



Politecnico di Milano

Dipartimento di Elettronica, Informazione e Bioingegneria

Informatica A - a.a 2021/2022 - 15 gennaio 2022

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** con un tratto di penna.
- Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
- **NON è possibile scrivere a matita.**
- Scrivere nome e cognome su tutti i fogli. Non verranno corretti compiti non firmati o con nome illeggibile
- È vietato utilizzare **calcolatrici, telefoni o pc**. Chi tenti di farlo vedrà **annullata** la sua prova.
- **Non è ammessa la consultazione di libri e appunti.**
- 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: **2h**

Valore indicativo degli esercizi, voti parziali e voto finale:

Esercizio 1 6 punti _____

Esercizio 2 7 punti _____

Esercizio 3 7 punti _____

Esercizio 4 12 punti _____

Totale(32) _____

Esercizio 1 - Algebra di Boole, Aritmetica Binaria, Codifica delle Informazioni (6 punti)

- (a) Si costruisca la tabella di verità della seguente espressione booleana in tre variabili, badando alla precedenza tra gli operatori logici. Eventualmente si aggiungano parentesi.
Scrivere l'espressione semplificata. Non si accetteranno soluzioni senza il procedimento. (2 punto)

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

Semplificazione: AB+C Output= 000 0 001 1 010 0 011 1 100 0 101 1 110 1 111 1

- (b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri A = 35dec e B = -47, 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. Non si accetteranno soluzioni senza il procedimento. (2 punto)

7 bit minimo

$$35 = 0100011 \quad 47 = 0101111 \quad -47 = 1010001$$

$$35-47 = 1110100 \text{ No OF né CR (operatori discordi sempre possibile l'operazione)} \quad 35+47 = 1010010 \\ = -46 \text{ No CR ma OF}$$

NOME e COGNOME: _____

- (c) Si converta il numero 24,3 in virgola fissa e in virgola mobile con codifica IEEE 754 con precisione singola. Non si accetteranno soluzioni senza il procedimento. (2 punto)

24 = 11000 0.3 = 0.01001 con 1001 parte periodica

virgola vissa = 24.0100110011001... 1001 periodico virgola mobile IEEE754 esponente = 127+4
= 131 = 10000011 mantissa = 1000010011001100110 segno = 0

Esercizio 2 - Teoria (7 punti)

- (a) Quando può verificarsi un errore legato all'heap?
- Quando si passa ad un sottoprogramma un array come parametro
 - Quando viene dichiarata una variabile locale in un sottoprogramma
 - Quando viene usato un variabile dichiarata in modo dinamico x
 - Quando si modifica il puntatore di un array
- (b) Il programma che viene eseguito immediatamente dopo l'accensione del PC si trova:
- Nella RAM
 - Nel DVD drive
 - Nell'hard disk
 - Nessuna delle precedenti x
- (c) Quale dei seguenti numeri è un numero primo?
- 010 x
 - 0100
 - 01000
 - Nessuno dei precedenti
- (d) 11001 rappresenta
- 25 in base 2 in cp2
 - 7 in base 2 in cp2 x
 - 9 in base 2 in cp2
 - Nessuna delle precedenti
- (e) Quale delle seguenti espressioni è una tautologia
- (A or not B) and (not A or B)
 - (A and not B) or (not A and B)
 - (A or not B) or (not A or B) x
 - Nessuna delle precedenti
- (f) Cosa stampa il seguente programma?

```
#include <stdio.h>
void funz(int * val);
int main(void) {
    int val[10];
    int i;

    val[0] = 5;
    for (i=1; i<10; i++)
        val[i] = val[i-1]*2;

    funz(val);
    return 0;
}

void funz(int * val){
    int v;
    v=*(val+5);
    printf("%d\n",v);
    val = val+1;
    printf("%d\n",*val);
}
```

(g) **Matlab (7 punti)**

Scrivere il codice Matlab che restituisca i valori richiesti. Attenersi al numero massimo di righe di codice indicato.

- Creare una matrice di 5 colonne e righe casuali tra 2 e 11, contenente tutti valori diversi e casuali da 0 a 10. (1 riga - 1 punto)

```
A = randi([0 10], [randi([2 11]) 5])
```

- Modificare le righe pari in modo che contengano la media delle righe stesse (1 riga - 1 punto)

```
A(2:2:end,:) = mean(A(2:2:end,:),2)
```

- Cancellare le colonne che iniziano con un numero inferiore a 10 (1 riga - 1 punto)

```
A(:,A(1,:)<10) = []
```

- Calcolare la radice quadrata della somma di tutte le celle della matrice (1 riga - 1 punti)

```
sqrt(sum(sum(A)))
```

- Sostituire le celle della matrice multiple di 3 con valori casuali da -1 a -3 (1 riga - 1 punti)

```
A(mod(A,3)==0) = randi([-3 -1])
```

oppure

```
A(mod(A,3)==0) = randi([-3 -1],size(A(mod(A,3)==0)))
```

- Scrivere una funzione Matlab che presi in ingresso la matrice, restituisca due nuove matrici: la prima contenente gli elementi della matrice elevati ognuno per un numero casuale tra 0 e 5; la seconda formata dalla matrice triangolare bassa. (2 punti)

```
function [M,N] = funzione(A)
    N = A.^randi([0 5]);
    for (i=1:1:size(A,1))
        for (j=1:1:size(A,2))
            if (j<i)
                M(i,j) = A(i,j);
            else
                M(i,j) = 0;
            end
        end
    end
```

(a) **Programmazione C (12 punti)**

Si ipotizzi di volere scrivere un programma per salvare i dati di varie città all'interno di una lista. Ogni città che viene inserita nella lista ha un nome, una nazione di appartenenza e una popolazione.

1. Si definiscano le strutture dati necessarie a salvare le città. (1 punto)

All'interno della lista le città vengono salvate in maniera ordinata e sono ordinate alfabeticamente per nazione di appartenenza; se due città appartengono alla stessa nazione sono poi ordinate alfabeticamente per nome.

Ad esempio:

-Italia, Milano, popolazioneDiMilano- viene prima di -Italia, Roma, popolazioneDiRoma-. Hanno la stessa nazione di appartenenza ma Milano viene alfabeticamente prima di Roma.

-Italia, Milano, popolazioneDiMilano- invece viene dopo -Grecia, Atene, popolazioneDiAtene- perché Grecia viene prima alfabeticamente di Italia.

2. Si definisca una funzione che date due città restituisca quale “viene prima dell'altra” secondo il criterio di ordinamento appena definito. La funzione restituisca -1 se viene prima la prima città, oppure 1 se viene prima la seconda città. La funzione restituisca infine 0 se le due città sono uguali secondo il criterio di ordinamento, ovvero se hanno la stessa nazione di appartenenza e lo stesso nome. (2 punto)

3. Si definisca una funzione in grado di gestire l'inserimento di una nuova città all'interno della lista. Si faccia attenzione a non permettere l'inserimento di doppioni. Ovvero, si impedisca l'inserimento di città che se confrontati con una città già presente nella lista risulterebbero uguali. Fare uso della funzione al punto 2(4 punti)

4. Si definisca una funzione ricorsiva che data una nazione elimina dalla lista tutte le città che appartengono a quella nazione. (5 punti).

5. (Solo per chi deve recuperare il laboratorio) Scrivere una funzione che, preso in ingresso un array di stringhe (ed ulteriori parametri ove necessario), modifichi le stringhe sostituendo le vocali come segue:

```
a = 7
e = 2
i = 1
o = 0
u = 8
```

Le stringhe devono essere anche convertite tutte in caratteri maiuscoli.

La funzione restituisce anche il numero di caratteri totali salvati nell'array ed il numero di vocali sostituite.

```
typedef struct node{
    char nome[20];
    char nazione[20];
    int pop;
    struct node *next;
} node;

typedef node* ptrNode;

int confrontaNazione(ptrNode a, ptrNode b){
    if (strcmp(a->nazione, b->nazione)<0)
```

```

        return -1;
    if (strcmp(a->nazione, b->nazione)<0)
        return 1;
    if (strcmp(a->nome, b->nome)<0)
        return -1;
    if (strcmp(a->nome, b->nome)>0)
        return 1;
    return 0;
}

ptrNode inserisciNodoOrdinato(ptrNode testa, char nome[], char nazione[], int popolazione){
    ptrNode nuovo = (ptrNode)malloc(sizeof(node));
    strcpy(nuovo->nome,nome);
    strcpy(nuovo->nazione,nazione);
    nuovo->pop = popolazione;
    nuovo->next = NULL;

    if (testa == NULL)
        return nuovo;

    ptrNode cursor = testa;
    ptrNode prec = NULL;

    while((cursor != NULL) && (confrontaNazione(cursor, nuovo) == -1)){
        prec = cursor;
        cursor = cursor->next;
    }

    if (cursor == NULL){
        prec -> next = nuovo;
        return testa;
    }

    else if (confronta(cursor, nuovo) == 0){
        free(nuovo);
        return testa;
    }

    else {
        prec->next = nuovo;
        nuovo->next = cursor;
        return testa;
    }
}

ptrNode inserisciNodoOrdinatoRic(ptrNode testa, char nome[], char nazione[], int popolazione){
    if (testa == NULL){
        ptrNode nuovo = (ptrNode)malloc(sizeof(node));
        strcpy(nuovo->nazione, nazione);
        strcpy(nuovo->nome, nome);
        nuovo->popolazione = popolazione;
        nuovo->next = testa->next;
        return nuovo;
    }

    if (confrontaNazione(cursor, nuovo) == -1)
        testa->next = inserisciNodoOrdinatoRic(testa->next, nome, nazione, popolazione);

    else (confrontaNazione(cursor, nuovo) == 0)
        return testa;
}

```

```
else{
    ptrNode nuovo = (ptrNode)malloc(sizeof(node));
    strcpy(nuovo->nazione, nazione);
    strcpy(nuovo->nome, nome);
    nuovo->popolazione = popolazione;
    nuovo->next = testa->next;
    return nuovo;
}

ptrNode eliminaNazione(ptrNode testa, char nazione[]){
    if (testa==NULL)
        return testa;
    if (strcmp(nazione,testa->nazione) == 0){
        ptrNode temp = testa;
        testa = testa->next;
        free(temp);
        return eliminaNazione(testa, nazione);
    }

    testa->next = eliminaNazione(testa->next, nazione);
    return testa;
}
```