# Informatica A

**Cognome Matricola**

**Nome Firma**

### Istruzioni

* Non separate questi fogli. Scrivete la soluzione **solo** **sui fogli distribuiti**, utilizzando il retro delle pagine in caso di necessità. **Cancellate le parti di brutta** (o ripudiate) con un tratto di **penna**.
* Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
* **È possibile scrivere a matita** (e non ricalcare al momento della consegna!).
* È **vietato** utilizzare **calcolatrici** o **telefoni**. Chi tenti di farlo vedrà **annullata** la sua prova.
* È ammessa la consultazione di **libri** e **appunti**, purché con pacata discrezione e senza disturbare.
* Qualsiasi **tentativo** di comunicare con altri studenti comporta **l’espulsione** dall’aula.
* È possibile **ritirarsi senza penalità**.
* Non è possibile lasciare l’aula conservando il tema della prova in corso.
* Tempo a disposizione: **2 h 30 m**

### Valoredegli esercizi, voti parziali e voto finale:

**Esercizio 1 ( 2 punti )**

**Esercizio 2 ( 4 punti )**

**Esercizio 3 ( 8 punti )**

**Esercizio 4 ( 12 punti )**

**Esercizio 5 ( 6 punti )**

**Totale:**

## Esercizio 1 (2 punti)

1. Si costruisca la tabella di verità della seguente espressione booleana, badando alla precedenza tra gli operatori logici. (1 punto).

**not ( ( not ( A and not B ) ) or B or ( C and A ) )**

1. Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri A = 111dec e B = –81dec, li si converta, se ne calcolino la somma (A+B) e la differenza (A–B) in complemento a due e si indichi se si genera riporto sulla colonna dei bit più significativi e se si verifica overflow (1 punto).

## Esercizio 2 ( 4 punti )

Il seguente database è utilizzato da negozio (per nostalgici) che consente di noleggiare CD

CD(Codice, Autore, Titolo, Genere, Durata)

Noleggio(CD, Cliente, DataInizio, DataRestituzione)

Cliente(Codice, Nome, Città)

Scrivere in SQL l’interrogazione che estrae il CD con durata maggiore.

Il seguente database è utilizzato da negozio (per nostalgici) che consente di noleggiare CD

CD(Codice, Autore, Titolo, Genere, Durata)

Noleggio(CD, Cliente, DataInizio, DataRestituzione)

Cliente(Codice, Nome, Città)

Scrivere in SQL l’interrogazione che estrae per ogni cliente la prenotazione più lunga iniziata e conclusa nel 2016.

**Esercizio 3 ( 8 punti )**

Scrivere una funzione che riceve due stringhe: *parola* e *elimina*. La funzione cerca in *parola* tutti i caratteri che compongono la stringa *elimina*, e li rimuove solamente se li trova tutti e nell’ordine in cui compaiono in *elimina*, anche se non consecutivi. L’operazione viene ripetuta finché in *parola* è contenuta un’intera istanza della stringa *elimina*. La funzione, oltre a modificare l’array *parola*, restituisce il numero di volte che ha eliminato i caratteri dell’intera stringa *elimina* da *parola*. Nell’eliminare caratteri non devono essere lasciati buchi, ma *parola* dev’essere ricompattata.

Esempio

parola: amaarrreeeemmmmarrreeaaaarrrmae

elimina: mare

La funzione trova tutte le lettere di “mare” in “aMAaRrrEeeemmmmarrreeaaaarrrmae” che diventa “aarreeemmmmarrreeaaaarrrmae”

La funzione trova tutte le lettere di “mare” in “aarreeeMmmmARrrEeaaaarrrmae” che diventa “aarreeemmmrreaaaarrrmae”

La funzione trova tutte le lettere di “mare” in “aarreeemmMrreAaaaRrrmaE” che diventa “aarreeemmrreaaarrma”

Poi non trova più tutte le lettere e restituisce 3

## Esercizio 4 ( 12 punti )

Si considerino le seguenti definizioni di tipi di dati che rappresentano una lista di visite a siti Web di cui si registra link, data e ora della visita in ordine cronologico (ovviamente lo stesso sito Web può essere stato visitato più volte in momenti diversi) e una lista di statistiche che registrano per ogni sito il numero totale di visite

typedef struct { int giorno, mese, anno; } Data;

typedef struct n {

char link[1000];

Data d;

Ora o;

struct n \*next;

} Visita;

typedef Visita \* Visite;

typedef struct { int ora, min, sec, milli; } Ora;

typedef struct li {

char link[1000];

int visiteTotali;

struct li \*next;

} Statistica;

typedef Statistica \* Statistiche;

Si codifichi in C una funzione che riceve come parametro una lista di visite Ve alloca e restituisce una lista contenente le visite totali per tutti i siti visitati almeno una volta nel 2016.

Statistiche calcola(Visite V)

## Esercizio 5 ( 6 punti )

Si consideri la seguente definizione di albero binario:

typedef struct t { char parola[1000];

struct t \* left, \* right; } Nodo;

typedef Nodo \* Tree;

Due parole si dicono simili se hanno al più due caratteri diversi (cioè se hanno gli stessi caratteri nello stesso ordine, eccetto due di essi che possono essere diversi o mancare nella seconda o essere in più nella seconda parola).

Una catena di parole si dice compatibile col telefono senza fili (cctsf) se ogni parola è simile alle adiacenti.

La funzione int simili (char \*s1, char \*s2); restituisce 1 se s1 e s2 sono simili, 0 altrimenti.

Usando la funzione simili(…) (senza codificarla), si codifichi in C una funzione f che riceve come parametro un albero di parole (secondo la definizione soprastante) e restituisce 1 se tutti i cammini dalla radice alle foglie rappresentano catene cctsf, 0 altrimenti.