

Corso di Algoritmi e Strutture Dati (IN110) – Prof. Marco Liverani – a.a. 2024/2025

## Esame scritto del 14 Febbraio 2025 (Appello B)

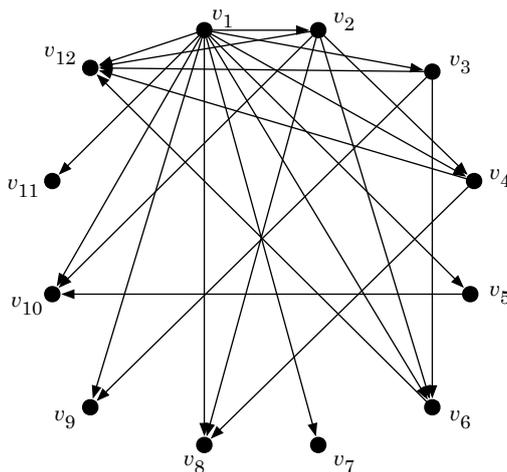
Si richiede di risolvere entrambi gli esercizi riportando una codifica in linguaggio C completa dei due programmi. Nel caso in cui non si riesca a completare entrambi gli esercizi si suggerisce di riportare almeno la codifica in C delle funzioni principali o una loro pseudo-codifica. È possibile consultare libri e appunti personali, ma non scambiare libri o appunti con altri studenti. 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  $n > 0$  costruire le liste di adiacenza del grafo  $G = (V, E)$ , con  $V = \{1, \dots, n\}$  ed  $E = \{(i, j) : i, j \in V, j = ki \text{ per } k > 0\}$ . Visualizzare in output le liste di adiacenza del grafo  $G$ . Disegnare sul foglio le liste di adiacenza del grafo rappresentato in figura.

**Esempio** Sia  $n = 12$ , allora il grafo  $G = (V, E)$  è quella rappresentato in figura.



## Soluzione

Le liste di adiacenza del grafo sono le seguenti:

```
v0 : null
v1 : 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12 → null
v2 : 4 → 6 → 8 → 10 → 12 → null
v3 : 6 → 9 → 12 → null
v4 : 8 → 12 → null
v5 : 10 → null
v6 : 12 → null
v7 : null
v8 : null
v9 : null
v10 : null
v11 : null
v12 : null
```

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 struct nodo {
6     int info;
7     struct nodo *next;
8 };
9
10 void aggiungiSpigolo(struct nodo *V[], int i, int j) {
11     struct nodo *p;
12     p = malloc(sizeof(struct nodo));
13     p->info = j;
14     p->next = V[i];
15     V[i] = p;
16     return;
17 }
18
19 void costruisciGrafo(struct nodo *V[], int n) {
20     int i, j;
21     for (i=1; i<=n; i++) {
22         V[i] = NULL;
23         j = 2*i;
24         while (j <= n) {
25             aggiungiSpigolo(V, i, j);
26             j = j+i;
27         }
28     }
29     return;
30 }
31
32 void stampaLista(struct nodo *p) {
33     while (p != NULL) {
34         printf("%d --> ", p->info);
35         p = p->next;
36     }
37     printf("Null\n");
```

```

38     return;
39 }
40
41 void stampaGrafo(struct nodo *V[], int n) {
42     for (int i=1; i<=n; i++) {
43         printf("%2d: ", i);
44         stampaLista(V[i]);
45     }
46     return;
47 }
48
49 int main(void) {
50     int n;
51     struct nodo *V[MAX];
52     printf("Inserisci il numero di vertici del grafo: ");
53     scanf("%d", &n);
54     costruisciGrafo(V, n);
55     stampaGrafo(V, n);
56     return 0;
57 }

```

## Esercizio n. 2

Letti in input due numeri interi  $n, k > 0$  costruire un array  $A$  con numeri interi casuali appartenenti all'insieme  $\{-k, \dots, 0, \dots, k\}$ . Costruire un secondo array  $B$  tale che  $B_i = \min\{A_i, A_{i+1}, \dots, A_{n-1}\}$ . Visualizzare in output gli array  $A$  e  $B$ .

**Esempio** Siano  $n = 10$  e  $k = 7$ . Consideriamo il seguente vettore di 10 numeri casuali scelti nell'insieme  $\{-7, \dots, 7\}$ :

$$A = (6, -5, 2, 7, 3, -3, 4, 6, -2, 1)$$

L'array  $B$  sarà quindi composto dai seguenti elementi:

$$B = (-5, -5, -3, -3, -3, -3, -2, -2, -2, 1)$$

## Soluzione

```

1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <time.h>
4  #define MAX 50
5
6  int min(int A[], int i, int n) {
7      int j, m = A[i];
8      for (j = i+1; j<n; j++)
9          if (A[j] < m)
10             m = A[j];
11     return m;
12 }
13
14 void costruisciArray(int A[], int B[], int n) {
15     int i;
16     for (i=0; i<n; i++)
17         B[i] = min(A, i, n);
18     return;
19 }
20

```

```

21 void arrayCasuale(int A[], int n, int k) {
22     srand((unsigned)time(NULL));
23     for (int i=0; i<n; i++)
24         A[i] = rand() % (2*k + 1) - k;
25     return;
26 }
27
28 void stampaArray(int A[], int n) {
29     for (int i=0; i<n; i++)
30         printf("%2d ", A[i]);
31     printf("\n");
32     return;
33 }
34
35 int main(void) {
36     int n, k, A[MAX], B[MAX];
37     printf("Inserisci due interi positivi n e k: ");
38     scanf("%d %d", &n, &k);
39     arrayCasuale(A, n, k);
40     costruisciArray(A, B, n);
41     stampaArray(A, n);
42     stampaArray(B, n);
43     return 0;
44 }

```