

Corso di Informatica 1 (IN110) – Prof. Marco Liverani – a.a. 2021/2022

## Prima prova di esonero – 12 novembre 2021

*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.

Letta in input una stringa  $S$  eliminare, se esistono, tutte le lettere ripetute da  $S$  e stampare la nuova stringa  $S$ .

**Esempio** Sia  $S = \text{"precipitevolissimevolmente"}$ . La stringa priva di lettere ripetute prodotta dal programma è la seguente:  $S = \text{"precitvolsmn"}$ .

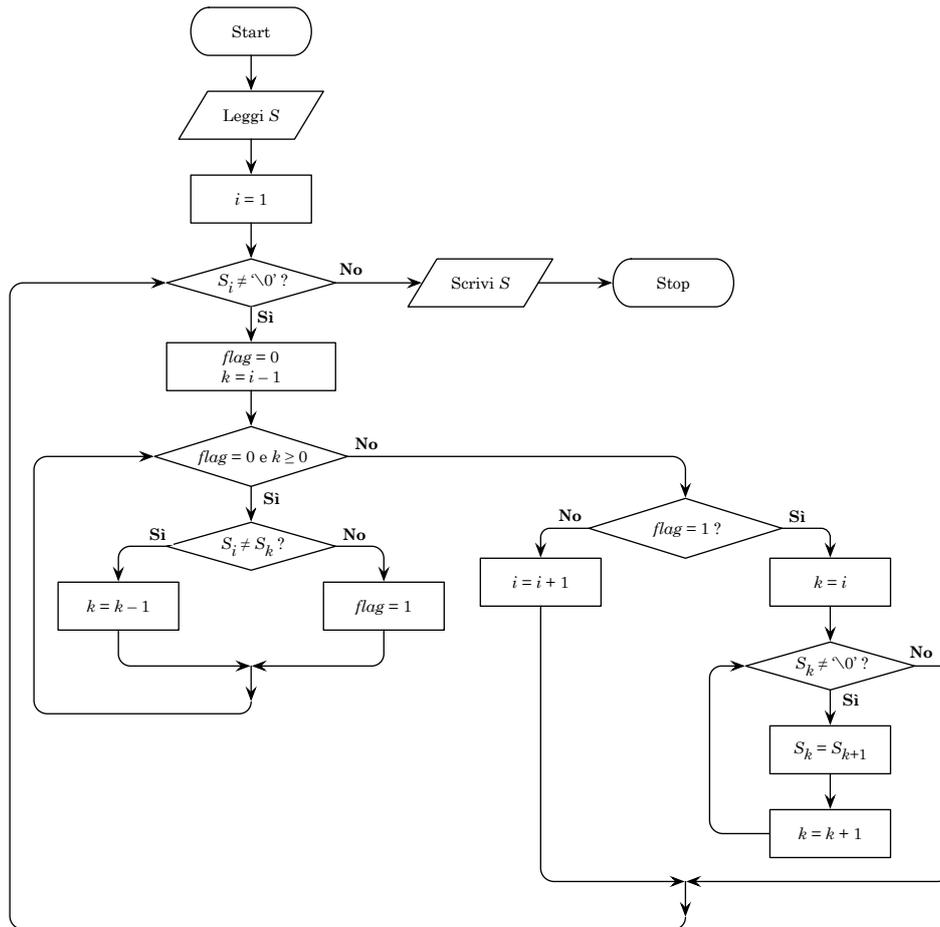
### Soluzione

#### Pseudo-codifica dell'algoritmo

```
1: leggi  $S$ 
2:  $i = 1$ 
3: fintanto che  $S_i \neq '\emptyset'$  ripeti:
4:    $flag = 0$ 
5:    $k = i - 1$ 
6:   fintanto che  $flag = 0$  e  $k \geq 0$  ripeti:
7:     se  $S_i \neq S_k$  allora
8:        $k = k - 1$ 
9:     altrimenti
10:       $flag = 1$ 
11:   fine-condizione
12: fine-ciclo
13: se  $flag = 1$  allora
14:    $k = i$ 
15:   fintanto che  $S_k \neq '\emptyset'$  ripeti:
16:      $S_k = S_{k+1}$ 
17:      $k = k + 1$ 
18:   fine-ciclo
```

- 19: altrimenti
- 20:  $i = i + 1$
- 21: fine-condizione
- 22: fine-ciclo
- 23: scrivi  $S$

**Diagramma di flusso**



## Codifica in linguaggio C

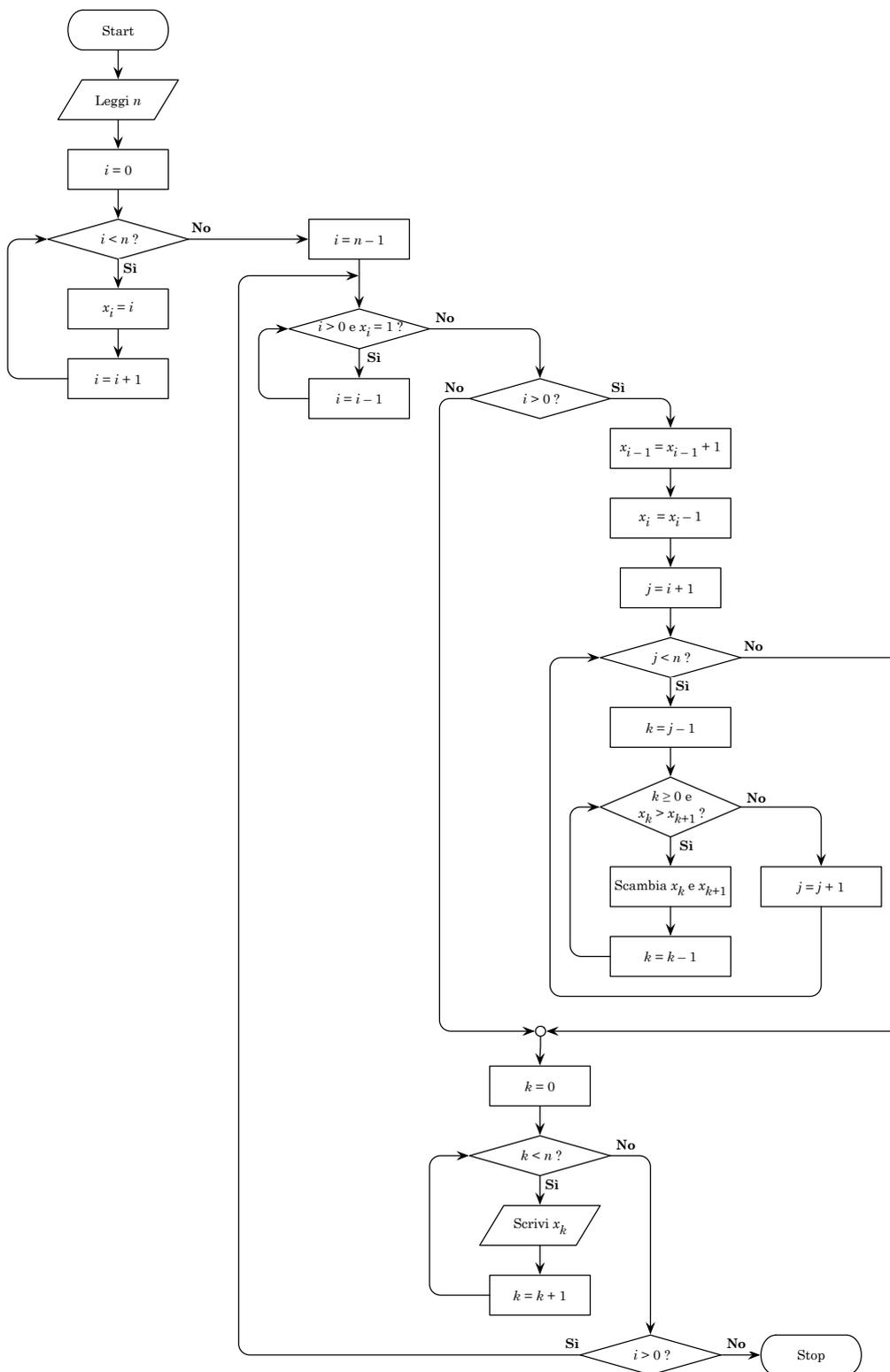
```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 int main(void) {
6     char S[MAX];
7     int i, k, flag;
8     printf("inserisci una parola: ");
9     scanf("%s", S);
10    i = 1;
11    while (S[i] != '\0') {
12        flag = 0;
13        k = i-1;
14        while (flag == 0 && k >= 0)
15            if (S[i] != S[k])
16                k = k-1;
17            else
18                flag = 1;
19        if (flag == 1) {
20            k = i;
21            while (S[k] != '\0') {
22                S[k] = S[k+1];
23                k = k+1;
24            }
25        } else
26            i = i+1;
27    }
28    printf("stringa priva di lettere duplicate: %s\n", S);
29    return(0);
30 }
```

## 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: per  $i = 0, 1, \dots, n - 1$  ripeti:
3:    $x_i = i$ 
4: fine-ciclo
5:  $i = n - 1$ 
6: ripeti:
7:   fintanto che  $i > 0$  e  $x_i = 1$  ripeti:
8:      $i = i - 1$ 
9:   fine-ciclo
10: se  $i > 0$  allora
11:    $x_{i-1} = x_{i-1} + 1$ 
12:    $x_i = x_i - 1$ 
13:   per  $j = i + 1, \dots, n - 1$  ripeti:
14:      $k = j - 1$ 
15:     fintanto che  $k \geq 0$  e  $x_k > x_{k+1}$  ripeti:
16:       scambia  $x_k, x_{k+1}$ 
17:        $k = k - 1$ 
18:     fine-ciclo
19:   fine-ciclo
20: fine-condizione
21: per  $k = 0, \dots, n - 1$  ripeti:
22:   scrivi  $x_k$ 
23: fine-ciclo
24: fintanto che  $i > 0$ 
```

# Diagramma di flusso



## Codifica in linguaggio C

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

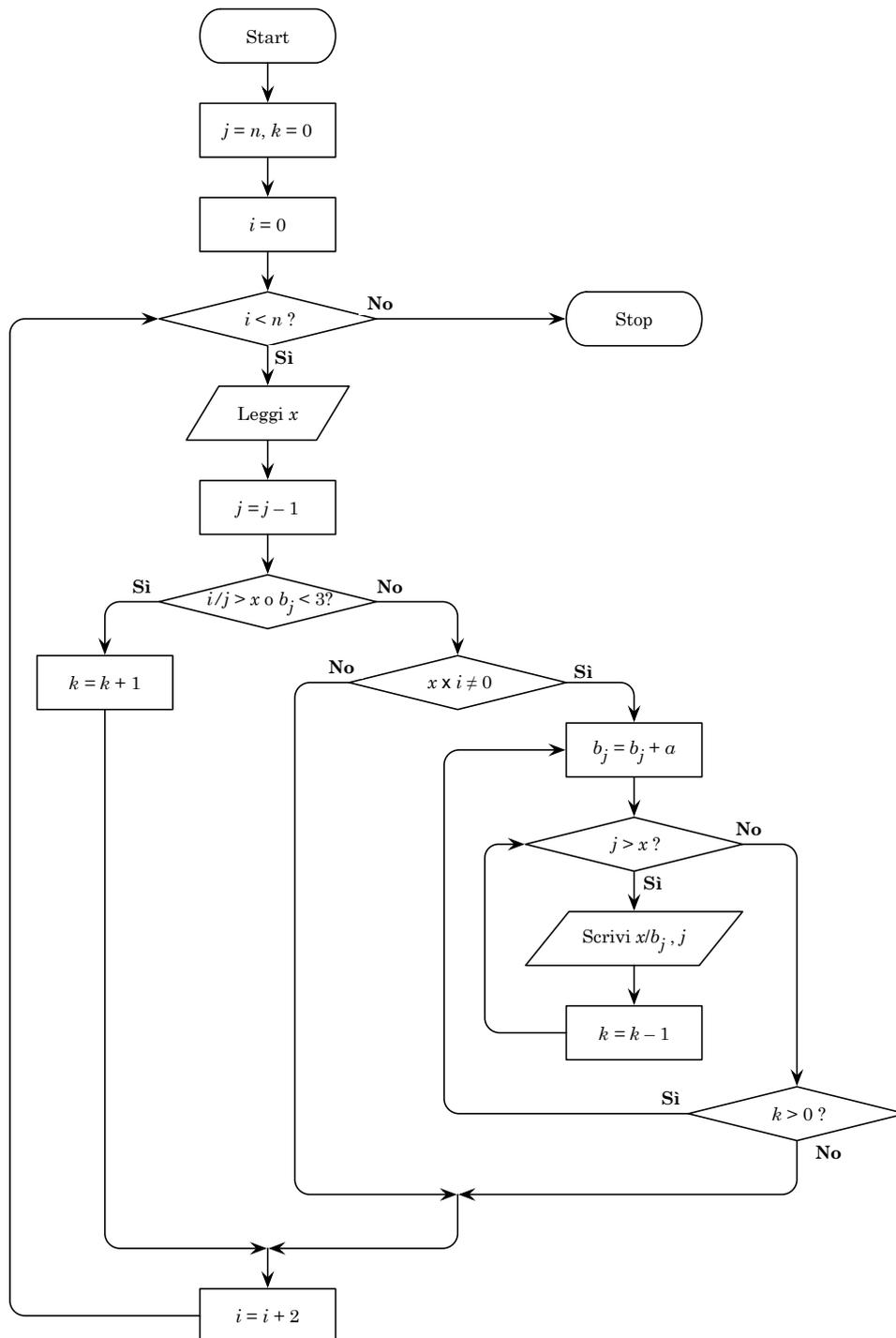
### Esercizio n. 3 (max 6 punti)

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, int b[], int n) {
2   int i, j=n, k=0;
3   float x;
4   for (i=0; i < n; i+=2) {
5     scanf("%f", &x);
6     j--;
7     if (i/j > x || b[j] < 3)
8       k++;
9     else if (x*i != 0) {
10      do {
11        b[j] = b[j] + a;
12        while (j > x) {
13          printf("%f %d", x/b[j], j);
14          k--;
15        }
16      } while(k>0);
17    }
18  }
19  return;
20 }
```

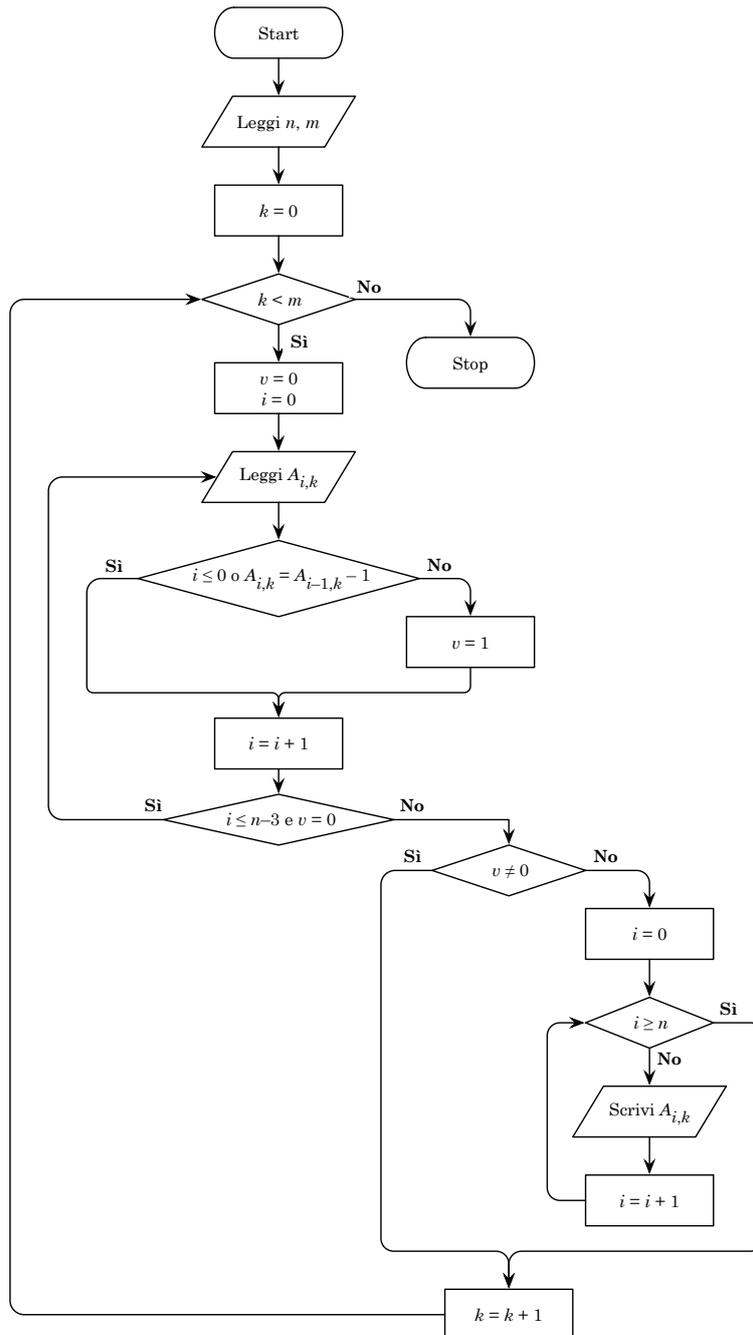
## Soluzione



### 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 #define MAX 100
4
5 int main(void) {
6     int n, m, i, k, v, A[MAX][MAX];
7     scanf("%d %d", &n, &m);
8     for (k=0; k<m; k++) {
9         v = 0;
10        i = 0;
11        do {
12            scanf("%d", &A[i][k]);
13            if (i > 0 && A[i][k] != A[i-1][k] - 1)
14                v = 1;
15            i++;
16        } while (i <= n-3 && v == 0);
17        if (v == 0)
18            for (i=0; i<n; i++)
19                printf("%d", A[i][k]);
20    }
21    return(0);
22 }
```