

# Prova finale di algoritmi e strutture dati

# Obiettivi didattici e realizzazione

- Obiettivi
  - Applicazione pratica delle tecniche apprese nel modulo di algoritmi e strutture dati del corso di algoritmi e principi dell'informatica
  - Implementazione di una soluzione ad un problema prestando attenzione ad aspetti concreti di efficienza del codice
- Realizzazione
  - Linguaggio C (C11, VLA ammessi)
  - Nessuna libreria esterna al di là della libreria standard C
  - No multithreading
  - Dati in ingresso ricevuti via stdin, risultati da fornire via stdout

# Modalità di realizzazione

- Il progetto è **strettamente individuale**
  - Non utilizzate **alcun frammento di codice altrui**
- Siete responsabili del vostro codice
  - Non caricatelo su repository pubblici
  - Non condividetelo con colleghi per "prendere ispirazione"
  - Non utilizzate alcun frammento di codice reperito su Internet
  - Non utilizzate assistenti automatici a-la Copilot
- In caso di plagi o uso di codice altrui, **tutti i progetti coinvolti** saranno annullati

# Criteri di valutazione

- Correttezza ed efficienza della soluzione proposta sono valutate con batterie di test automatizzate
- Verranno forniti input/output d'esempio per poter collaudare la soluzione in locale
  - Non sottoponete soluzioni senza aver verificato che funzionino localmente
  - Verrà fornito anche uno strumento di generazione automatica di casi di test (input/output), per facilitarvi il testing in locale
- Il sistema di verifica calcola il tempo macchina e la memoria utilizzati
- La valutazione è immediatamente calcolata (e subito visibile), mediante 6 batterie di test (task, nel lessico del verificatore):
  - Ogni batteria ha una valutazione associata tra queste: {18,21,24,27,30,30 e lode}
  - Per ottenere una valutazione X è necessario superare la batteria di test *con valutazione associata X*

# Criteri di valutazione

- Nessun limite al numero di sottoposizioni, né penalità per sottoposizioni multiple
- È possibile migliorare la valutazione quante volte si desidera
- **Avvertenza:** viene valutata l'*ultima* sottoposizione fatta.
  - *Sottoponete sempre il vostro sorgente definitivo al test in cui ottenete la valutazione più alta*
- Verificatore disponibile all'indirizzo <https://dum-e.deib.polimi.it>
- Credenziali di accesso ricevute via mail istituzionale polimi
- Invio delle credenziali e apertura verificatore entro il 7 giugno

# Scadenze e pianificazione

- Per i laureandi di luglio
  - **3 luglio**, ore 23.59 CEST. Segnalate (email al docente) la necessità di valutazione
- Per tutti gli altri
  - **4 settembre**, ore 23.59 CEST, dopo di che la piattaforma verrà **chiusa**
- Per laureandi di gennaio/febbraio (SUPERATO 145 CFU (di qualsiasi genere) + essere iscritto all'esame di laurea)
  - la piattaforma sarà riaperta per 10 giorni nella sessione d'esame invernale (**gennaio/febbraio 2025**). Le date precise saranno comunicate in seguito.
- Iniziare a lavorare ad una settimana dalla scadenza è uno dei modi migliori per **non** riuscire a superare la prova

# Tutoraggio

- Sezione Barenghi:
  - Giuseppe Boccia [giuseppe.boccia@mail.polimi.it](mailto:giuseppe.boccia@mail.polimi.it)
  - Matteo Cenzato [matteo.cenzato@mail.polimi.it](mailto:matteo.cenzato@mail.polimi.it)
- Sezione Martinenghi:
  - Cristian Lo Muto [cristian.lomuto@mail.polimi.it](mailto:cristian.lomuto@mail.polimi.it)
  - Giorgio Miani [giorgio.miani@mail.polimi.it](mailto:giorgio.miani@mail.polimi.it)
- Sezione Pradella:
  - Giorgio Pristia [giorgio.pristia@mail.polimi.it](mailto:giorgio.pristia@mail.polimi.it)
  - Federico Toschi [federico.toschi@mail.polimi.it](mailto:federico.toschi@mail.polimi.it)
- Canale Telegram gestito dai tutor: <https://t.me/+oaOqymJIMP0zMmU0>
- Incontro con descrizione degli strumenti di sviluppo: venerdì 31 maggio, 15:15 in aula 2.0.1 (l'incontro è accessibile alla pagina webex del prof. Barenghi; sarà registrato)

# La pasticceria

- Obiettivo: implementare il software di gestione di una pasticceria
  - Gestione degli ordini
  - Gestione delle scorte
  - Gestione del ricettario
- Premesse:
  - La pasticceria funziona a tempo discreto, istante iniziale 0
  - I macchinari di produzione della pasticceria sono in grado di preparare i dolci istantaneamente



# Ingredienti, ricette e forniture

- *Ingredienti*: sono identificati da un nome (sequenza di caratteri)
- *Ricette*:
  - Sono identificate da un nome
  - Utilizzano diversi *ingredienti*, in diverse quantità (intere, positive)
- *Magazzino*: contiene *lotti* di ingredienti. Ogni *lotto* è caratterizzato da:
  - *nome* dell'ingrediente
  - *quantità*: intero, positivo
  - *data di scadenza*: intero, indica l'istante di tempo a partire dal quale il lotto è scaduto

# Ordini e ritiro

- *Ordini*: viene ordinata una *quantità* di un dolce con ricetta
  - Gli ordini sono smaltiti in ordine cronologico di *arrivo dell'ordine*
  - Nella preparazione, si usano gli ingredienti più vicini alla scadenza
  - Se gli ingredienti non sono disponibili, l'ordine è messo in *attesa*
  - Ad ogni rifornimento (arrivo di un lotto) la pasticceria valuta quali ordini in attesa possono ora essere evasi
- *Corriere*: periodicamente, un corriere passa a ritirare gli ordini evasi
  - Il periodo di passaggio del corriere è fissato, e il furgone ha una capienza (espressa come peso massimo) limitata
  - Vengono scelti gli ordini da spedire in base a quanto sono vecchi, indipendentemente dal peso. Al primo ordine che non ci sta, ci si ferma.
  - Gli ordini scelti vengono caricati in ordine di peso decrescente; a parità di peso gli ordini sono caricati in ordine cronologico di arrivo
  - Non è possibile caricare parzialmente un ordine sul furgone

# Comandi e risposte attese

- `aggiungi_ricetta nome_ricetta nome_ingrediente quantità ...`
  - Aggiunge una ricetta al ricettario, il numero di coppie `nome_ingrediente quantità` è arbitrario. Se la ricetta è già presente nel ricettario, il comando viene ignorato.
  - Output atteso: aggiunta / ignorato
- `rimuovi_ricetta nome_ricetta`
  - Rimuove una ricetta dal catalogo, se presente e non sono presenti ordini relativi ad essa ancora non spediti.
  - Output atteso: rimossa/ordini in sospeso/non presente
- `rifornimento nome_ingrediente quantità scadenza ...`
  - Rifornisce la pasticceria con un insieme di lotti, uno per ingrediente. Il numero di lotti per rifornimento è arbitrario.
  - Output atteso: rifornito
- `ordine nome_ricetta numero_elementi_ordinati`
  - Effettua un ordine di `numero_elementi_ordinati` dolci secondo la ricetta `nome_ricetta`. L'ordine viene rifiutato se non esiste una ricetta con nome specificato.
  - Output atteso: accettato/rifiutato

# Esempio di ingresso e risposte attese

Su stdin

 t: -  
Testo in ingresso: 5 325  
Risposta attesa:  
Commento: Configurazione corriere: tempo di passaggio e capienza

t: 0  
Testo in ingresso: aggiungi\_ricetta torta farina 50 uova 10 zucchero 20  
Risposta attesa: aggiunta  
Commento: Viene aggiunta la ricetta .

t: 1  
Testo in ingresso: aggiungi\_ricetta ciambella farina 20 uova 5 burro 2  
Risposta attesa: aggiunta

t: 2  
Testo in ingresso: aggiungi\_ricetta profiterole farina 10 uova 2 latte 3 zucchero 3 cioccolato 4  
Risposta attesa: aggiunta

# Esempio di ingresso e risposte attese

t: 3

Testo in ingresso: rimuovi\_ricetta sfogliatella

Risposta attesa: non presente

Commento: La ricetta non esiste

t: 4

Testo in ingresso: rifornimento farina 100 10 uova 100 10 zucchero 100 10 burro 100 10 latte 100 10 cioccolato 100 10

Risposta attesa: rifornito

Commento: I lotti del rifornimento sono aggiunti al magazzino

t: 5

Testo in ingresso: ordine ciambella 6

Risposta attesa: camioncino vuoto

accettato

Commento: Il corriere passa a t=5, viene gestito prima della lettura da stdin  
l'ordine è messo in attesa per mancanza di farina

t: 6

Testo in ingresso: ordine profiterole 3

Risposta attesa: accettato

Commento: Ordine preparato e subito messo in coda per la spedizione

# Esempio di ingresso e risposte attese

t: 7

Testo in ingresso: rimuovi\_ricetta profiterole

Risposta attesa: ordini in sospeso

Commento: I profiterole sono in coda per la spedizione

t: 8

Testo in ingresso: aggiungi\_ricetta pane\_dolce farina 1 zucchero 1 uova 1

Risposta attesa: aggiunta

Commento:

.

t: 9

Testo in ingresso: ordine ciambella 3

Risposta attesa: accettato

Commento: Ordine preparato e subito messo in coda per la spedizione

t: 10

Testo in ingresso: ordine torta 1

Risposta attesa: 9 ciambella 3

6 profiterole 3

accettato

Commento: L'ordine delle ciambelle è più pesante (81g vs. 66g), viene caricato per primo

# Esempio (cont.)

t: 11

Testo in ingresso: rifornimento farina 100 15 farina 50 13 uova 45 20 zucchero 20 20 burro 15 20

Risposta attesa: rifornito

Commento: L'ordine delle ciambelle al tempo 5 viene preparato perché è in attesa da più tempo, la torta ordinata al tempo 10 resta in attesa

t: 12

Testo in ingresso: rifornimento farina 100 15 uova 7 15 zucchero 25 15 latte 5 15 cioccolato 5 15

Risposta attesa: rifornito

Commento: Questo terzo rifornimento sblocca l'ordine della torta

t: 13

Testo in ingresso: ordine torta 1

Risposta attesa: accettato

Commento:

t: 14

Testo in ingresso: ordine profiterole 1

Risposta attesa: accettato

Commento: Torta e profiterole vengono subito preparati

# Esempio (cont.)

t: 15

Testo in ingresso:

Risposta attesa: 5 ciambella 6  
10 torta 1  
13 torta 1

Commento: I profiterole restano in coda per la spedizione perché non ci stanno sul camioncino e sono l'ordine più leggero. Le due torte hanno lo stesso peso, quindi vengono caricate in ordine cronologico. Quello che resta dei lotti con scadenza a tempo 15 viene tolto dalla dispensa.

A fine esecuzione rimangono inutilizzati 3g di burro e 2g di zucchero. L'ordine dei profiterole al tempo 14 non viene mai spedito.