



## ***Laboratorio n.0***

### **Esercizio n. 1: Sequenze numeriche in vettore**

*Esercizio da 2 punti, parte di programmazione semplificata, appello del 13 settembre 2018*

Sia  $V$  un vettore di  $N$  interi ( $N \leq 30$ ). Si scriva un programma in C che, una volta acquisito da tastiera tale vettore, visualizzi tutti i sottovettori di dimensione massima formati da celle contigue contenenti dati non nulli.

#### ***Esempio:***

dato il vettore  $[1 \ 3 \ 4 \ 0 \ 1 \ 0 \ 9 \ 4 \ 2 \ 0]$ , i due sottovettori di dimensione massima (3) contenenti dati non nulli sono  $[1 \ 3 \ 4]$  e  $[9 \ 4 \ 2]$ .

### **Esercizio n. 2: Manipolazione di stringhe**

*Esercizio da 2 punti, parte di programmazione semplificata, appello del 21 giugno 2018*

Un file di testo contiene informazioni con il seguente formato:

- la prima riga del file contiene un intero  $N$  che indica il numero di parole
- ciascuna delle  $N$  righe successive contiene una parola per riga (massimo 20 caratteri).

Si scriva un programma C che conti, tra le parole del file, quante sono le sottostringhe di una data lunghezza con 2 vocali.

Il programma:

- legga i dati da un file di ingresso, il cui nome (massimo 20 caratteri) sia letto da tastiera
- legga da tastiera un intero  $n$  che rappresenta la lunghezza delle sottostringhe cercate
- per ogni parola acquisita chiami la funzione `conta` che conta quante sottostringhe di  $n$  caratteri contenenti esattamente due vocali appaiono nella stringa  $S$  passata come argomento
- al termine stampi il numero complessivo di sottostringhe trovate con esattamente due vocali.

Il prototipo della funzione sia:

```
int conta(char S[20], int n);
```

#### ***Esempio:***

se  $S = \text{"forExample"}$  e  $n=4$ , le sottostringhe di  $S$  di lunghezza 4 con 2 vocali sono 4 e sono `"forE"`, `"orEx"`, `"rExa"` e `"Exam"`.

### **Esercizio n. 3: Rotazione di vettori**

Si scriva una funzione C in grado di permettere all'utente di far ruotare verso destra o verso sinistra i contenuti di un vettore di  $N$  interi, di un numero a scelta di posizioni  $P$ . Il vettore è da intendersi come *circolare*, nel senso che l'elemento a destra della cella di indice  $N-1$  è la cella di indice 0 e l'elemento a sinistra della cella di indice 0 è la cella di indice  $N-1$ . La figura seguente illustra una rotazione a destra di 3 posizioni:





La funzione abbia il seguente prototipo:

```
void ruota(int v[maxN], int N, int P, int dir);
```

Il main:

1. acquisisca da tastiera  $N$  ( $N \leq \text{maxN}$  con  $\text{maxN}$  pari a 30)
2. acquisisca da tastiera il vettore  $V$
3. effettui ripetutamente delle rotazioni, acquisendo ciascuna volta  $P$  ( $P < N$ ,  $P=0$  per terminare) e la direzione ( $\text{dir} = -1$  per rotazione a destra,  $\text{dir} = 1$  per rotazione a sinistra) e stampi il vettore risultante.

#### **Esercizio n. 4: Iterazione su matrici**

Un file di testo contiene una matrice di interi con il seguente formato:

- la prima riga del file specifica le dimensioni della matrice (numero di righe  $n_r$  e numero di colonne  $n_c$ ). Si assuma che entrambi i valori siano comunque al più pari a 20
- ciascuna delle  $n_r$  righe successive contiene gli  $n_c$  valori corrispondenti a una riga della matrice, separati da uno o più spazi.

Si scriva un programma C che:

- legga tale matrice dal file di ingresso, il cui nome (massimo 20 caratteri) sia letto da tastiera
- chieda ripetutamente all'utente un valore  $\text{dim}$  compreso tra 1 e il minimo tra  $n_r$  e  $n_c$  e stampi tutte le sottomatrici quadrate di tale dimensione contenute nella matrice
- termini l'iterazione se l'utente inserisce un valore non coerente con le dimensioni della matrice
- memorizzi in un'opportuna matrice e stampi al termine la sottomatrice quadrata, tra quelle precedentemente individuate, la somma dei cui elementi è massima.