

Esame scritto del 19 Giugno 2026 (Appello C)

Si richiede di risolvere entrambi gli esercizi riportando una codifica in linguaggio C completa dei due programmi. Nel caso in cui non si riesca a completare entrambi gli esercizi si suggerisce di riportare almeno la codifica in C delle funzioni principali o una loro pseudo-codifica. È possibile consultare libri e appunti personali, ma non scambiare libri o appunti con altri studenti o utilizzare strumenti digitali (tablet, smartphone, computer, ecc.). I compiti che presenteranno evidenti ed anomale «similitudini» saranno annullati. La prova scritta ha una durata di tre ore, durante le quali non è consentito allontanarsi dall'aula, se non dopo aver consegnato il compito.

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.

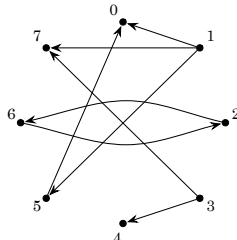
Esercizio n. 1

Letti in input due interi $n, k > 0$ costruire un vettore $A = \{a_0, a_1, \dots, a_{n-1}\}$ di numeri interi casuali maggiori di zero e minori di k . Costruire le liste di adiacenza del grafo orientato $G = (V, E)$ tale che $V(G) = \{0, 1, \dots, n-1\}$ ed $E(G) = \{(u, v) : u, v \in V(G), a_v = ha_u \text{ per qualche } h \in \mathbb{N}, h > 0\}$. Stampare il vettore A e le liste di adiacenza del grafo G .

Esempio Si consideri il caso in cui $n = 8$ e $k = 10$; il vettore di numeri casuali minori di k è il seguente:

$$A = (a_0 = 8, a_1 = 2, a_2 = 7, a_3 = 3, a_4 = 9, a_5 = 4, a_6 = 7, a_7 = 6)$$

Il grafo $G = (V, E)$ è dato dai seguenti insiemi ed è rappresentato in figura: $V(G) = \{0, 1, 2, 3, 4, 5, 6, 7\}$, $E(G) = \{(1, 0), (1, 5), (1, 7), (2, 6), (3, 4), (3, 7), (5, 0), (6, 2)\}$.



Soluzione

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 100
5
6 struct nodo {
7     int info;
8     struct nodo *next;
9 };
10
11 int costruisciVettore(int A[]) {
12     int n, i, k;
13     printf("Numero di elementi: ");
14     scanf("%d", &n);
15     printf("Valore soglia: ");
16     scanf("%d", &k);

```

```

17  srand((unsigned)time(NULL));
18  for (i=0; i<n; i++)
19      A[i] = (rand() % (k-1)) + 1;
20  return n;
21  }
22
23  void stampaVettore(int A[], int n) {
24      int i;
25      for (i=0; i<n; i++)
26          printf("%d ", A[i]);
27      printf("\n");
28      return;
29  }
30
31  void costruisciGrafo(struct nodo *G[], int A[], int n) {
32      int i, j;
33      struct nodo *p;
34      for (i=0; i<n; i++)
35          G[i] = NULL;
36      for (i=0; i<n; i++) {
37          for (j=0; j<n; j++) {
38              if (i != j && A[j] % A[i] == 0) {
39                  p = malloc(sizeof(struct nodo));
40                  p->info = j;
41                  p->next = G[i];
42                  G[i] = p;
43              }
44          }
45      }
46      return;
47  }
48
49  void stampaLista(struct nodo *p) {
50      while (p != NULL) {
51          printf("%d --> ", p->info);
52          p = p->next;
53      }
54      printf("NULL\n");
55      return;
56  }
57
58  void stampaGrafo(struct nodo *G[], int n) {
59      for (int i=0; i<n; i++) {
60          printf("%d: ", i);
61          stampaLista(G[i]);
62      }
63      return;
64  }
65
66  int main(void) {
67      int n, A[MAX];
68      struct nodo *G[MAX];
69      n = costruisciVettore(A);
70      stampaVettore(A, n);
71      costruisciGrafo(G, A, n);
72      stampaGrafo(G, n);
73      return 0;
74  }

```

Esercizio n. 2

Letti in input tre numeri interi positivi n, h, k , con $h < k$, costruire una sequenza di n numeri casuali compresi tra h e k (estremi inclusi) e memorizzarla nella prima colonna di una matrice quadrata A di ordine n . Costruire le rimanenti $n - 1$ colonne della matrice A sulla base della seguente formula:

$$a_{i,j} = \sum_{p=n-1}^i a_{p,j-1} \text{ per } i = 0, 1, 2, \dots, n-1 \text{ e } j = 1, 2, \dots, n-1$$

Stampare la matrice A .

Esempio Sia $n = 5, h = 4, k = 10$. La matrice A è la seguente:

$$A = \begin{pmatrix} 7 & 33 & 100 & 232 & 458 \\ 4 & 26 & 67 & 132 & 226 \\ 8 & 22 & 41 & 65 & 94 \\ 9 & 14 & 19 & 24 & 29 \\ 5 & 5 & 5 & 5 & 5 \end{pmatrix}$$

Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 100
5
6 void costruisciMatrice(int A[MAX][MAX], int n, int h, int k) {
7     int i, j, p;
8     srand((unsigned)time(NULL));
9     for (i=0; i<n; i++)
10        A[i][0] = (rand() % (k-h+1)) + h;
11    for (j=1; j<n; j++) {
12        for (i=0; i<n; i++) {
13            A[i][j] = 0;
14            for (p=n-1; p>=i; p--)
15                A[i][j] += A[p][j-1];
16        }
17    }
18    return;
19 }
20
21 void stampaMatrice(int A[MAX][MAX], int n) {
22     int i, j;
23     for (i=0; i<n; i++) {
24         for (j=0; j<n && A[i][j]>0; j++)
25             printf("%3d ", A[i][j]);
26         printf("\n");
27     }
28     return;
29 }
30
31 int main(void) {
32     int n, h, k, A[MAX][MAX];
33     printf("Ordine della matrice: ");
34     scanf("%d", &n);
35     printf("Valori soglia: ");
36     scanf("%d %d", &h, &k);
37     costruisciMatrice(A, n, h, k);
```

```
38 stampaMatrice(A, n);  
39 return 0;  
40 }
```