

Istruzioni

- Tempo disponibile: 90 minuti.
- Non è permesso l'uso di dispositivi elettronici (a parte il PC della propria postazione).
- Visual Studio Code è installato su tutte le postazioni. Si può comunque usare l'editor che si preferisce fra quelli installati.

Esercizio 1 (max. 9 punti)

Scrivere una funzione di nome `f` che restituisca il numero primo più grande che sia minore o uguale al suo parametro intero. Per verificare se un determinato numero è primo, `f` deve chiamare la funzione `primo` riportata nell'allegato `primo.c`.

Utilizzare la funzione `f` per scrivere un programma che

1. richieda all'utente un numero intero M ;
2. per ogni intero n tale che $2 \leq n \leq M$, stampi una riga contenente
 - n ;
 - il più grande numero primo minore o uguale a n ;
 - il numero di numeri primi minori o uguali a n .

Ad esempio, se $M = 8$, un output corretto è il seguente:

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

Per consegnare, caricare un file di nome `Esercizio1.c` con tutto il codice richiesto. L'elaborato sarà valutato per

- Identificazione delle strutture dati e degli algoritmi appropriati alle specifiche
- Corretta implementazione di strutture dati e algoritmi
- Strutturazione del programma in funzioni come richiesto dalle istruzioni.
- Stile (chiarezza, utilizzo di costrutti appropriati).
- Utilizzo efficiente delle risorse

I programmi non compilabili saranno valutati 0 punti. L'utilizzo di costrutti estranei al programma della prova (in particolare array) sarà penalizzato.

Esercizio 2 - max. 2 punti

Nell programma allegato [d3.c](#), riportato in figura 1, qual è il primo valore assunto da `i` che sia un divisore del numero di matricola del/la candidato/a, o del numero `44122` se il/la candidato/a non è immatricolato/a?

Per rispondere, caricare un file di nome [Esercizio2.txt](#) contenente solo il valore richiesto.

Valutazione:

- 2 punti per risposta corretta;
- 0 punti per risposta non data (file [Esercizio2.txt](#) non caricato)
- -1 punto per risposta errata.

```
int main(void) {  
    int i = 10000;  
    while (i > 0)  
        if (i % 3)  
            i /= 3;  
        else  
            i--;  
    return 0;  
}
```

Figura 1: Codice a cui si riferisce l'esercizio 2.