

## Compito di Programmazione 20 Dicembre 2021

**Nota Bene.** Consegnare solamente la bella copia possibilmente su foglio bianco senza quadretti o righe. Ogni esercizio deve essere svolto su una pagina diversa. Alla fine del compito, usando una applicazione che faccia la scansione, tipo CamScanner, fare una foto a tutto il compito col cellulare e inviare le scansioni in pdf per email a [cosimo.laneve@unibo.it](mailto:cosimo.laneve@unibo.it), [giuseppe.lisanti@unibo.it](mailto:giuseppe.lisanti@unibo.it), [adele.veschetti2@unibo.it](mailto:adele.veschetti2@unibo.it).

1. (**punti 6**) Scrivere una funzione **ricorsiva** `there_are_primes` che prende in input un array di interi e ritorna vero se l'array contiene soltanto numeri primi, falso altrimenti. È possibile usare funzioni ausiliarie, purché ricorsive.
2. (**punti 9**) Un ristorante utilizza una lista di tavoli per memorizzare i tavoli liberi per la giornata. Ogni elemento della lista memorizza il numero di persone che il tavolo può ospitare, l'orario dal quale è disponibile (l'orario è rappresentato come un double es. 19.45, 20.30, etc) e un booleano che indica se il tavolo è all'interno o all'esterno (**true**: interno, **false**: esterno). Definire le strutture dati necessarie a rappresentare la lista di tavoli e le seguenti funzioni:
  - `prenota_tavolo` che prende come parametri il numero di persone per le quali si vuole prenotare e un orario. La funzione cancella dalla lista il tavolo che si prenota, se disponibile.
  - `quale_orario` che prende come parametro il numero di persone per le quali si desidera un tavolo e l'opzione dentro/fuori. La funzione ritorna il primo orario disponibile.
  - `quanti_tavoli` che prende come parametro un booleano e ritorna il numero di tavoli disponibili dentro al ristorante (se il booleano è true) o all'esterno del ristorante (se il booleano è false).

**NB.** Se non è disponibile un tavolo per il numero **esatto** delle persone per cui si vuole prenotare, le funzioni considerano il primo tavolo con un numero **sufficiente** di posti disponibili.

3. (**punti 9**) Un magazzino è caratterizzato da un insieme di tipologie di prodotti che può ospitare e dal numero di queste tipologie di prodotti. Ciascun prodotto è caratterizzato da un codice (un intero), un nome e la quantità di tale prodotto disponibile nel magazzino. Si rappresenti il singolo prodotto usando una struttura dati.

Si implementi la classe `Magazzino`, il relativo costruttore e il seguente metodo:

- `decrementa_disponibilita`, il quale riceve in input il codice del prodotto e la quantità di prodotto da decrementare, controlla se è disponibile la quantità richiesta, in caso positivo decrementa e restituisce l'indice del prodotto nell'insieme, altrimenti restituisce -1.

Si implementi la classe `MagazzinoPlus` la quale è caratterizzata anche da una soglia e da un intero che memorizza il conteggio delle tipologie di prodotti che devono essere approvvigionati. La soglia è uguale per tutte le tipologie di prodotto. Sfruttare l'ereditarietà e modificare il metodo `decrementa_disponibilita`, in modo che controlli se il prodotto per cui è stata decrementata la quantità è sceso sotto la soglia di approvvigionamento. In caso positivo incrementa il conteggio delle tipologie di prodotti da approvvigionare. Si implementi il costruttore e i seguenti metodi:

- `da_approvvigionare`, il quale riceve come input un array di interi nel quale memorizza i codici dei prodotti da approvvigionare.
- `aggiorna_disponibilita`, il quale riceve come input due array di interi contenenti il codice e la quantità di prodotto con cui aggiornare l'approvvigionamento dell'elenco di prodotti della classe. Questo metodo ha inoltre un terzo parametro che definisce la lunghezza dei due array appena descritti.