# 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 ( 4 punti )**

**Esercizio 4 ( 8 punti )**

**Esercizio 5 (10 punti )**

**Esercizio 6 ( 4 punti )**

**Totale: ( 32 punti )**

## Esercizio 1 (2 punti)

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

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

1. Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due entrambi i numeri A = 119dec e B = –95dec

Si converta A e B in CP2 usando bit

Si calcoli quindi (A+B) e (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 schema rappresenta le informazioni riguardo al noleggio di automobili:

AUTO ( Targa, Modello, Marca )

CLIENTE ( CodiceFiscale, Nome, Cognome, Indirizzo, Telefono, Convenzione )

NOLEGGIO ( CodiceFiscale, Targa, DataInizio, DataFine, Costogg, giorni, km )

Estrarre in SQL i clienti che hanno noleggiato almeno due modelli diversi della stessa marca nei mesi di luglio e agosto 2018.

Il seguente schema rappresenta le informazioni riguardo al noleggio di automobili:

AUTO ( Targa, Modello, Marca )

CLIENTE ( CodiceFiscale, Nome, Cognome, Indirizzo, Telefono, Convenzione )

NOLEGGIO ( CodiceFiscale, Targa, DataInizio, DataFine, Costogg, giorni, km )

Estrarre in SQL il nome e il cognome dei clienti che, avendo noleggiato qualche auto nel 2018, non hanno mai più noleggiato la medesima auto nello stesso anno (l'anno a cui appartiene un noleggio è definito dalla sua data d'inizio).

**Esercizio 3** ( 4 punti )

Si dica cosa stampa il seguente codice

**#include <stdio.h>**

**#include <stdlib.h>**

**typedef struct el {**

**char c;**

**struct el \* next; } nodo;**

**void f(char \* str, nodo \*\* n, int i) {**

**int d = 0;**

**if(i > 12)**

**return;**

**if(i==0)**

**(\*(\*n+i)).c = 'A';**

**else**

**switch(i % 4)**

**{**

**case 0:**

**(\*(\*n+i)).c = (\*(\*n + i - 1)).c;**

**break;**

**case 1:**

**(\*(\*n+i)).c = '-';**

**break;**

**case 2:**

**(\*(\*n+i)).c = '>';**

**break;**

**case 3:**

**(\*(\*n+i)).c = (\*(\*n + i - 3)).c + str[0] - '0';**

**d = 1;**

**break;**

**}**

**f( str+d, n, i+1 );**

**}**

**int main() {**

**char str[3];**

**int dim,i;**

**printf("Inserire le ultime tre cifre del tuo codice persona");**

**scanf("%s", str);**

**printf("%s", str);**

**dim = sizeof(nodo);**

**nodo \* a = (nodo \*)malloc(3 \* 4 \* dim);**

**f(str, &a, 0 );**

**for(i = 0; i <12; i++)**

**printf("%c ", (a+i)-> c);**

**return 0;}**

## Esercizio 4 ( 8 punti )

Definire una funzione

void incastro(int[][N] mat1, int[][M] mat2, int \*riga, int \*col)

che, presa in input una matrice NxN ed una matrice MxM (con N ed M costanti definite tramite #define…, con M<N), verifica se è possibile collocare la seconda matrice all’interno della prima in modo tale che nessun elemento positivo nella seconda matrice si vada a sovrapporre ad un elemento positivo della prima matrice. In caso affermativo, la funzione incastro restituisce le coordinate di una delle celle della prima matrice in cui è possibile collocare l’elemento in posizione (0,0) della seconda matrice.

Si scriva quindi una porzione dei main.c contenente l’invocazione della funzione incastro e la sola dichiarazione di tutte le variabili coinvolte.

Ad esempio:

mat1:  1 -1   1 e    mat2: 1 -1

      -1 1 -1           -1 1

       1 1 1

La funzione restituisce riga 0 e col 1.

## Esercizio 5 ( 10 punti )

Si consideri la seguente definizione di lista:

**typedef struct EL {**

**int dato;**

**struct EL \* next;**

**} nodo;**

**typedef nodo \* lista;**

Scrivere una funzione

**lista intreccia(lista lis1, lista lis2)**

che, prese in input due liste, le *intreccia* in modo tale che un elemento della prima lista sia sempre seguito da un elemento della seconda lista e viceversa. Nel caso in cui una delle due liste sia più lunga dell’altra, gli elementi rimanenti della lista devono comparire in coda alla lista intrecciata.

**Ad esempio:**

Se si forniscono in input:

lis1: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> NULL

lis2:  7 -> 8 -> 9 -> 10 -> NULL

la funzione restituisce:

1 -> 7 -> 2 -> 8 -> 3 -> 9 -> 4 -> 10 -> 5 -> 6 -> NULL

Nota: Assumere per semplicità che le due liste siano non vuote.

Si codifichi una funzione che **cancellaElementiGrandi** che rimuove da una lista tutti gli elementi maggiori della somma degli elementi seguenti

Ad esempio

Input: 1 -> 40 -> 3 -> 15 -> 5 -> 6 -> NULL

La funzione restituisce

1 -> 3 -> 5 -> 6 -> NULL

Esercizio 6 **( 4 punti )**

Sia dato un albero binario (non necessariamente completo) definito come segue:

**typedef struct nodeS { int v;**

**struct nodeS \* left, right; } node;**

**typedef node \* tree;**

Si scriva una funzione:

**int f(tree t)**

che, dato in ingresso l’albero, restituisce 1 se tutte le foglie si trovano allo stesso livello (hanno la stessa profondità), e 0 altrimenti.