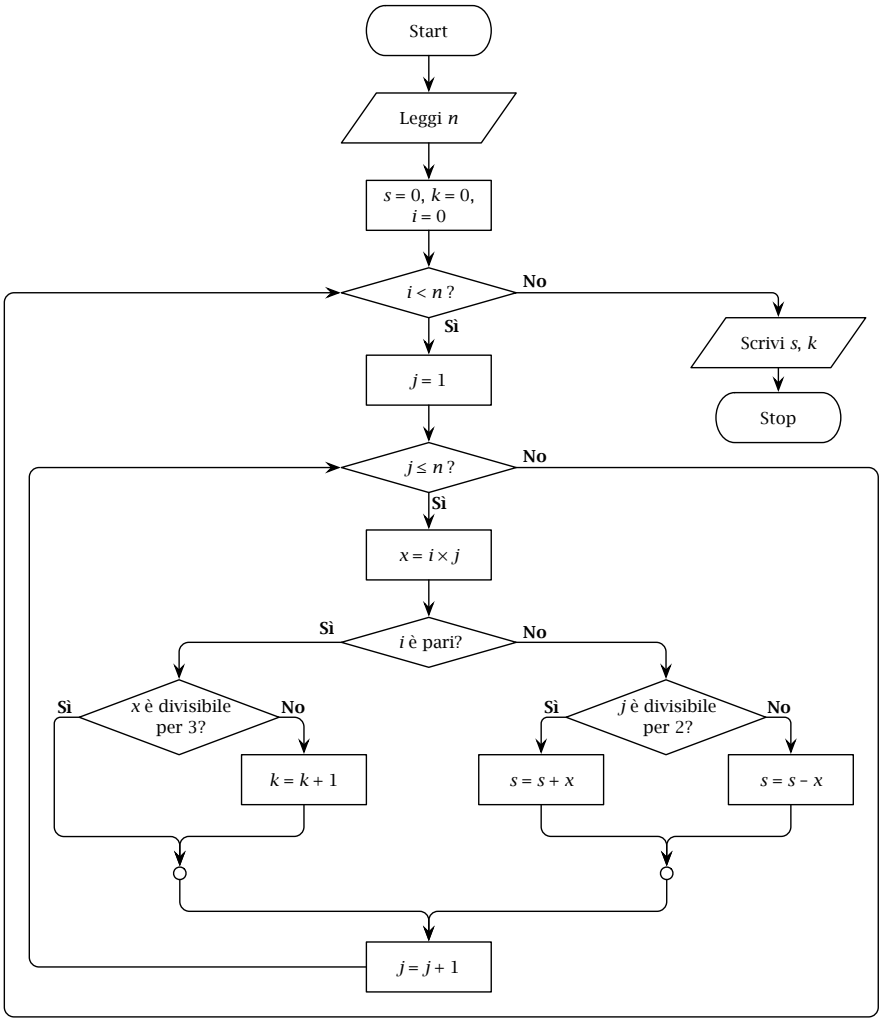
Soluzione


Start
$s = 0,\ k = 0$
Leggi n, m
$i = 1$
$i \le n$
No
Sì
Scrivi s, k
Stop
$j = 1$
$j \le m$
No
Sì
$x = i \times j$
No
x è dispari
Sì
$s = s - x$
$s = s + x$
No
$x > 5$
Sì
$k = k + 1$
$j = j + 1$
$i = i + 1$

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

```c
#include <stdlib.h>
#include <stdio.h>

int main(void) {
  int i, k, p, q;
  scanf("%d", &p);
  q = 3*p;
  if (q % 2 != 0) {
    i = 2*p + 1;
    do {
      q = p/2;
      while (i >= p || i < k) {
        k = (q+p)/2;
        q = q + 1;
      }
    } while (i % 2 != 0);
  }
  printf("%d\n", q);
  return 0;
}
```