

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 ordina in senso crescente l'array di dimensione DL che riceve come input.

Scrivere un programma che

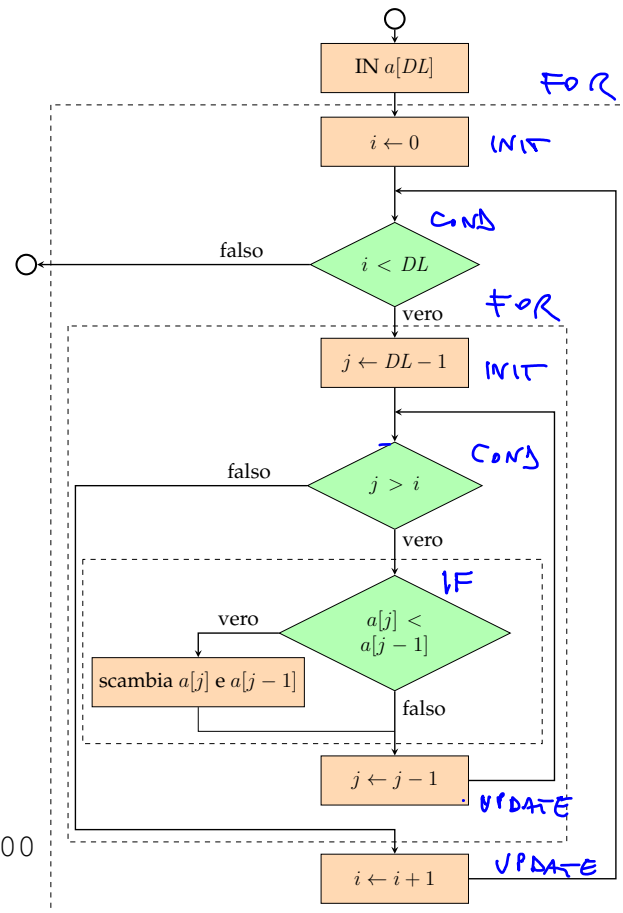
1. richieda all'utente un numero massimo di 50 numeri reali positivi e li memorizzi in un array, fermandosi quando l'utente inserisce un numero negativo;
2. ordini l'array utilizzando fedelmente l'algoritmo in Figura 1;
3. stampi l'array ordinato.

Ad esempio, se l'input fosse

1.3 2.5 0.7 5.0 4.1 2.5 0.2 -1

un output corretto sarebbe il seguente:

0.20 0.70 1.30 2.50 2.50 4.10 5.00



Suggerimento In figura 1, ognuno dei tre rettangoli tratteggiati racchiude una parte dell'algoritmo che può essere implementata con una istruzione di controllo di flusso.

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.

Figura 1: Algoritmo di ordinamento di un array di dimensione DL

Esercizio 2 - max. 2 punti

Nell programma allegato [d3.c](#), riportato in figura 2, qual è il numero massimo di record di attivazione presenti contemporaneamente sullo stack, se si dà come input l'ultima cifra del proprio numero di matricola, o il numero 2 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 h(int n) {
    return 1;
}

int g(int n) {
    if (n < 5) return 1;
    else return h(n - 2);
}

int f(int n) {
    if (n < 5) return 1;
    else return g(n - 2);
}

int main(void) {
    int m;
    scanf("%d", &m);
    printf("%d\n", f(m));
    return 0;
}
```

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