



Esercitazione di laboratorio n. 1

(Caricamento sul portale entro le 23.59 del 31/10/2016 dell'esercizio 1 e di almeno uno tra gli esercizi 2 e 3)

Esercizio n. 1: Espressioni semplificate in forma prefissa

Competenze: manipolazioni matematiche di dati scalari, manipolazione di stringhe, uso di funzioni

Categoria: problemi di elaborazione testi e problemi numerici non iterativi su dati scalari (*Dal problema al programma: 3.3,3.1*)

Una forma semplificata di espressioni intere è quella in cui uno dei 4 operatori aritmetici $+$ $-$ $*$ e $/$ viene applicato a 2 operandi. Queste espressioni semplificate possono essere in forma:

- infissa: `<operando1><operatore><operando2>`
- prefissa: `<operatore><operando1><operando2>`
- postfissa: `<operando1><operando2><operatore>`.

Si scriva un programma C che acquisisca ripetutamente da tastiera un carattere seguito da 2 stringhe (massimo 5 caratteri), separati da spazi. I dati in ingresso rappresentano un'espressione con numeri reali (tipo C float) semplificata in forma prefissa, dove il carattere è l'operatore ($'+' , '-' , '*' , '/'$) e le 2 stringhe gli operandi. Le stringhe contengono in alternativa:

- la rappresentazione di una costante compatibile con il tipo float (convertibile da stringa a float con `atof()` oppure input formattato con direttiva `"%f"`)
- la stringa "PREV", indicante l'utilizzo, come operando, del risultato della precedente operazione.

L'acquisizione avvenga in 2 passi:

- nel primo si legge l'operatore
- nel secondo si leggono e si convertono gli operandi in numeri di tipo float. A tale scopo si chiede di realizzare un'apposita funzione `operando()`, che, data una stringa, ritorni l'operando richiesto (conversione a float oppure risultato dell'operazione precedente).

NOTA: la funzione `operando()` deve corrispondere al prototipo
`float operando(char s[], float precedente);`

la funzione confronta `s` con "PREV", se il confronto ha esito affermativo, viene ritornato il valore di precedente, in caso contrario si ritorna il contenuto di `s` convertito mediante `atof()`.

Dopo ogni acquisizione, si visualizzi il risultato dell'operazione richiesta sugli operandi forniti, calcolato mediante apposita funzione che riceve come parametri gli operandi. Si termini quando l'operatore letto da tastiera non è uno dei 4 operatori leciti

Esempio:

Dati inseriti da tastiera:
+ 35 186
- PREV 121
!

Messaggio visualizzato:
Risultato: 221
Risultato: 100
Fine



Esercizio n. 2: manipolazione di matrici

Competenze: lettura/scrittura di file, manipolazioni di matrici, costrutti iterativi.

Classificazione: problemi di selezione iterativi (*Dal problema al programma: 3.4.2, 4.5.2*)

Un file di testo contiene una matrice $N \times N$ di interi con il seguente formato:

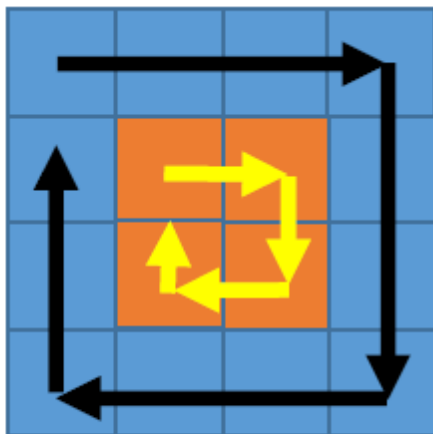
- N compare sulla prima riga del file ($N \leq 20$)
- ciascuna delle N righe successive contiene gli N valori corrispondenti a una riga della matrice, separati da uno o più spazi.

Si scriva un programma C che:

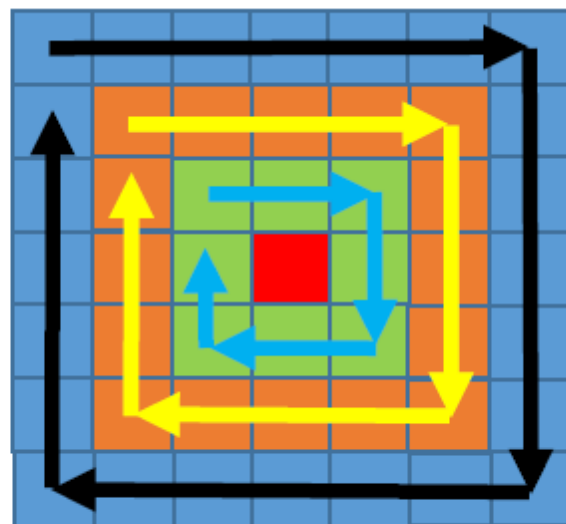
- legga da tastiera il nome del file di ingresso (massimo 10 caratteri)
- legga la matrice dal file di ingresso
- visualizzi i contenuti della matrice “a spirale” (in senso orario), partendo dall'angolo in alto a sinistra seguendo cornici via via più strette fino a stampare gli elementi più interni.

Suggerimenti:

- determinare il numero di cornici (iterazioni) in funzione di N
- ad ogni iterazione selezionare una cornice e percorrerla con 4 movimenti: lungo la riga da SX a DX, lungo la colonna all'alto in basso, lungo la riga da DX a SX, lungo la colonna dal basso all'alto
- tener conto dell'eventuale cella centrale.



$N = 4$



$N = 7$

Esempio:

Contenuto del file di ingresso:

3

1 2 3

4 5 6

7 8 9



Messaggio in uscita:

1 2 3 6 9 8 7 4 5

Esercizio n. 3: Integrazione numerica mediante metodo dei rettangoli

Competenze: uso di funzioni, passaggio di parametri a funzioni, lettura/scrittura di file.

Classificazione: problemi numerici iterativi (*Dal problema al programma: 3.1, 4.1*)

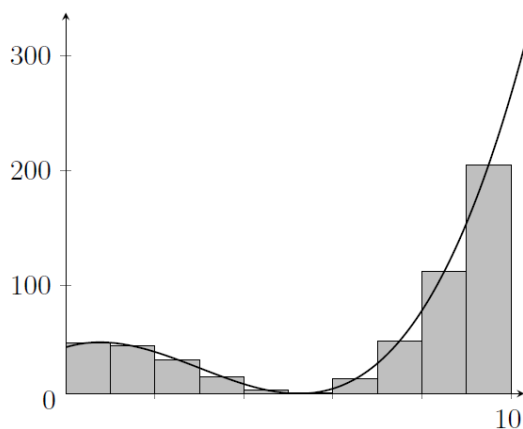
Dato un polinomio $p(x)$ di grado n e gli estremi di integrazione a e b , il suo integrale definito

$$\int_a^b p(x) dx$$

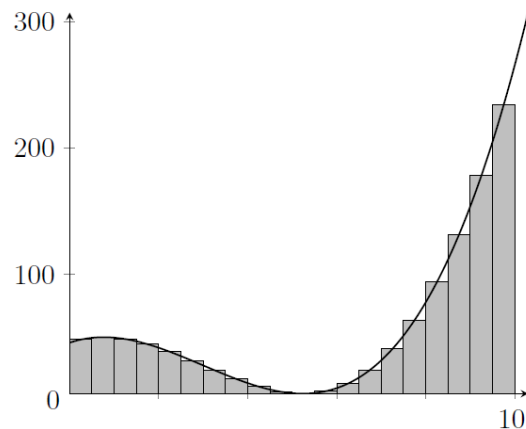
può essere calcolato approssimativamente mediante la regola dei rettangoli. L'intervallo di integrazione $[a, b]$ viene suddiviso in M sottointervalli di ampiezza uniforme $h = \frac{b-a}{M}$. Detto \hat{x}_k il punto medio del k -esimo intervallo ($0 \leq k < M$), vale:

$$\int_a^b p(x) dx \approx h \sum_{k=0}^{M-1} p(\hat{x}_k)$$

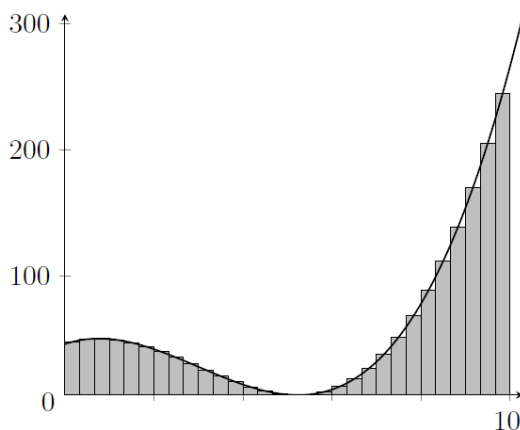
In figura è riportato un esempio per la funzione $p(x) = x^3 - 9x^2 + 12x + 46$ la cui area sottesa nell'intervallo $[0, 10]$ vale 560, considerando quattro diversi gradi di approssimazione ($M = 10, 20, 30$ o 40).



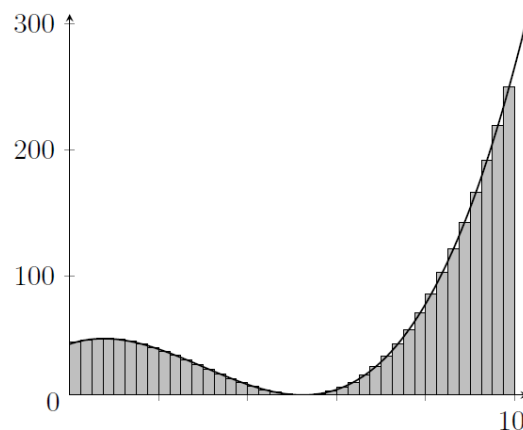
(a) $N=10$, $I=555$



(b) $N=20$, $I=558.75$



(c) $N=30$, $I=559.44$



(d) $N=40$, $I=559.69$



Si scriva un programma C che:

- acquisisca da tastiera il grado n ($n \leq 10$) del polinomio $p(x)$ e i suoi $n + 1$ coefficienti reali C_i .
- acquisisca da tastiera gli estremi di integrazione (interi) a e b
- acquisisca da tastiera il numero M di sottointervalli
- calcoli e visualizzi il valore di $\int_a^b p(x)dx$.

Si noti l'uso dell'eventuale vettore introdotto come contenitore di dati, con corrispondenza indice dato (*Dal problema al programma: 4.1*)

Dato un polinomio $p(x) = c_n x^n + c_{n-1} x^{n-1} + \dots + c_2 x^2 + c_1 x + c_0$, si realizzi una funzione, di prototipo

```
float valutaHorner(float P[11], float x);
```

che lo valuta per un valore di x dato mediante il metodo di Horner, che evita il calcolo (costoso) delle potenze di x :

$$p(x) = c_n x^n + c_{n-1} x^{n-1} + \dots + c_2 x^2 + c_1 x + c_0 = (\dots((c_n x + c_{n-1})x + c_{n-2}) \dots)x + c_0$$

Esempio: il polinomio $p(x) = c_4 x^4 + c_3 x^3 + c_2 x^2 + c_1 x + c_0$ viene espresso e poi valutato come:

$$p(x) = (((((c_4 x + c_3)x + c_2)x + c_1)x + c_0)$$