# 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 - Algebra di Boole, Aritmetica Binaria, Codifica delle Informazioni (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 ( A and B and C )**

1. Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due entrambi i numeri A = 19dec 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 database descrive le liste dei candidati alle elezioni per il Senato Accademico e i voti scrutinati, che sono caricati man mano che si procede allo spoglio.

Candidato ( Lista, NomeCandidato )

VotoScrutinato ( CodiceProgressivo, Lista, Preferenza )

Le preferenze sono rappresentate dall’attributo Preferenza, che ha valore nullo se la preferenza non è stata espressa o è stata espressa in modo non valido

Estrarre in SQL i nomi dei candidati che hanno avuto più di 40 preferenze.

Candidato ( Lista, NomeCandidato )

VotoScrutinato ( CodiceProgressivo, Lista, Preferenza )

Estrarre in SQL la lista più votata.

**Esercizio 3** ( 4 punti )

Si dica cosa stampa il seguente codice

**#include <stdio.h>**

**#include <stdlib.h>**

**typedef struct el {**

**int intero;**

**char c;**

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

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

**if( i >= 4 )**

**return;**

**if(i==2)**

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

**else**

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

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

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

**}**

**void foo( nodo nodes[], int i ) {**

**int j = 1;**

**while( i-- > 0 ){**

**(\*(nodes)).intero = j;**

**nodes++;**

**j++;**

**}**

**}**

**int main() {**

**char str[] = "AFAJ";**

**int dim,i;**

**for(i=0;i<4;i++)**

**str[i]=str[i]+1;**

**dim = sizeof(nodo);**

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

**foo( a, 4 );**

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

**return 0;**

**}**

## Esercizio 4 ( 8 punti )

Sia data la struttura dati

**typedef struct P { int i,j; } Punto;**

e si assuma che sia stata definita **#define N** ad un preciso valore

Si scriva una funzione

**float f(int M[][N], Punto P[])**

che riceve in ingresso una matrice **M** quadrata di dimensioni **NxN** di interi positivi ed un array **P** di tipo **Punto**. La funzione **f** scorre la matrice **M** e inserisce in ogni elemento del vettore P tutte le coordinate delle celle di **M** dove **M(i,j) = 0**. .

Il vettore **P** restituito deve contenere in totale **NxN** elementi ed le coordinate (-1;-1) in corrispondenza delle celle di **M** dove i valori sono diversi da zero.

## Esercizio 5 ( 10 punti )

Si considerino due liste concatenate semplici di dati, una contenente dati di clienti e una di dati di spese effettuate dai clienti stessi ordinata cronologicamente. La struttura della liste è:

**typedef struct Date { int giorno; int mese; int anno; } Data;**

**typedef struct Cli { int CodiceCliente;**

**char cognome[100], nome[100];**

**int puntiAccumulati; } Cliente;**

**typedef struct CliN { Cliente C;**

**struct CliN \* next; } ClienteNodo;**

**typedef ClienteNodo \* ListaDiClienti;**

**typedef struct ES { char nomeProdotto[100];**

**int prezzoUnitario;**

**int quantitaAcquistata;**

**struct ES \* next; } ElementoScontrino;**

**typedef ElementoScontrino \* Scontrino;**

**typedef struct Sp { int CodiceCliente;**

**Scontrino s;**

**Data data;**

**struct Vi \* next; } Spesa;**

**typedef Spesa \* ListaDiSpese;**

Si codifichi una funzione **clienteMigliore** che restituisca una struct di tipo **Cliente** contenente i dati del cliente che ha speso più soldi in totale (si supponga che questo sia unico).

Si codifichi una funzione **pulisciLista** che cancella dalla lista dei clienti i clienti non hanno effettuato acquisti dal 2015 al 2019.

Esercizio 6 **( 4 punti )**

Si consideri la seguente definizione di albero:

**typedef struct** n **{ int** dato**;**

**struct \*** left**, \*** right**; }** Node**;**

**typedef** Node **\*** Tree**;**

Si codifichi in C una funzione **media** che restituisce la media dei valori interi contenuti su tutti i nodi foglia.