LABORATORIO DI PROGRAMMAZIONE
CORSO DI LAUREA IN SICUREZZA DEI SISTEMI E DELLE RETI INFORMATICHE
UNIVERSITÀ DEGLI STUDI DI MILANO
2025–2026

Indice

Tratti dagli esercizi del corso del prof. Vincenzo Marra.
Ultima revisione: 8 novembre 2025.

Parte 1. **Puntatori: primi esercizi**

Esercizio 1

Scambio di interi.

Tempo: 5 min.

Scrivete un programma che dichiari due puntatori di tipo `int` di nome `px` e `py`, assegni al contenuto delle locazioni di memoria puntate i valori `0` e `1`, rispettivamente, e poi scambi i valori in modo che `px` punti al valore precedentemente puntato da `py` e viceversa. Per eseguire lo scambio usate un terzo puntatore a `int` ausiliario. Eseguite le stampe appropriate per verificare che lo scambio di valori sia avvenuto correttamente.

Esercizio 2

Scambio di stringhe.

Tempo: 5 min.

Ripetete l'esercizio precedente usando puntatori a `char` (inizializzati in modo da puntare a stringhe di caratteri) in luogo di puntatori a `int`. Dichiarate le stringhe di caratteri direttamente nel codice, senza leggerle da tastiera. Per esempio, dichiarate e inizializzate una stringa in questo modo:

```
char s[] = "Galileo Galilei";
```

Esercizio 3

Banca con menu (versione con puntatori).

Tempo: 20 min.

Scrivete un programma che simula le operazioni di una banca. All'avvio, il programma chiede all'utente di inserire il saldo iniziale del conto corrente, di tipo `double`. Il saldo può essere negativo. In seguito, il programma visualizza il menu seguente.

```
1. Visualizza saldo.
2. Deposito.
3. Prelievo.
4. Calcolo interesse.
5. Esci.
>
```

dove `>` è un "prompt" che indica all'utente che la macchina è in attesa dell'input dell'utente. L'utente inserisce una scelta. Se la scelta è inesistente o sbagliata, il programma stampa un messaggio d'errore, e visualizza nuovamente il menu. Se la scelta è `5`, il programma termina. Se la scelta è `1`, il programma visualizza il saldo corrente. Se la scelta è `2`, il programma chiede all'utente quanto denaro intende depositare, il quale è di tipo `double` e > 0. Il programma aggiorna quindi il saldo del conto e lo mostra all'utente. Se il denaro da depositare è ≤ 0, il programma visualizza un messaggio d'errore e mostra nuovamente il menu. Se la

scelta è `3`, il programma chiede all'utente quanto denaro intende prelevare, il quale è di tipo `double` e > 0 e $\leq$ del saldo corrente. Il programma aggiorna quindi il saldo del conto e lo mostra all'utente. Se l'utente vuole prelevare più di quanto ha sul conto, o inserisce un numero ≤ 0, il programma visualizza un messaggio d'errore appropriato, e mostra nuovamente il menu. Se la scelta è `4`, il programma chiede all'utente di inserire per quanti anni prevede di lasciare il proprio denaro sul conto, e calcola gli interessi che maturerebbero assumendo un tasso fisso noto a priori. Il numero di anni dev'essere di tipo `int` e > 0. Il programma mostra quindi il saldo e l'interesse ipotetico. Questo calcolo non è possibile se l'utente ha saldo negativo, e dev'esserne informato. Dopo aver completato ogni operazione secondo le alternative `1`–`4`, con successo oppure no, il programma mostra nuovamente il menu. Implementate il programma usando delle funzioni il cui tipo di ritorno è sempre `void`.

L'interesse indica il guadagno ottenuto lasciando il proprio denaro sul conto corrente.

Non dovete usare variabili con scope globale. Usate i puntatori per operare sul saldo.

Interesse

Assumendo che il proprio saldo sia s e che si intenda lasciare il proprio denaro sul conto per y anni con un tasso t, l'interesse i si calcola come

$$i = s \times t \times y \tag{1}$$

Di conseguenza, lasciando il proprio denaro sul conto si avrebbe un saldo di $s + i$.

Naturalmente, il calcolo descritto è una semplificazione.

Esercizio 4

Statistiche su sequenze di numeri.

Tempo: 20 min.

Scrivete un programma che legge in input una sequenza di numeri (di tipo `double`) la cui lunghezza non è nota a priori. Ogni volta che l'utente inserisce un nuovo numero, il programma calcola, e mostra, la somma dei numeri inseriti fino a quel momento, il valore massimo, e il valore minimo. L'inserimento termina quando l'utente inserisce un numero speciale a vostra scelta. Definita una funzione `calcola_stats`, il cui tipo di ritorno è `void`, che aggiorna i tre indicatori (somma, massimo, e minimo), richiamata dalla funzione `main`. Di seguito, un esempio di esecuzione, in cui `3.14` è il numero speciale che termina l'esecuzione. Il testo che segue `>` è l'input dell'utente, il resto è l'output del programma.

Non dovete usare array.

```
Inserisci un numero:
> 5
Somma: 5.0, min: 5.0, max: 5.0
Inserisci un numero:
> 10
Somma: 15.0, min: 5.0, max: 10.0
Inserisci un numero:
> 2.5
Somma 17.5, min: 2.5, max: 10.0
> 3.14
```

Dipartimento di Informatica, Università degli Studi di Milano,