

INFORMAZIONI SULL'ORGANIZZAZIONE

- ORE 8:45 – 10:50 – AULA 1 - Studenti con matricola PARI e studenti DSA
- ORE 11:00 – 13:15 – AULA 1 - Studenti con matricola DISPARI
- I primi 15 minuti sono riservati alla compilazione online delle OPINIONI STUDENTI sui 2 moduli del corso (modulo 1 prof. Giunta, modulo 2 prof. Ciaramella)
- La Prova inizia dopo la compilazione delle Opinioni Studenti; durante la Prova Intercorso non è consentito l'uso di pc, tablet, cellulari, cuffie. La Prova dura 2 ore.
- La prova richiede solo l'uso di fogli di carta (forniti dal CdS) e di una penna (personale DA PORTARE DA CASA)
- **La Prova Intercorso consiste nello sviluppo di 2 algoritmi, nella loro codifica in C e nella valutazione della loro complessità di tempo.**
- Gli studenti che vogliono sostenere la Prova Intercorso devono prenotarsi su Esse3. Gli studenti in trasferimento che non hanno ancora accesso a Esse3 devono mandare una mail di prenotazione al prof. Ciaramella (angelo.ciaramella@uniparthenope.it).

Suggerimenti per la Prova intercorso:

la studentessa/lo studente deve essere in grado di scrivere ed analizzare le seguenti function C sviluppate a lezione:

scambiare, somma_array, media_array, appartiene, fusione, minimo_array, massimo_array, min_val_ind, max_val_ind, min_ind, max_ind, best_matching, string_matching

Esercitazione di preparazione per la Prova Intercorso del 24/11/2023

1. Scrivere una function C che, data come parametro di input una stringa **testo** che rappresenta un testo in italiano, determina e restituisce come parametro di output la media delle lunghezze delle parole contenute nella stringa **testo**. Nel testo le parole sono separate da uno o più 'spazi'. Valutare e commentare la complessità di tempo della function (operazione dominante: somma).
2. Scrivere una function C che, data come parametro di input una stringa **testo** che rappresenta un testo in italiano, determina e restituisce come parametro di output il massimo numero di vocali contenute in una parola del testo. Nel testo le parole sono separate da un unico 'spazio'. Scrivere e usare una function logica **e_vocale** che stabilisce se un carattere è una vocale. Valutare e commentare la complessità di tempo della function (operazione dominante: confronto tra char).
3. Scrivere una function C che ha tre parametri di input: una stringa **testo** che rappresenta un testo in italiano, e poi una stringa **chiave** e una stringa **subs** (**chiave** e **subs** hanno la stessa lunghezza). La function deve restituire come parametro di output la stringa **testo** modificata nel seguente modo: ogni occorrenza della stringa **chiave** come sottostringa di **testo** è sostituita dalla stringa **subs**. Nel testo le parole sono separate da uno o più 'spazi'. Valutare e commentare la complessità di tempo della function (operazione dominante: confronto tra char).
4. Scrivere una function C che ha tre parametri di input: un array di float **vettore**, il suo size **n** e un float **soglia**. La function deve effettuare un ciclo ripetitivo in cui a ogni passo tutti gli elementi dell'array **vettore** vengono dimezzati; il ciclo ripetitivo termina quando la somma degli elementi dell'array **vettore** risulta minore del valore della variabile **soglia**. Valutare e commentare la complessità di tempo della function (operazione dominante: somma; si consiglia di utilizzare la function: float somma_array(float a[],int size)).