# 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 ( 7 punti )**

**Esercizio 5 ( 8 punti )**

**Esercizio 6 ( 4 punti )**

**Totale: ( 29 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 ( not 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 i numeri A = 19dec e B = –85dec, 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 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 preferenze in numero compreso tra 2 e 5.

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 il candidato con più preferenze di ciascuna lista.

**Esercizio 3** ( 4 punti )

Si dica cosa stampa il seguente codice

**#include <stdio.h>**

**#include <stdlib.h>**

**#include<conio.h>**

**#define N 10**

**typedef struct el {**

**int os;**

**char \* cp;**

**struct el \* next; } elem;**

**void f(char \* s, elem \*\* e, int i) {**

**if( i >= 4 )**

**return;**

**(\*e)[i].cp = (char \*)malloc(1);**

**\*( (\*(\*e+i)).cp ) = s[0] + (\*e+i)->os;**

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

**f( s+1, e, i+1 );**

**}**

**void foo( elem v[], int i ) {**

**int k = 1;**

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

**(\*(v++)).os = k--;**

**}**

**int main() {**

**char s[] = "BIBQ";**

**int so = sizeof(elem);**

**elem \* v = (elem \*)malloc( 4 \* so );**

**foo( v, 4 );**

**f( s, &v, 0 );**

**getch();**

**return 0;**

**}**

## Esercizio 4 ( 7 punti )

Sia data la struttura dati

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

Si scriva una funzione

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

che riceve in ingresso una matrice M quadrata NxN (con N costante predefinita con l’istruzione #define N …) di interi positivi ed un array P di dimensione N di punti. La funzione cerca nella matrice i valori che stanno nelle caselle indicate dai punti dell’array P e restituisce la media dei valori trovati se tutte le coordinate sono valide (cioè se i punti sono realmente coordinate della matrice) o -1 se anche solo uno dei punti non è valido.

## Esercizio 5 ( 8 punti )

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

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

**typedef struct Cli { int CodiceCliente;**

**char \* cognome, \* nome;**

**int puntiAccumulati;**

**struct Cli \* next } Cliente;**

**typedef Cliente \* ListaDiClienti;**

**typedef struct ES { char \* nomeProdotto;**

**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 Viaggio \* ListaDiViaggi;**

Si codifichi una funzione che aggiorna automaticamente dopo ogni spesa i punti totali del cliente aggiungendo un punto per ogni 10 euro spesi (arrotondando per difetto). Inoltre, i clienti che in un mese spendono più di 5000 euro hanno diritto a ricevere il doppio dei punti su tutte le spese del mese.

Esercizio 6 **( 4 punti )**

Si consideri la seguente definizione di albero:

**typedef struct** n **{ char** dato**;**

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

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

Si codifichi in C una funzione che stampa a video la “frase” formata dalla sequenza di caratteri posti sui nodi foglia, in ordine da quello più a sinistra fino a quello più a destra.