# 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 ( 6 punti )**

**Esercizio 5 ( 5 punti )**

**Esercizio 6 ( 6 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 and ( B and (not A) ) ) and ( ( A or ( not B ) ) and C )**

1. Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri A=126dec e B=–111dec, li si converta, se ne calcoli 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 schema descrive le misurazioni di temperatura e umidità in tutti i comuni del territorio italiano.

Rilevazione ( Data, Comune, TempMax, TempMin, PercentualeUmidità )

Comune ( Nome, Provincia, Regione, NumeroAbitanti )

Scrivere in SQL l'interrogazione che estrae i comuni siciliani in cui non si è mai registrata una temperatura inferiore a 0 °C.

Il seguente schema descrive le misurazioni di temperatura e umidità in tutti i comuni del territorio italiano.

Rilevazione ( Data, Comune, TempMax, TempMin, PercentualeUmidità )

Comune ( Nome, Provincia, Regione, NumeroAbitanti )

Esprimere in SQL l’interrogazione che estrae i comuni in cui la temperatura massima in tutto il 2013 si è sempre mantenuta tra i 10 e i 25 °C.

**Esercizio 3** ( 4 punti )

Si dica cosa stampa il seguente codice e si spieghi cosa calcola la funzione mistero

#include <stdio.h>

#define N 10

int mistero(int a, int b) {

for (; a>=b ;a=a-b)

;

return a;

}

int main() {

int vett[N]={1,1,2,3,5,8,13,21,34,55};

int i,j,k;

for (i=0;i<N;i++){

k=1;

for (j=2;j<vett[i];j++)

if (mistero(vett[i],j)==0)

k=0;

if (k==0)

printf ("Falso\n");

else

printf ("Vero\n");

}

getch();

return 0;

}

## Esercizio 4 ( 6 punti )

Si consideri una matrice di caratteri di dimensione NxN. Si codifichi in C una funzione

char f(char M[][N], int K)

che, se esiste una colonna in M in cui ci sono almeno K caratteri consecutivi tutti uguali e diversi da ‘\0’ (si consideri K minore di N per ipotesi), restituisce il carattere presente su tale colonna, altrimenti restituisce il carattere ‘\0’. Si trascuri (per semplicità) il caso in cui più colonne godano della proprietà da verificare.

## Esercizio 5 ( 5 punti )

Si consideri la seguente struttura dati usata da una associazione di maratoneti:

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

**typedef struct { char cognome[N], nome[N];**

**int eta; /\* età della persona \*/ } Persona;**

**typedef struct Item { Persona p;**

**int punti;**

**struct Item \* next } Socio;**

**typedef Socio \* ListaDiSoci;**

**typedef struct Node { Data d;**

**Persona \* ordineDiArrivo; /\* la lista contiene i**

**concorrenti in ordine**

**di arrivo al traguardo \*/**

**struct Node \* next; } Maratona;**

**typedef Maratona \* ListaDiMaratone;**

Si codifichi una funzione che, ricevuta in ingresso la lista dei soci e la lista delle maratone calcoli e assegni il valore nel campo **punti** (il cui valore iniziale è indefinito e non significativo) di ogni socio attribuendo per ogni maratona 10 punti a chi è arrivato primo, 9 punti a chi è arrivato secondo, 8 punti a chi è arrivato terzo e così a scalare fino a 1 punto a chi è arrivato decimo. La funzione restituisce poi l’età del socio con più punti (si trascuri la possibilità di un parimerito). Si consiglia fortemente l’uso di funzioni ausiliarie.

## Esercizio 6 ( 6 punti )

Si consideri la seguente definizione di un albero contenente cognomi e nomi di persone

**typedef struct EL { char cognome[N], nome[N];**

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

**typedef node \* tree;**

Si scriva una funzione che riceve in ingresso un albero, verifica che l’albero non contenga omonimi (cioè persone con lo stesso nome e lo stesso cognome) e, in caso la verifica vada a buon fine, restituisca una lista contenente i nomi e i cognomi delle persone in ordine alfabetico, mentre in caso la verifica non vada a buon fine, restituisca NULL.