

## 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)

L'algoritmo in figura 1 calcola il massimo comun divisore dei suoi due parametri naturali, cioè il più grande intero divisore di entrambi. Nella figura, IN significa input, OUT output e  $\leftarrow$  assegnamento.

Scrivere una funzione di nome `MCD` che restituisca il massimo comun divisore fra i suoi due parametri interi, usando l'algoritmo in figura 1 e senza usare l'istruzione `goto`.

Utilizzare la funzione `MCD` in un programma che

1. definisca un array di 10 interi;
2. richieda all'utente un numero  $M$ ;
3. per  $i = 1, \dots, 10$ , assegni all'elemento  $i$ -esimo dell'array il massimo comun divisore di  $M$  e  $i$ ;
4. per  $i = 1, \dots, 10$ , stampi su output una riga di formato `MCD(M, i) = n`, dove  $n$  è l' $i$ -esimo elemento dell'array.

Ad esempio, se l'input fosse 12, un output corretto sarebbe

MCD(12, 1) = 1  
 MCD(12, 2) = 2  
 MCD(12, 3) = 3  
 MCD(12, 4) = 4  
 MCD(12, 5) = 1  
 MCD(12, 6) = 6  
 MCD(12, 7) = 1  
 MCD(12, 8) = 4  
 MCD(12, 9) = 3  
 MCD(12, 10) = 2

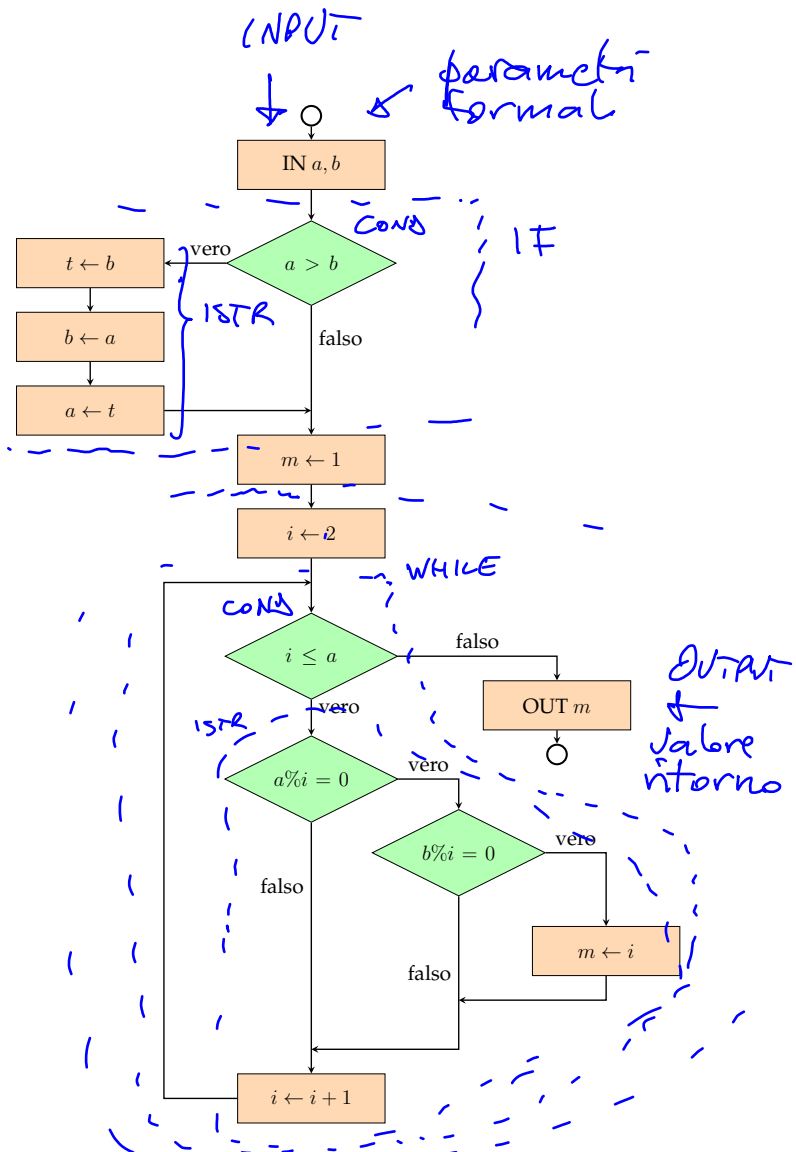


Figura 1: Algoritmo per il calcolo del massimo comun divisore di due numeri

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 sarà penalizzato.

## Esercizio 2 - max. 2 punti

Il programma nell'allegato [d1.c](#), riportato in figura 2, stampa il numero di numeri primi minori di 1.000.000.

23273

Usarlo (senza modificarlo) per calcolare il più piccolo numero primo maggiore del proprio del numero di matricola (o del numero 44122 se non si è immatricolati).

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.

```
#include <stdio.h>

int primo(int n) {
    int i, potenzialmente_primo;
    potenzialmente_primo = n % 2 == 2;
    i = 3;
    while (potenzialmente_primo && i * i <= n) {
        if (n % i == 0)
            potenzialmente_primo = 0;
        i += 2;
    }
    return potenzialmente_primo;
}

int main(void) {
    int i, cont = 0;
    for (i = 2; i < 1000000; i++)
        if (primo(i))
            cont++;
    printf("%d\n", cont);
    return 0;
}
```

Figura 2: Codice a cui si riferisce la domanda 2.