

TECNICHE DI PROGRAMMAZIONE

Esercitazione di Laboratorio 5

Obiettivi

- Risolvere problemi numerici iterativi, utilizzando vettori e matrici (*Dal problema al programma: Cap. 5.1 e 5.2*),

Contenuti tecnici

- Basi di Input Output
- Utilizzo di funzioni
- Costrutti condizionali e iterativi
- Manipolazioni elementari di vettori e matrici (di int e float)

Da risolvere durante il laboratorio oppure prima/dopo il laboratorio stesso

Esercizio 1.

Competenze: manipolazioni di vettori di numeri, costrutti iterativi

Categoria: problemi su sequenze di numeri (Dal problema al programma: 4.2) con sotto-problemi di verifica/selezione (3.4 e 4.5)

Sequenze numeriche in vettore

Sia V un vettore di N interi (con $N \leq 30$). Si scriva un programma in C che, una volta acquisito da tastiera (o da file, la scelta è libera) il contenuto del vettore, chiami una funzione avente prototipo

```
sottoSequenze(int V[], int N);
```

La funzione 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 $[9 \ 4 \ 2]$.

Suggerimenti:

Si noti che il problema può essere affrontato risolvendo due sottoproblemi:

- *identificare sottovettori di dati non nulli*
 - *è sufficiente “riconoscere” l’inizio e la fine di tali sottovettori*
- *selezionare quelli di lunghezza massima:*
 - *un metodo semplice consiste nel “provare” tutte le lunghezze in modo decrescente, fermandosi alla prima lunghezza per cui si trovano sottovettori;*
 - *un metodo più efficiente consiste nel determinare prima la lunghezza massima e poi cercare i sottovettori corrispondenti*
 - *si potrebbe anche (in alternativa) cercare di fare una sola iterazione sul vettore principale per riconoscere i sottovettori, determinare la lunghezza massima e “ricordare” (usando un altro vettore) gli inizi dei sotto-vettori: ma la complessità/efficienza non cambierebbe (oc-*

correrebbe comunque una ulteriore iterazione per stampare i sotto-vettori) e (probabilmente) il programma sarebbe più complicato che nella versione precedente.

Esercizio 2.

Competenze: manipolazioni di vettori di numeri, costrutti iterativi

Categoria: problemi su sequenze ordinate di numeri (Dal problema al programma: 4.2)

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 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 3.

Competenze: manipolazioni di matrici di numeri, costrutti iterativi

Categoria: problemi su matrici di numeri (Dal problema al programma: 4.1 e 4.2)

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 `dim` compreso tra 1 e il minimo tra `nr` e `nc` 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
- determini e stampi al termine la sottomatrice quadrata, tra quelle precedentemente individuate, la somma dei cui elementi è massima.

Se ad esempio il file contenesse la seguente matrice di 3 righe per 4 colonne

```
3 4
1 2 3 4
5 6 7 8
9 0 1 1
```

e `dim` valesse 2, occorrerebbe visualizzare:

Le sottomatrici quadrate di dimensione 2 sono:

```
1 2
5 6

2 3
6 7

3 4
7 8

5 6
9 0

6 7
0 1

7 8
1 1
```

La sottomatrice con somma degli elementi massima (22) è':

```
3 4
7 8
```