

Prima prova di esonero – 11 novembre 2022

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 $\{-k, -(k-1), -(k-2), \dots, -2, -1, 0, 1, 2, \dots, k-2, k-1, k\}$. Costruire un secondo array B con n elementi, tali che, per ogni $i = 0, 1, \dots, n-1$, B_i sia il massimo in $\{A_i, A_{i+1}, \dots, A_{n-1}\}$. Stampare in output i due vettori A e B .

Esempio Siano $n = 10$ e $k = 30$. Sia inoltre $A = (-30, 28, -22, 15, 27, -1, -18, 25, -29, 16)$. Allora risulta $B = (28, 28, 27, 27, 27, 25, 25, 25, 16, 16)$.

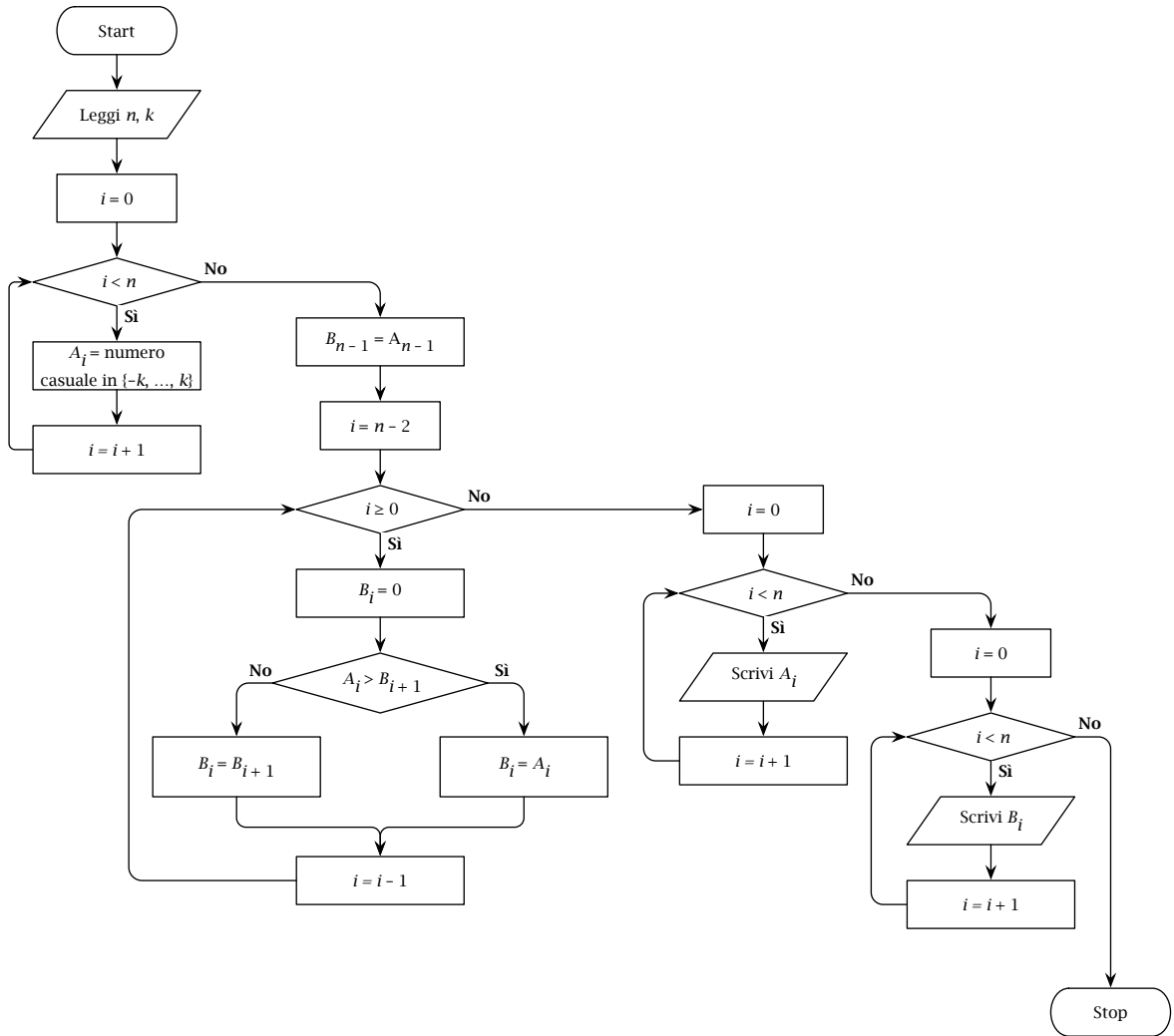
Soluzione

Pseudo-codifica dell'algoritmo

- 1: leggi n e k
- 2: per $i = 0, 1, \dots, n-1$ ripeti:
- 3: $A_i =$ numero casuale in $\{-k, \dots, 0, \dots, k\}$
- 4: fine-ciclo
- 5: $B_{n-1} = A_{n-1}$
- 6: per $i = n-2, n-3, \dots, 2, 1, 0$ ripeti:
- 7: se $A_i > B_{i+1}$ allora
- 8: $B_i = A_i$
- 9: altrimenti
- 10: $B_i = B_{i+1}$
- 11: fine-condizione
- 12: fine-ciclo
- 13: per $i = 0, 1, \dots, n-1$ ripeti:
- 14: scrivi A_i
- 15: fine-ciclo
- 16: per $i = 0, 1, \dots, n-1$ ripeti:

- 17: scrivi B_i
- 18: fine-ciclo

Diagramma di flusso



Codifica in linguaggio C

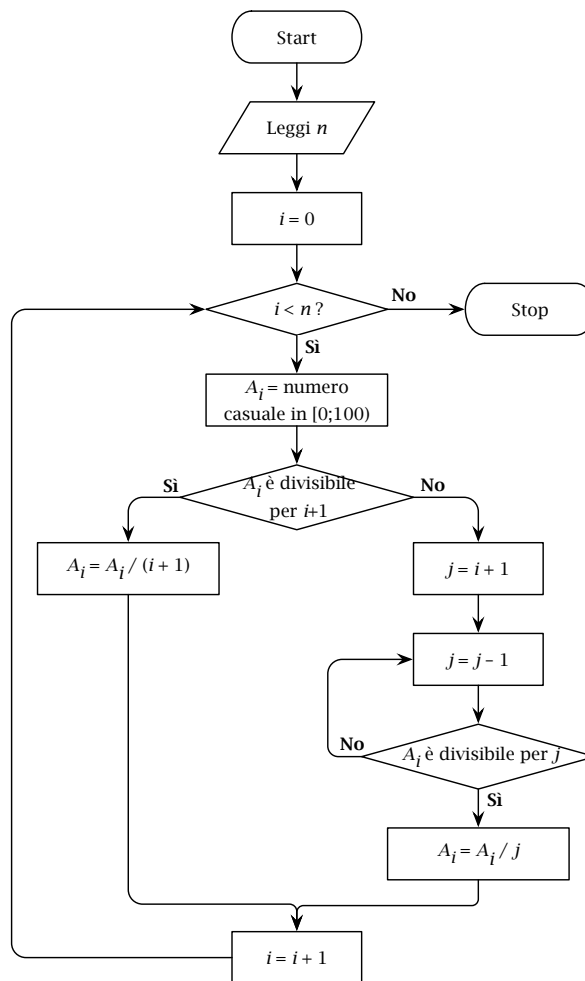
```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 100
5
6 void stampaArray(int V[], int n) {
7     int i;
8     for (i=0; i<n; i++)
9         printf("%3d ", V[i]);
10    printf("\n");
11    return;
12 }
13
14 int generaArray(int V[]) {
15     int n, k, i;
16     printf("Inserisci due interi: ");
17     scanf("%d %d", &n, &k);
18     for (i=0; i<n; i++)
19         V[i] = rand() % (2*k + 1) - k;
20     return(n);
21 }
22
23 void costruisci(int A[], int B[], int n) {
24     int i;
25     B[n-1] = A[n-1];
26     for (i=n-2; i>=0; i--)
27         if (A[i] > B[i+1])
28             B[i] = A[i];
29         else
30             B[i] = B[i+1];
31     return;
32 }
33
34 int main(void) {
35     int A[MAX], B[MAX], n;
36     srand((unsigned)time(NULL));
37     n = generaArray(A);
38     costruisci(A, B, n);
39     stampaArray(A, n);
40     stampaArray(B, n);
41     return(0);
42 }
```

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: $A_i =$ numero casuale intero positivo minore di 100
 - 4: se A_i è divisibile per $i + 1$ allora
 - 5: $A_i = A_i / (i + 1)$
 - 6: altrimenti
 - 7: $j = i + 1$
 - 8: ripeti:
 - 9: $j = j - 1$
 - 10: fintanto che A_i non è divisibile per j
 - 11: $A_i = A_i / j$
 - 12: fine-condizione
 - 13: 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, j, n;
8     srand((unsigned)time(NULL));
9     scanf("%d", &n);
10    for (i=0; i<n; i++) {
11        A[i] = rand() % 100;
12        if (A[i] % (i+1) == 0) {
13            A[i] = A[i] / (i+1);
14        } else {
15            j = i+1;
16            do {
17                j--;
18            } while (A[i] % j != 0);
19            A[i] = A[i] / j;
20        }
21    }
22    return(0);
23 }
```

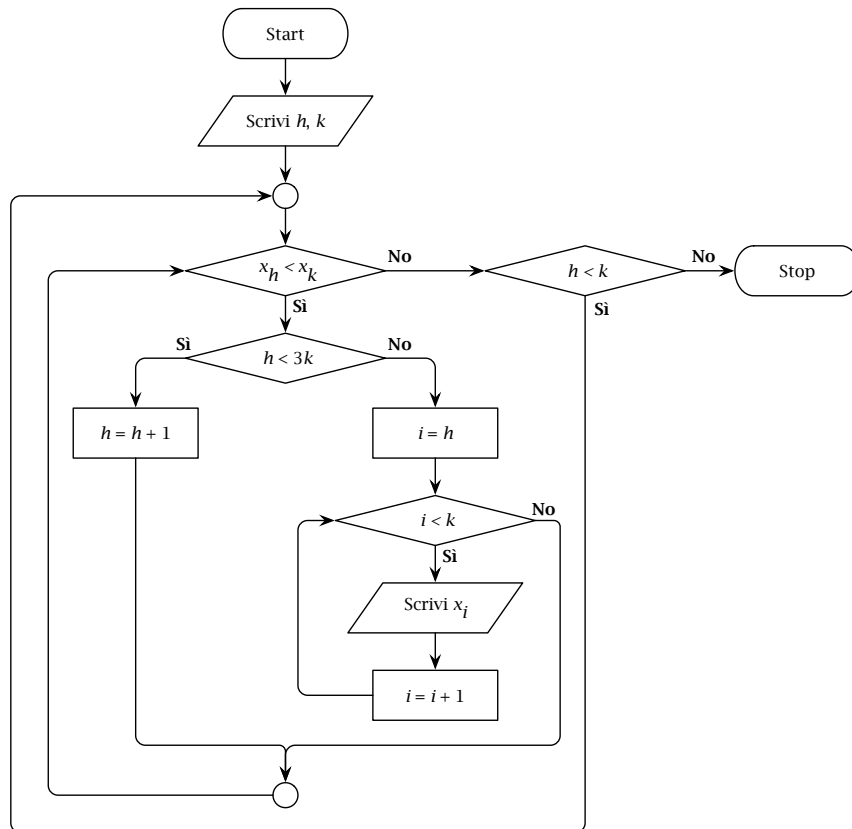
Esercizio n. 3 (max 6 punti)

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 x[], int h, int k) {  
2   int i;  
3   printf("%d %d\n", h, k);  
4   do {  
5     while (x[h] < x[k]) {  
6       if (h < 3*k)  
7         h++;  
8       else  
9         for (i=h; i<k; i++)  
10          printf("%f\n", x[i]);  
11    }  
12  } while (h<k);  
13  return;  
14 }
```

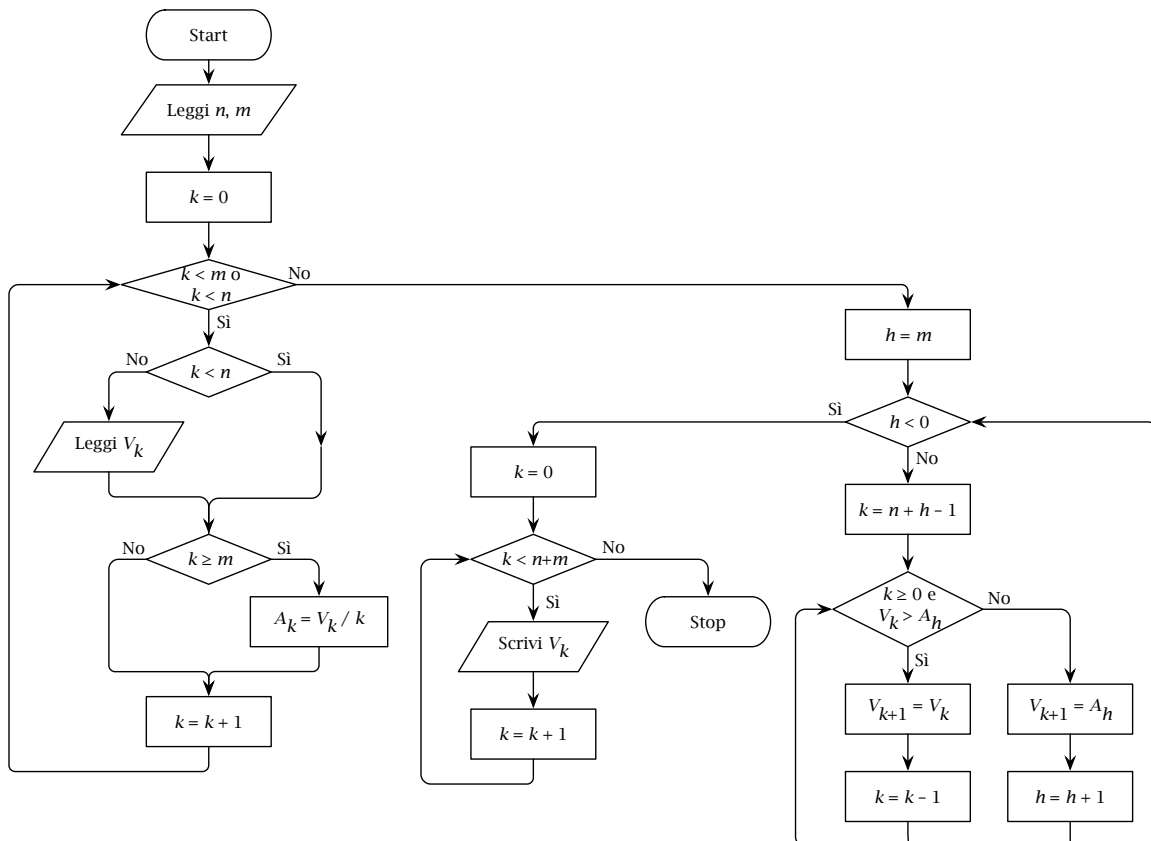
Soluzione



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