

# Informatica A – 29-1-2020

Cognome (in STAMPATELLO) \_\_\_\_\_

Nome \_\_\_\_\_

Matricola o Codice studente \_\_\_\_\_

## Istruzioni

Non separate questi fogli. Scrivete la soluzione **solo sui fogli distribuiti**, utilizzando il retro delle pagine in caso di necessità.

È **possibile scrivere a matita** (e non ricalcare al momento della consegna).

Ogni parte non cancellata sarà considerata parte integrante della soluzione.

È **vietato** utilizzare **calcolatrici** o **telefoni**. Chi tenti di farlo vedrà **annullata** la sua prova.

Non è ammesso consultare **libri** o **appunti** o altro.

Qualsiasi **tentativo** di comunicare con altri studenti comporta **l'espulsione** dall'aula.

È possibile **ritirarsi senza penalità** lasciando il tema d'esame con nome e cognome.

Non è possibile lasciare l'aula conservando il tema della prova in corso.

Tempo a disposizione: 2 ore e 30 minuti

Esercizio 1 ( 2 punti ) \_\_\_\_\_

Esercizio 2 ( 4 punti ) \_\_\_\_\_

Esercizio 3 ( 4 punti ) \_\_\_\_\_

Esercizio 4 ( 8 punti ) \_\_\_\_\_

Esercizio 5 ( 10 punti ) \_\_\_\_\_

Esercizio 6 ( 4 punti ) \_\_\_\_\_

Voto finale: \_\_\_\_\_

### Esercizio 1 ( 2 punti )

Si costruisca la tabella di verità della seguente espressione booleana.

$( C \text{ and } ( A \text{ or } ( \text{not } B ) ) ) \text{ or } ( A \text{ or } B )$

Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri  $A = -170$  e  $B = 19$ , 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 )

Lo schema sottostante descrive gli studenti, i docenti, gli insegnamenti erogati e gli esami sostenuti dagli studenti.

STUDENTE (MATRICOLA, COGNOME, NOME, DATANASCITA, DATAIMMATRICOL)

INSEGNAMENTO (CODINS, ANNOACCAD, NOMEINS, MATRDOCENTE)

ESAME (MATRICOLA, CODINS, DATA, VOTO)

DOCENTE (MATRICOLA, NOME, COGNOME, DIPARTIMENTO)

Scrivere una query che estrae nome e cognome dei docenti che hanno tenuto almeno due insegnamenti nell'anno 2018-2019.

STUDENTE (MATRICOLA, COGNOME, NOME, DATANASCITA, DATAIMMATRICOL)

INSEGNAMENTO (CODINS, ANNOACCAD, NOMEINS, MATRDOCENTE)

ESAME (MATRICOLA, CODINS, DATA, VOTO)

DOCENTE (MATRICOLA, NOME, COGNOME, DIPARTIMENTO)

Scrivere una query che estrae per ogni docente il numero di esami che ha verbalizzato nel 2019.

### Esercizio 3 ( 4 punti )

Si dica cosa stampa il seguente codice

```
#include<stdio.h>
#include<stdlib.h>
#define M 6

void m(int k);
void f(int * a, int b, int c);
void g(int* d, int e);

int main(){
    m(1);
    m(2);
    m(4);
    m(12);
    m(24);
}

void m(int k) {
    int* h = (int*) malloc(sizeof(int) * M)

    f(h + M - 1, k, 0);

    g(h, M);

    if(h)
        free(h);
}

void f(int * a, int b, int c) {
    if(c == M)
        return;
    c++;
    *(a) = b % 2;
    a--;

    f(a, b/2, c);
}

void g(int* d, int e) {
    if (e) {
        printf("%d ",*d);
        d++;
        g(d, e - 1);
    } else printf("\n");
}
```

## Esercizio 4 ( 8 punti )

Si scriva una funzione **comprimiStringa** che prende in ingresso una sequenza di caratteri composta solo da lettere dell'alfabeto e la comprime nel modo seguente: ogni sequenza di due o più caratteri identici consecutivi viene sostituita da due caratteri [NUMERO] [CARATTERE] dove NUMERO è il numero di occorrenze e CARATTERE è il carattere nella stringa.

Ad esempio la stringa **ciiaaaaoooo** viene compressa in **c3i4a5o**

Si scriva poi una funzione **decomprimiStringa** che prende in ingresso una sequenza di caratteri compressa da **comprimiStringa** e restituisce la sequenza originale.

Ad esempio la stringa **c3i4a5o** viene decompressa in **ciiaaaaoooo**

Per semplificare lo sviluppo, si assuma che non possa mai capitare che vi siano più di 9 occorrenze consecutive dello stesso carattere e che nessuna stringa sia più lunga di 1000 caratteri.

Si ricordi che per passare da un valore char all'intero che esso rappresenta basta sottrarre il carattere **'0'**.



## Esercizio 5 ( 10 punti )

Sia data una lista con valori interi. Si scriva una funzione ricorsiva

**int listePerDue(Lista l)**

che restituisce **1** se la lista è vuota o se ciascun elemento ha un successore con valore almeno doppio del valore corrente, **0** altrimenti.

Esempi:

**Input: 2->3->7 Output: 0**

**Input: 2 Output: 1**

**Input: 2->10->20 Output: 1**

Si scriva poi una funzione **correggiLista** che aumenta i valori che portano a violare il vincolo fino al minimo indispensabile per rispettarlo. La funzione inoltre restituirà il totale degli incrementi che sono stati necessari. Si noti che l'aumento di un elemento può rendere necessario aumentare i suoi successori che magari rispettavano il vincolo prima della correzione.

Ad esempio, se l'input è **2->3->7**, la funzione modificherà la lista rendendola **2->4->8** e restituirà **2**.





## Esercizio 6 ( 4 punti )

Si consideri la seguente definizione di un albero binario:

```
typedef struct node_s {  
    char v;  
    struct node_s * left, * right;  
} node_t;  
typedef node_t * tree;
```

Scrivere una funzione che preso in ingresso il puntatore alla radice di un albero restituisca il numero di percorsi palindromi dalla radice alle foglie.