

## Esame scritto del 10 Febbraio 2017 (Appello B)

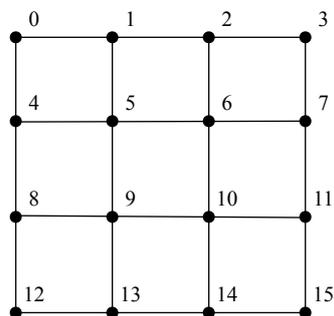
Si richiede di risolvere entrambi gli esercizi riportando una codifica in linguaggio C completa dei due programmi. Nel caso in cui non si riesca a completare entrambi gli esercizi si suggerisce di riportare almeno la codifica in C delle funzioni principali o una loro pseudo-codifica. È possibile consultare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

### Esercizio n. 1

Letto in input un intero  $n > 0$  costruire le liste di adiacenza del grafo non orientato  $G = (V, E)$ , con  $V = \{0, 1, 2, \dots, n^2 - 1\}$ , costituito da una griglia quadrata di  $n \times n$  vertici. Stampare le liste di adiacenza del grafo.

**Esempio** Sia  $n = 4$ . Il grafo  $G$  costituito da una griglia di  $4 \times 4$  vertici è rappresentato in figura.



### Soluzione

```

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 stampaLista(struct nodo *p) {
11     while (p!=NULL) {
12         printf("%d --> ", p->info);

```

```

13     p = p->next;
14 }
15 printf("NULL\n");
16 return;
17 }
18
19 void stampaGrafo(struct nodo *V[], int n) {
20     int i;
21     for (i=0; i<n; i++) {
22         printf("%d: ", i);
23         stampaLista(V[i]);
24     }
25     return;
26 }
27
28 void aggiungiSpigolo(struct nodo *V[], int u, int v) {
29     struct nodo *p;
30     p = malloc(sizeof(struct nodo));
31     p->info = v;
32     p->next = V[u];
33     V[u] = p;
34     return;
35 }
36
37 void grafoGriglia(struct nodo *V[], int n) {
38     int i;
39     for (i=0; i<n*n; i++) {
40         V[i] = NULL;
41         if (i % n < n-1)
42             aggiungiSpigolo(V, i, i+1);
43         if (i % n > 0)
44             aggiungiSpigolo(V, i, i-1);
45         if (i >= n)
46             aggiungiSpigolo(V, i, i-n);
47         if (i < n*(n-1))
48             aggiungiSpigolo(V, i, i+n);
49     }
50     return;
51 }
52
53 int main(void) {
54     int n;
55     struct nodo *V[MAX];
56     printf("Inserisci un intero positivo: ");
57     scanf("%d", &n);
58     grafoGriglia(V, n);
59     stampaGrafo(V, n*n);
60     return(0);
61 }

```

## Esercizio n. 2

Leggere in input una stringa  $S$  composta da singole cifre numeriche alternate con i simboli “+” e “-”. Calcolare e stampare il valore dell’espressione aritmetica rappresentata da  $S$ .

**Esempio** Sia  $S = “3 - 5 + 4 + 7 - 2”$ . Il valore dell’espressione rappresentato da  $S$  è 7. Sia  $S = “- 3 + 8 - 7 - 2”$ . Il valore dell’espressione rappresentato da  $S$  è -4.

### Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 int valuta(char *s) {
6     int i = 0, r = 0;
7     if (s[0] != '-' && s[0] != '+') {
8         r = s[0] - 48;
9         i = 1;
10    }
11    while (s[i] != '\0') {
12        if (s[i] == '+')
13            r = r + (s[i+1] - 48);
14        else
15            r = r - (s[i+1] - 48);
16        i = i+2;
17    }
18    return(r);
19 }
20
21 int main(void) {
22     char S[MAX];
23     int n;
24     printf("Inserisci l'espressione: ");
25     scanf("%s", S);
26     n = valuta(S);
27     printf("%s = %d\n", S, n);
28     return(0);
29 }
```