

Esame scritto del 12 Febbraio 2026 (Appello B)

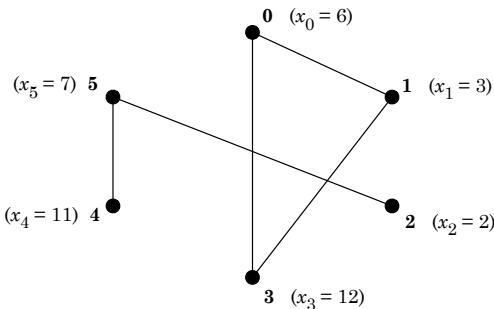
Si richiede di risolvere entrambi gli esercizi riportando una codifica in linguaggio C completa dei due programmi. Nel caso in cui non si riesca a completare entrambi gli esercizi si suggerisce di riportare almeno la codifica in C delle funzioni principali o una loro pseudo-codifica. È possibile consultare 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

Letto in input un intero $n > 0$, generare un array $X = (x_0, x_1, \dots, x_{n-1})$ di n interi positivi casuali in $\{0, 1, 2, \dots, 17\}$. Costruire le liste di adiacenza del grafo orientato $G = (V, E)$ con n vertici $V = (0, 1, \dots, n - 1)$ tale che $(i, j) \in E(G) \iff x_i + x_j$ è divisibile per 3. Stampare le liste di adiacenza di G e il vettore X .

Esempio Sia $n = 6$ e si consideri il vettore di numeri casuali $X = (x_0 = 6, x_1 = 3, x_2 = 2, x_3 = 12, x_4 = 11, x_5 = 7)$; allora il grafo G prodotto dal programma è rappresentato nella seguente figura.

**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 arrayCasuale(int x[]) {
12     int i, n;
13     srand((unsigned)time(NULL));
14     printf("Numero di vertici del grafo: ");
15     scanf("%d", &n);
16     for (i=0; i<n; i++)
17         x[i] = rand() % 18;
18     return n;
19 }
  
```

```

20
21 void stampaArray(int x[], int n) {
22     for (int i=0; i<n; i++)
23         printf("%d ", x[i]);
24     printf("\n");
25     return;
26 }
27
28 void stampaLista(struct nodo *p) {
29     while (p != NULL) {
30         printf("%d --> ", p->info);
31         p = p->next;
32     }
33     printf("Null\n");
34     return;
35 }
36
37 void stampaGrafo(struct nodo *V[], int n) {
38     for (int i=0; i<n; i++) {
39         printf("%2d: ", i);
40         stampaLista(V[i]);
41     }
42     return;
43 }
44
45 void aggiungiSpigolo(struct nodo *V[], int i, int j) {
46     struct nodo *p;
47     p = malloc(sizeof(struct nodo));
48     p->info = j;
49     p->next = V[i];
50     V[i] = p;
51     p = malloc(sizeof(struct nodo));
52     p->info = i;
53     p->next = V[j];
54     V[j] = p;
55     return;
56 }
57
58 void costruisciGrafo(struct nodo *V[], int n, int x[]) {
59     for (int i=0; i<n; i++)
60         V[i] = NULL;
61     for (int i=0; i<n-1; i++)
62         for (int j=i+1; j<n; j++)
63             if ((x[i]+x[j]) % 3 == 0)
64                 aggiungiSpigolo(V, i, j);
65     return;
66 }
67
68 int main(void) {
69     struct nodo *V[MAX];
70     int n, X[MAX];
71     n = arrayCasuale(X);
72     costruisciGrafo(V, n, X);
73     stampaGrafo(V, n);
74     stampaArray(X, n);
75     return 0;
76 }

```

Esercizio n. 2

Letto in input due numeri interi $n, m > 2$ costruire una matrice A di n righe ed m colonne di numeri interi così composta:

- la prima riga sia composta da numeri interi casuali nell'insieme $\{1, 2, \dots, m\}$;
- gli elementi $a_{i,j}$ delle righe successive sono dati dalla somma degli elementi delle righe precedenti e delle colonne dalla 0 alla j .

Visualizzare in output la matrice A .

Esempio Siano $n = 4$ e $m = 6$. Si consideri la sequenza di m numeri casuali $A_0 = (3, 5, 6, 1, 3, 2)$. La matrice A è allora composta come segue:

$$A = \begin{pmatrix} 3 & 5 & 6 & 1 & 3 & 2 \\ 3 & 8 & 14 & 15 & 18 & 20 \\ 6 & 19 & 39 & 55 & 76 & 98 \\ 12 & 44 & 103 & 174 & 271 & 391 \end{pmatrix}$$

Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 50
5
6 void costruisciMatrice(int A[MAX][MAX], int n, int m) {
7     int i, j, h, k;
8     srand((unsigned)time(NULL));
9     for (j = 0; j < m; j++)
10        A[0][j] = rand() % m + 1;
11    for (i=1; i<n; i++) {
12        for (j=0; j<m; j++) {
13            A[i][j] = 0;
14            for (h=0; h<i; h++)
15                for (k=0; k<=j; k++)
16                    A[i][j] = A[i][j] + A[h][k];
17        }
18    }
19    return;
20 }
21
22 void stampaMatrice(int A[MAX][MAX], int n, int m) {
23     for (int i=0; i<n; i++) {
24         for (int j=0; j<m; j++)
25             printf("%4d ", A[i][j]);
26         printf("\n");
27     }
28     return;
29 }
30
31 int main(void) {
32     int m, n, A[MAX][MAX];
33     printf("Inserisci due interi positivi n e m: ");
34     scanf("%d %d", &n, &m);
35     costruisciMatrice(A, n, m);
36     stampaMatrice(A, n, m);
37     return 0;
38 }
```