Informatica A

**Cognome**

**Nome**

**Matricola o Codice studente**

Istruzioni

1. 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**.
2. Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
3. **È possibile scrivere a matita** (e non ricalcare al momento della consegna).
4. È **vietato** utilizzare **calcolatrici** o **telefoni**. Chi tenti di farlo vedrà **annullata** la sua prova.
5. È ammessa la consultazione di **libri** e **appunti**, purché con pacata discrezione e senza disturbare.
6. Qualsiasi **tentativo** di comunicare con altri studenti comporta **l’espulsione** dall’aula.
7. È possibile **ritirarsi senza penalità**.
8. Non è possibile lasciare l’aula conservando il tema della prova in corso.
9. Tempo a disposizione: 2 ore e 40 minuti

**Esercizio 1 ( 2 punti )**

**Esercizio 2 ( 4 punti )**

**Esercizio 3 ( 4 punti )**

**Esercizio 4 ( 6 punti )**

**Esercizio 5 ( 12 punti )**

**Esercizio 6 ( 4 punti )**

**Voto finale:**

Esercizio 1 ( 2 punti )

Si costruisca la tabella di verità della seguente espressione booleana.

( C or ( not B ) ) or ( ( not A ) or B)

Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri A = -101 e B = 18, 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.

Esercizio 2 ( 4 punti )

Si consideri il seguente schema di base di dati relativa ad autoveicoli:

MARCA(Nome, Nazione)

TIPOVEICOLO(NomeMarca, Modello, Propulsione, Contatore)

ACQUIRENTE(CodFisc, Targa, NomeMarca, Modello, Propulsione, NazioneDiNascita)

Scrivere una query SQL che estrae la nazione che produce almeno un veicolo a propulsione “elettrica”.

MARCA(Nome, Nazione)

TIPOVEICOLO(NomeMarca, Modello, Propulsione, Contatore)

ACQUIRENTE(CodFisc, Targa, NomeMarca, Modello, Propulsione, NazioneDiNascita)

Scrivere una query SQL che estrae la cittadinanza degli acquirenti che abbiano acquistato almeno due veicoli e che non abbiano mai acquistato veicoli in produzione in una nazione diversa da quella in cui sono nati.

Esercizio 3 ( 4 punti )

Si dica cosa stampa il seguente codice e si spieghi cosa calcola la funzione mistero1

#include <stdio.h>

void mistero1(int x, int y);

int mistero2(int \*x, int y);

int mistero3(int x, int \* y, int z);

int main() {

char c;int i, v[4] = { 15, 4, 12, 625 };

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

printf("\n%d: ", v[i]); mistero1(v[i],2);

}

return 0;

}

void mistero1(int x, int y) {

int z = mistero2(&x,y);

if( z > 1 )

printf("%d ", z);

if( x > 1 )

mistero1(x,y);

}

int mistero2(int \*x, int y) {

if( \*x <= 0 )

return 0;

if( \*x == 1 )

return 1;

return mistero3(y,x,1);

}

int mistero3(int x, int \* y, int z) {

if( \*y % x != 0 )

return mistero3( x+z, y, z );

\*y = \*y / x;

return x;

}

Esercizio 4 ( 6 punti )

Scrivere una funzione verifica che prende in ingresso una matrice di caratteri M di dimensioni NxN (supponendo N predefinita con #define), una stringa S, e due indici i e j. La funzione verifica se in M esiste un percorso connesso (una sequenza di celle consecutive in direzione verticale, orizzontale o diagonale) che inizia nella posizione (i, j) che corrisponde a S. Se il percorso esiste la funzione restituisce 1, altrimenti 0. Per semplicità si supponga di poter passare più volte sulla stessa cella della matrice M mentre si ricerca la parola S.

Con M = a c k

s m o

d f t

S = “amo”, i = 0, j = 0;

La funzione restituisce 1

Invece con la stessa M, ma

S = “doc”, i = 2, j = 0;

La funzione restituisce 0.

Esercizio 5 ( 12 punti )

Si definisca un tipo di dato atto a contenere una lista di studenti con i campi numero di matricola, nome, cognome e voto all’esame di “Informatica A”. Si implementi quindi la funzione eliminaStudente che rimuove da una lista definita come sopra tutti gli studenti che hanno conseguito un punteggio all’esame minore di 18.

Si implementi quindi la funzione listaDiStudendiPerVoto che riceve una lista definita come sopra e restituisce una lista di liste di studenti composta da sottoliste costruite dividendo gli studenti per voto ottenuto. Avremo quindi la sottolista degli studenti che hanno preso 18, quella degli studenti che hanno preso 19, e così via fino alla lista degli studenti che hanno preso 30 (per semplicità non si consideri il “30 e Lode” e si immagini non esistano voti mai assegnati).

Esercizio **6** ( 4 punti )

Si consideri la seguente definizione di un albero binario

typedef struct nodeS { int val;

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

typedef node \* tree;

Scrivere una funzione che riceve un albero T e restituisce 1 se esiste almeno un percorso dalla radice ad una foglia in cui si alternano elementi pari e dispari, 0 altrimenti.