



Politecnico di Milano

## Dipartimento di Elettronica, Informazione e Bioingegneria

**Informatica A - a.a 2018/2019 - 19 Luglio 2019**

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.**
- 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: **2h30m**

### Valore indicativo degli esercizi, voti parziali e voto finale:

Esercizio 1	3 punti	_____
Esercizio 2	3 punti	_____
Esercizio 3	4 punti	_____
Esercizio 4	6 punti	_____
Esercizio 5	12 punti	_____
<b>Totale(28)</b>		_____

**Esercizio 1 - Algebra di Boole, Aritmetica Binaria, Codifica delle Informazioni (3 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. Non si accetteranno soluzioni senza il procedimento. (1 punto)

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

A	B	C	
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

**Risposta:**

**Semplificata:**  $A + BC$

A	B	C	OUT
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

- (b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri  $A = 75_{dec}$  e  $B = -53$ , 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. (1 punto)

**Risposta:**

$75_d = 1001011_b$

$cp2 = 01001011_b$

$53_d = 0110101_b$

$-53_d \text{ cp2} = 1001011_b$

A + B

riporto	11 1 11	<b>Riporto perduto ma non overflow</b>
A +	01001011b	
B	11001011b	
A+B	00010110b	

A - B

riporto	1111111	<b>No riporto perduto, ma overflow</b>
A -	01001011b	
B	00110101b	
A-B	10000000b = -128d	

Segni concordi, risultato discorde

con 8 bit si possono rappresentare numeri da -128 a 127

- (c) Si converta il numero 37.625 in virgola fissa e in virgola mobile con codifica IEEE 754, sapendo che  $1/2 = 0.5$ ,  $1/4 = 0.25$ ,  $1/8 = 0.125$ ,  $1/16 = 0.0625$ ,  $1/32 = 0.03125$ ,  $1/64 = 0.015625$ , e  $1/128 = 0.0078125$ . Non si accetteranno soluzioni senza il procedimento. (1 punto)

**Risposta:**

6bit + segno

37d = 100101b

$0.5 + 0.125 = 0.101$

37.625 = 0100101.101 j- Virgola fissa

segno = 0

mantissa = 1.00101101 non normalizzata

exp = 5 + 127 = 132d = 10000100

IEEE754

segno (1bit) = 0

esponente (8bit) = 10000100

mantissa (23bit) = 001 0110 1000 0000 0000 0000 - normalizzato eliminando il primo 1

## Esercizio 2 - Domanda di teoria (3 punti)

1. Illustrare la differenza tra linguaggio compilato e linguaggio interpretato

2. Riportare la porzione di codice in C e Matlab, per effettuare un ciclo for che sommi i primi 100 numeri.

**Risposta:**

MATLAB

```
c = 0;  
for (i=1:1:100)  
    c = c+i;  
end
```

C

```
int i;  
int c = 0;  
for (i=0; i<=100; i++)  
    c = c+1;
```

### Esercizio 3 - Esercizio C (4 punti)

Scrivere la funzione TrovaCompresi che prende in ingresso un vettore di numeri interi VETT e due numeri interi MIN e MAX. La funzione TrovaCompresi restituisce al programma chiamante (main) il numero di elementi del vettore VETT compresi tra MIN e MAX (estremi inclusi) e la media aritmetica di tali valori.

ES.

Se VETT = [1 2 3 4 5 6]

e MIN = 3 e MAX = 5.

Il numero di elementi tra 3 e 5 è 3 e la media è  $(3+4+5)/3 = 4$

Si scriva inoltre il prototipo della funzione e la porzione di programma con l'invocazione della funzione (non è necessario scrivere tutto il main, scanf ecc.).

#### Risposta:

```
#include <stdio.h>

int trovaCompresi(int values[], int len, int min, int max, float *media);

int main(int argc, const char * argv[]) {
    int val[6] = {1,2,3,4,5,6};
    float media;
    int cont;
    cont = trovaCompresi(val, 6, 3, 5, &media);
    printf("Il numero di elementi tra 3 e 5  %d e la media  %.2f",cont, media);
    return 0;
}

int trovaCompresi(int values[], int len, int min, int max, float *media){
    int i, cont = 0;
    *media = 0;

    for (i=0; i<len; i++)
    {
        if ((values[i]>=min) && (values[i]<=max))
        {
            cont++;
            *media = *media + values[i];
        }
    }
    *media = *media/cont;
    return cont;
}
```

#### Esercizio 4 - Matlab (6 punti)

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

- (a) Scrivere una funzione che presi in ingresso tre numeri, restituisca
- 1 se i tre numeri sono una terna cartesiana ( $A^2+B^2=C^2$ )
  - 0 altrimenti (max 8 riga) (2 punto)

**Risposta:**

```
function out = luglio19_19(a,b,c)
    ipo = sqrt(a^2+b^2);
    if (ipo == c)
        out = 1
    else
        out = 0
    end
end
```

- (b) Generare una matrice M con 5 righe e 7 colonne, contenente numeri casuali tra -15 e 15 (max 1 riga) (1 punto).

**Risposta:**

```
randi([-15 15],[5 7])
```

- (c) Eliminare la riga con media più bassa (max 2 riga) (1 punto).

**Risposta:**

```
M(mean(M,2)==min(mean(M,2)),:)=[]
```

- (d) Moltiplicare le colonne dispari per la media dell'intera matrice (max 1 riga) (1 punto).

**Risposta:**

```
M(:,1:2:end) = M(:,1:2:end) * mean(mean(M));
```

- (e) Inserire, dopo la prima colonna, una colonna contenente multipli di 3 da 3 a 15(max 2 righe) (1 punto).

**Risposta:**

```
M = [M(:,1) (3:3:15)' M(:,2:end)]  
oppure  
a = 3:3:15  
M = [M(:,1) a' M(:,2:end)]
```



## Esercizio 5 - Programmazione C Liste (12 punti)

Si ipotizzi di dovere immagazzinare le letture realizzate da un sensore di temperatura. Una lettura è composta da l'orario in cui è stata effettuata (HH:MM:SS), una stringa che indica il locale in cui è stata effettuata la lettura (cucina, salotto, camera1, camera2, ecc.), e il valore stesso della lettura in gradi Celsius.

L'immagazzinamento delle letture deve avvenire mediante una lista, ordinata per orario di lettura. L'inserimento di una nuova misura viene fatto in coda. La lista può contenere al massimo 100 misurazioni; una volta raggiunte le 100 misure, inserendo un nuovo nodo in coda viene automaticamente eliminato il primo nodo in testa.

Si assuma che, inserendo sempre il nuovo nodo in coda, la lista sia già sempre ordinata.

Svolgere l'esercizio attenendosi a quanto richiesto. **NON È RICHiesto SCRIVERE IL MAIN.**

1. Si definiscano le strutture dati necessarie allo sviluppo di questo programma. (1 punto)
2. Scrivere la funzione **RICORSIVA** *int contaNodi(ptrLista testa)* che conta il numero di nodi presenti nella lista. (3 punti)
3. Facendo uso della funzione al punto precedente, scrivere la funzione *ptrLista aggiungiLetture(ptrLista testa, int ora, int minuto, int secondo, char \*locale, int valore)* per inserire un nuovo nodo. Se sono presenti già 100 letture, la prima lettura viene automaticamente cancellata. (4 punti)
4. Scrivere una funzione che riceve due orari ed elimina dalla lista tutte le letture avvenute nella finestra temporale definita dai due orari. Si ipotizzi che il primo orario sia sempre "prima" del secondo orario. *ptrLista rimuoviLetture(ptrLista int daOra, int aOra)* (4 punti)

### Risposta:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

typedef struct node{
    int HH,MM,SS;
    char locale[20];
    int gradi;
    struct node *next;
}node;

typedef node *ptrLista;

int contaNodi(ptrLista lista){
    if (lista==NULL)
        return 0;
    return 1 + contaNodi(lista->next);
}

ptrLista aggiungiLetture(ptrLista lista, int ora, int minuto, int secondo, char locale[20], int valore){
    ptrLista temp;
    if ((contaNodi(lista))>=100){
        temp = lista;
        lista = lista->next;
    }
    temp->HH = ora;
    temp->MM = minuto;
    temp->SS = secondo;
    strcpy(temp->locale, locale);
    temp->gradi = valore;
    temp->next = NULL;
    if (lista == NULL)
        lista = temp;
    else
        temp->next = lista;
    return lista;
}
```

```

        free(temp);
    }
    if (lista!=NULL)
    {
        lista->next = aggiungiLettura(lista->next, ora, minuto, secondo, locale, valore);
        return lista;
    }
    else{
        lista = (ptrLista)malloc(sizeof(node));
        lista->HH = ora;
        lista->MM = minuto;
        lista->SS = secondo;
        strcpy(lista->locale,locale);
        lista->gradi = valore;
        lista->next = NULL;
        return lista;
    }
}

ptrLista rimuoviLettture(ptrLista lista,int min, int max){
    ptrLista temp;
    ptrLista head;
    head = lista;
    temp = NULL;
    while (lista!=NULL){
        if ((lista->HH>=min) && (lista->HH<=max)){
            if (temp==NULL){
                temp = head;
                head = head->next;
                free(temp);
                temp = NULL;
            }
            else{
                temp->next = lista->next;
                temp = lista;
                free(lista);
            }
        }
        temp = lista;
        lista = lista->next;
    }
    return head;
}

void printLista(ptrLista lista)
{
    if (lista==NULL)
        return;
    else
    {
        printf("%d %d %d %s %d\n",lista->HH, lista->MM, lista->SS, lista->locale, lista->gradi);
        printLista(lista->next);
    }
}

int main(int argc, const char * argv[]) {
    int h,m,s,c=-1;
    char loc[20];

```

```
ptrLista lista = NULL;

while(c!=0)
{
    scanf("%d",&h);
    scanf("%d",&m);
    scanf("%d",&s);
    scanf("%s",loc);
    scanf("%d",&c);
    lista = aggiungiLettura(lista, h, m, s, loc, c);
    printLista(lista);
}
lista = rimuoviLettture(lista, 10,11);
printLista(lista);

return 0;
}
```





