



Politecnico di Milano

## Dipartimento di Elettronica, Informazione e Bioingegneria

**Informatica A - a.a 2018/2019 - 5 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	6 punti	_____
Esercizio 4	6 punti	_____
Esercizio 5	10 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 and B or C and not (A or B and not C)

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 and b) or (not a and c)

A	B	C	OUT
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
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  $A = 46_{dec}$  e  $B = -17$ , 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:**

$46_d = 101110_b - 6bit + 1 \text{ di segno} = 7bit$

$cp2 = 0101110_b$

$-17_d = 10001_b - 5 \text{ bit} + 1 \text{ segno} = 6bit$

$cp2 = 1101111_b - cp2 \text{ 7bit}$

A + B

riporto	11 111	<b>Riporto perduto ma non overflow</b>
A +	0101110b	
B	1101111b	
A+B	0011101b	

A - B

riporto		<b>Né riporto perduto, né overflow</b>
A -	0101110b	
B	0010001b	
A-B	0111111b	

- (c) Si converta il numero 28.6875 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:**

5bit + segno

28d = 011100b

$0.55 + 0.125 + 0.0625 = 0.1011$

37.59375 = 011100.1011 j- Virgola fissa

segno = 0

mantissa = 1.11001011 non normalizzata

exp = 4 + 127 = 131d = 10000011

IEEE754

segno (1bit) = 0

esponente (8bit) = 10000011

mantissa (23bit) = 110 0101 1000 0000 0000 0000

elimino il primo 1

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

Illustrare le caratteristiche principali della memoria del calcolatore. In particolare spiegare:

1. La differenza tra memoria centrale e memoria di massa

2. Che cosa sono la memoria RAM e la memoria ROM; che ruolo hanno in un sistema di elaborazione, che cosa le differenzia.

### Esercizio 3 - Esercizio C (6 punti)

Si scriva un sottoprogramma in linguaggio C che riceve in ingresso un array bidimensionale che contiene N espressioni, ognuna composta da numeri, parentesi e operatori +, -, x, /.

N viene passato alla funzione.

Il sottoprogramma deve salvare in un array bidimensionale di destinazione solamente le espressioni corrette, ovvero con le parentesi riportate correttamente (**numero di parentesi aperte uguale a numero di parentesi chiuse**). Per semplicità si assuma che le priorità tra le parentesi sia corrette (quindi non vi saranno espressioni del tipo "[2+( 1x3)]").

Il sottoprogramma deve anche restituire il numero di espressioni errate.

(Nell'esempio la funzione restituisce 1)

Array partenza:	Array destinazione:
$[(3-2) \times 4 + 3] \times (2+4) + (3-6)/4$	$[(3-2) \times 4 + 3] \times (2+4) + (3-6)/4$
$[(3-2) \times 4 + 3] \times (2+4)) + (3-6)/4$	$5+6-[8 \times 3]$
$5+6-[8 \times 3]$	$7 \times 8 - (3 \times 2)$
$7 \times 8 - (3 \times 2)$	

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>
#include <string.h>

int modificaArray(char array[200][200], char arrayDest[200][200], int len){
    int r,q,g;

    int count = 0;

    int i;
    int j;
    for (i=0; i<len; i++)
    {
        r = 0;
        q = 0;
        g = 0;

        for (j=0; array[i][j]!='\0'; j++)
        {
            switch (array[i][j])
            {
                case '(':
                    r=r+1;
                    break;
                case ')':
                    r=r-1;
                    break;
                case '[':
                    q=q+1;
                    break;
                case ']':
                    q=q-1;
                    break;
                case '{':
                    g=g+1;
                    break;
```

```

        case '}':
            g=g-1;
            break;
    }
    if ((g<0) || (q<0) || (r<0)){
        count = count+1;
        break;
    }
}
if ((g==0) && (q==0) && (r==0))
    strcpy(arrayDest[i-count],array[i]);

}
return count;
}

void printArray(char array[200][200], int len){
    for (int i=0; i<len; i++)
        printf("%s",array[i]);
}

int main(int argc, const char * argv[]) {
    char array[200][200];
    char arrayDest[200][200];
    int k;

    strcpy(array[0], "{[(3-2)x4+3]x(2+4)}+(3-6)/4\n");
    strcpy(array[1], "{[(3-2]x4+3]x(2+4)}+(3-6)/4\n");
    strcpy(array[2], "5+6-[8x3]\n");
    strcpy(array[3], "7x8-(3x2)\n");

    printArray(array, 4);

    k = modificaArray(array, arrayDest, 4);

    printf("\n\nArray destinazione: \n");
    printArray(arrayDest, 4-k);
    printf("\n\nEspressioni errate: %d\n",k);
    return 0;
}

```

#### 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 il numero di righe ed il numero di colonne, crea la matrice M(righe,colonne) contenente i primi N = (righe x colonne) numeri dispari(max 12 riga) (2 punto)

**Risposta:**

```
function out = luglio5_19(row,column)
    c = 1;
    for (i=1:row)
        for (j=1:column)
            out(i,j) = c;
            c = c+2;
        end
    end
end
```

- (b) Sostituire in una colonna a caso, tutti valori pari a uno (NB.: l'istruzione deve essere la più generale possibile, in modo da poter essere utilizzata con matrici di dimensioni differenti) (max 1 riga) (1 punto).

**Risposta:**

```
A(:,randi(size(A,2))) = ones(size(A,1),1)
A(:,randi(size(A,2))) = ones(1,size(A,1))
```

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

**Risposta:**

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



- (d) Moltiplicare tutte le celle per un numero casuale intero tra 5 e 18 (max 1 riga) (1 punto).

**Risposta:**

```
M = M * randi([5,18],1)
M = M * randi([5,18])
```

- (e) Moltiplicare per -1 tutte le celle che contengono numeri divisibili per 7 (max 1 riga) (1 punto).

**Risposta:**

```
M(rem(M,7)==0) = M(rem(M,7)==0)*-1
M(mod(M,7)==0) = M(mod(M,7)==0)*-1
```

### Esercizio 5 - Programmazione C Liste (10 punti)

Si vuole creare un programma per la gestione di software. Tutte i software disponibili sono contenuti in **una lista**. Ogni nodo della lista corrisponde ad un software ed ha come attributi:

- nome software
- versione
- software house
- prezzo

es.

- nome software: Winzoz9
- versione: 5
- software house: Microfrost
- prezzo: 10

1. Definire la struttura dati necessaria per la realizzazione del programma. (1 punto)

2. Scrivere la funzione **RICORSIVA** che ricerca un software all'interno della lista. La funzione riceve in ingresso il nome del software e la versione e:

- Se il nome del software non è presente nella lista dei software, la funzione restituisce -1;
- Se il nome del software è presente, ma non lo è la versione, restituisce -2.
- Se software e versione sono presenti, restituisce 0 e il node contenente il software.

(3 punti)

3. Utilizzando la funzione precedente, scrivere una funzione che presi in ingresso la lista completa dei software e due array contenenti nome software e versione (lunghezza massima dell'array 10 - alla funzione va passata la lunghezza dell'array), crea una nuova lista con i software elencati nell'array. (3 punti)

4. Scrivere una funzione **RICORSIVA** che presi in ingresso la lista e il nome di un software, ritorni quante versioni del software sono state rilasciate.

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

#### Risposta:

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

//Punto 1 -----
typedef struct nodo{
    char nomeSoftware[20];
    int versione;
    char softwareHouse[20];
    int prezzo;
    struct nodo *next;
} nodo;

typedef nodo *ptrNode;
//-----
```

```

void stampaLista(ptrNode lista)
{
    if (lista == NULL)
        return;
    printf("Nome software: %s\nVersione: %d\nSoftware house: %s\nPrezzo: %d\n\n",
        lista->nomeSoftware, lista->versione, lista->softwareHouse, lista->prezzo);
    stampaLista(lista->next);
}

ptrNode inserisci(ptrNode lista, char nomeSoftware[20], int versione, char softwareHouse[20],
int prezzo)
{
    ptrNode nuovo;
    nuovo = (ptrNode)malloc(sizeof(nodo));

    strcpy(nuovo->nomeSoftware, nomeSoftware);
    nuovo->versione=versione;
    strcpy(nuovo->softwareHouse, softwareHouse);
    nuovo->prezzo = prezzo;

    nuovo->next = lista;
    return nuovo;
}

//Punto 2: restituisce un nodo nuovo, contenente i dati del software ricercato.
//La malloc viene fatta nel main.
int ricerca(ptrNode lista, ptrNode nodoTrovato, int nome, char nomeSoftware[20], int versione)
{
    if (lista == NULL){
        nodoTrovato = NULL;
        if (nome==0)
            return -1;
        else
            return -2;
    }
    if (strcmp(lista->nomeSoftware, nomeSoftware)==0) //trovato nome almeno una volta
    {
        if (lista->versione == versione) //trovato software corretto
        {
            strcpy(nodoTrovato->nomeSoftware, lista->nomeSoftware);
            nodoTrovato->versione = lista->versione;
            strcpy(nodoTrovato->softwareHouse, lista->softwareHouse);
            nodoTrovato->prezzo = lista->prezzo;
            nodoTrovato->next = NULL;
            return 0;
        }
        else
            return ricerca(lista->next, nodoTrovato, 1, nomeSoftware, versione);
    }
    else
        return ricerca(lista->next, nodoTrovato, 0, nomeSoftware, versione);
}

//Punto 2: restituisce il puntatore al software ricercato. Non pu essere usato direttamente per
//creare la nuova lista, altrimenti si modificherebbe anche la lista di partenza.
int ricerca2(ptrNode lista, ptrNode *nodoTrovato, int nome, char nomeSoftware[20], int versione)
{
    if (lista == NULL){
        *nodoTrovato = NULL;

```

```

        if (nome==0)
            return -1;
        else
            return -2;
    }
    if (strcmp(lista->nomeSoftware,nomeSoftware)==0) //trovato nome almeno una volta
    {
        if (lista->versione == versione) //trovato software corretto
        {
            *nodoTrovato = lista;
            return 0;
        }
        else
            return ricerca2(lista->next,nodoTrovato,1,nomeSoftware,versione);
    }
    else
        return ricerca2(lista->next,nodoTrovato,0,nomeSoftware,versione);
}

//Punto 3, con funzione che restituisce un nuovo nodo, contenente i dati del software trovato
ptrNode creaLista(ptrNode listaSoftware, char arrayNomi[10][20], int arrayVersioni[10], int len){
    int i = 0;
    ptrNode trovato;
    ptrNode lista = NULL;

    if (listaSoftware == NULL)
        return NULL;

    for (i = 0; i<len; i++)
    {
        trovato = (ptrNode)malloc(sizeof(nodo));
        if (ricerca(listaSoftware, trovato, 0, arrayNomi[i], arrayVersioni[i])==0)
        {
            trovato->next = lista;
            lista = trovato;
        }
    }
    return lista;
}

//Punto 3, con funzione che restituisce il puntatore al software trovato.
ptrNode creaLista2(ptrNode listaSoftware, char arrayNomi[10][20], int arrayVersioni[10], int len){
    int i = 0;
    ptrNode trovato = NULL;
    ptrNode lista = NULL;
    ptrNode temp = NULL;

    if (listaSoftware == NULL)
        return NULL;

    for (i = 0; i<len; i++)
    {
        if (ricerca2(listaSoftware, &trovato, 0, arrayNomi[i], arrayVersioni[i])==0)
        {
            temp = (ptrNode)malloc(sizeof(nodo));
            strcpy(temp->nomeSoftware,trovato->nomeSoftware);
            temp->versione = trovato->versione;
            temp->prezzo = trovato->prezzo;

```

```

        strcpy(temp->softwareHouse,trovato->softwareHouse);
        temp->next = lista;
        lista = temp;
    }
}
return lista;
}

int contaVersioni(ptrNode listaSoftware, char nomeSoftware[20])
{
    if (listaSoftware == NULL)
        return 0;
    else
    {
        if (strcmp(listaSoftware->nomeSoftware,nomeSoftware)==0)
            return 1+contaVersioni(listaSoftware->next, nomeSoftware);
        else
            return contaVersioni(listaSoftware->next, nomeSoftware);
    }
}

int main(int argc, const char * argv[]) {
    ptrNode listaSoftware = NULL;
    ptrNode mysoftware = NULL;

    ptrNode trovato;

    char nomeSoftware[20];
    int versione;

    char softwares[10][20];
    int versions[10];

    strcpy(softwares[0],"winzoz9");
    strcpy(softwares[1],"winzoz9");
    strcpy(softwares[2],"winzoz9");
    strcpy(softwares[3],"winzoz9");
    versions[0] = 3;
    versions[1] = 1;
    versions[2] = 5;
    versions[3] = 11;

    int r;
    int k;

    listaSoftware = inserisci(listaSoftware,"winzoz9",3,"Microfrost",15);
    listaSoftware = inserisci(listaSoftware,"winzoz9",2,"Microfrost",10);
    listaSoftware = inserisci(listaSoftware,"winzoz9",1,"Microfrost",10);
    listaSoftware = inserisci(listaSoftware,"winzoz9",4,"Microfrost",10);
    listaSoftware = inserisci(listaSoftware,"maxos",12,"Pineapple",10);
    listaSoftware = inserisci(listaSoftware,"winzoz9",5,"Microfrost",100);
    listaSoftware = inserisci(listaSoftware,"maxos",10,"Pineapple",0);
    listaSoftware = inserisci(listaSoftware,"maxos",5,"Pineapple",100);

    stampaLista(listaSoftware);

    do{
        printf("MENU\n");
        printf("1) Ricerca\n");

```

```

printf("2) Ricerca con puntatore\n");
printf("3) Crea nuova lista\n");
printf("4) Conta versioni\n");
printf("5) ESCI\n\n>> \n");
scanf("%d",&r);
switch (r){
    case 1:
        //RICERCA
        printf("Nome software: ");
        scanf("%s",nomeSoftware);
        printf("Versione: ");
        scanf("%d",&versione);

        trovato = (ptrNode)malloc(sizeof(nodo));

        k = ricerca(listaSoftware, trovato, 0, nomeSoftware, versione);
        if (k==0)
        {
            printf("Software trovato\n\n");
            printf("Nome software: %s\nVersione: %d\nSoftware house: %s\nPrezzo: %d\n\n",
                trovato->nomeSoftware,trovato->versione,trovato->softwareHouse,trovato->prezzo);
        }
        else if (k==-1)
            printf("Software non nella lista\n\n");
        else
            printf("Esiste un'altra versione del software ma non quella ricercata\n\n");
        break;

    case 2:
        printf("Nome software: ");
        scanf("%s",nomeSoftware);
        printf("Versione: ");
        scanf("%d",&versione);
        k = ricerca2(listaSoftware, &trovato, 0, nomeSoftware, versione);
        if (k==0)
        {
            printf("Software trovato\n\n");
            printf("Nome software: %s\nVersione: %d\nSoftware house: %s\nPrezzo: %d\n\n",
                trovato->nomeSoftware,trovato->versione,trovato->softwareHouse,trovato->prezzo);
        }
        else if (k==-1)
            printf("Software non nella lista\n\n");
        else
            printf("Esiste un'altra versione del software ma non quella ricercata\n\n");
        break;

    case 3:
        //CREA LISTA
        mysoftware = creaLista(listaSoftware, softwares, versions, 4);
        //mysoftware = creaLista2(listaSoftware, softwares, versions, 4);
        stampaLista(mysoftware);
        break;

    case 4:
        //CONTA VERSIONI
        printf("Numero di versioni di Winzoz9: %d\n",contaVersioni(listaSoftware, "winzoz9"));
        break;

}
}while(r!=5);

```

}











Politecnico di Milano

## Dipartimento di Elettronica, Informazione e Bioingegneria

**Informatica A - a.a 2019/2020 - 14 Gennaio 2020**

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 not C) and C OR not B and C

**Risposta:**

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

OUT = AC + !BC

- (b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri A = -128dec e B = 63, 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:**

128d = 1000 0000b - 8bit

-128 cp2 = 1000 0000b - 8bit.

63d = 011 1111b - 6 bit + 1 segno

-63 cp2 = 100 0001b - cp2 7bit

A + B = -128 + 63 = -65 = 1011 1111b - No riporto perduto né overflow

A - B = -128 - 63 = -192 = [1] 0100 0001 - Riporto perduto con overflow

**NOME e COGNOME:** \_\_\_\_\_

8 bit va da -128 a 127

- (c) Si converta il numero 43.625 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

43d = 010 1011b

$0.5 + 0.125 = 0.101b$

43.625 = 0101011.101 j- Virgola fissa

segno = 0

mantissa = 1.01011101 non normalizzata

exp = 5 + 127 = 132d = 10000100

IEEE754

segno (1bit) = 0

esponente (8bit) = 10000100

mantissa (23bit) = 010 1110 1000 0000 0000 0000

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

- (a) Spiegare cosa si intende per allocazione dinamica della memoria. Quando è necessaria? Riportare un esempio di allocazione dinamica (ANSI C) per un array con dimensione N (N letto mediante una scanf).

- (b) Cosa sono i dangling reference (puntatori ciondolanti)? Riportarne un esempio.

**Risposta:**

**Esercizio 3 - Comprensione del Codice (4 punti)**

Scrivere il valore di n e cosa stampa a video il seguente codice:

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

int doSomething(char *a, char b);
int main(void) {
    char *a;
    char b;

    int n = 10;
    a = (char*)malloc(sizeof(char)*n);

    strcpy(a, " "); //INSERIRE QUI IL CODICE PERSONA o LA MATRICOLA
    b = a[strlen(a)];
    b = (int)doSomething(a,b);
    printf("%s %c\n",a,(b+'a'));
    return 0;
}

char doSomethingElse(char *c, char d){
    if (*c == d)
        return 0;
    else
        return 1+doSomethingElse(++c, d);
}

int doSomething(char *a, char b){
    int n;
    int val = 0;
    int i;
    n = (int) doSomethingElse(a,b);
    for (i=1;*a!= b; a++, n--, i++){
        val += (*(a+n-i)-'0');
        printf("%c", (val%26)+'a');
    }
    printf("\n");
    return val%26;
}
```

n: \_\_\_\_\_

printf:

---

---

---

---

---

---

---

---

---

---



**NOME e COGNOME:** \_\_\_\_\_

**Risposta:**

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

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

- (a) Creare un matrice A di dimensioni righe e colonne differenti e casuali da 5 a 10 e contenente solo numeri 4; (max 1 riga) (1 punto)

**Risposta:**

```
A = 4*ones(randi([5 10]), randi([5 10]))
```

- (b) Inserire dopo la prima colonna, una nuova colonna contenente numeri casuali da 0 a NxM dove N numero di righe e M numero di colonne. (max 1 riga) (2 punti)

**Risposta:**

```
A= [A(:,1) randi([0 size(A,1)*size(A,2)], [size(A,1) 1]) A(:,2:end)]
```

- (c) Eliminare le righe con media inferiore a 6 (max 1 riga) (1 punto).

**Risposta:**

```
A(mean(A,2)<6,:) = []
```

- (d) Calcolare la somma totale di tutte le colonne pari (max 1 riga) (1 punto).

**Risposta:**

```
sum(sum(A(:,2:2:end)))
```

- (e) Scrivere una funzione che presa in ingresso la matrice A, crea una nuova matrice "quadrata", ovvero elimina le righe o le colonne in eccesso in modo da ottenere una matrice con numero righe = numero di colonne (max 8 righe compreso end) (1 punto).

**Risposta:**

```
function quadrata = test(B)  
i = min(size(B));
```

```
    for j=1:1:i
        for k=1:1:i
            quadrata(k,j) = B(k,j);
        end
    end
end

oppure

function quadrata = test(B)
    i = min(size(B));
    quadrata = B(1:i,1:i);
end
```

## Esercizio 5 - Programmazione C (12 punti)

Si vuole creare un programma per la gestione della tombola. La tombola è caratterizzata attraverso una struttura dati DINAMICA contenente i numeri estratti. Ogni numero estratto è caratterizzato dal numero stesso e da un vettore di 5 elementi contenete le vincite per "ambo", "terno", "quaterna", "cinquina", "tombola").

Es. vettore = 1 0 2 1 0 Significa che con quel numero sono stati vinti: 1 ambo, 0 terni, 2 quaterne, 1 cinquina e 0 tombole.

Le cartelle della tombola sono composte da una matrice 5x3 di numeri interi.

Tramite un'altra struttura dati, ad ogni numero della tombola (da 1 a 90) è associato una descrizione estratta dalla smorfia napoletana.

1. Definire le strutture dati necessari per la realizzazione del programma. (1 punto)
2. Scrivere una funzione RICORSIVA che calcoli quante vincite sono state fatte fino ad ora (2 punti)
3. Scrivere una funzione per l'inserimento in coda di un nuovo numero estratto. Il numero estratto deve essere casuale da 1 a 90 ma non ancora estratto. La funzione restituisce 0 se è stato possibile estrarre il numero, 1 altrimenti (sono già stati estratti 90 numeri). La funzione deve inoltre restituire il significato della smorfia relativo all'ultimo numero estratto. (4 punti)
4. Scrivere una funzione che presa in ingresso la lista di numeri estratti e una cartella della tombola, aggiorni l'ultimo elemento della lista con le eventuali vincite sulla cartella (si consideri solamente la vittoria più importante). (5 punti)

### Risposta:

```

typedef struct node{
    int numero;
    int vincite[5];
    struct nodo *next;
} node;

typedef node* ptrNode;

char smorfia[90][100];
int cartella[3][5];
ptrNode tombola;

int contaVittorie(ptrNode lista){
    if (lista == NULL)
        return 0;
    int i, val=0;
    for (i=0; i<5; i++)
        val += lista->vincite[i];
    return val + contaVittorie(lista->next);
}

ptrNode inserisciInCoda(ptrNode lista, char smorfia[][100], char stringa[100], int *ris)
{
    int numero;
    int trovato;
    ptrNode testa = lista;
  
```

```

ptrNode temp;
int i = 0;
do{
    i++;
    trovato = 0;
    lista = testa;
    numero = rand() % 90 + 1;
    while ((lista!=NULL) && (trovato==0)){
        if (lista->numero==numero)
            trovato = 1;
    }
}while((trovato==1) && (i<90));
if (i>=90)
{
    *ris =1;
    return testa;
}

lista = testa;
strcpy(stringa,smorfia[numero]);
temp = (ptrNode)malloc(sizeof(node));
temp->numero = numero;
temp->next = NULL;

*ris = 0;

if (lista == NULL)
    return temp;
else
{
    while(lista->next!=NULL)
        lista = lista->next;
    lista->next = temp;
    return testa;
}
}

void aggiornaTombola(ptrNode tombola, int cartella[3][5])
{
    int i,j;
    int num=0;
    int max = 0;
    int cont[3];
    for (j=0; j<3; j++)
        cont[j] = 0;

    if (tombola==NULL)
        return;

    while (tombola->next!=NULL)
    {
        for (i=0; i<3; i++)
        {
            for (j=0; j<5; j++){
                if (tombola->numero == cartella[i][j]) {
                    cont[i]++;
                    break;
                }
            }
        }
    }
}

```

```
tombola = tombola->next;
}
if (cont[0]+cont[1]+cont[2]==15)
    tombola->vincite[4]++;
else{
    if ((cont[0]>cont[1]) && (cont[0]>cont[2]))
        max = cont[0];
    else if ((cont[1]>cont[0]) && (cont[1]>cont[2]))
        max = cont[1];
    else
        max = cont[2];

    if (max>=2)
    {
        tombola->vincite[max-2]++;
    }
}
}
```

**NOME e COGNOME:** \_\_\_\_\_

**NOME e COGNOME:** \_\_\_\_\_





Politecnico di Milano

## Dipartimento di Elettronica, Informazione e Bioingegneria

**Informatica A - a.a 2018/2019 - 28 Gennaio 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	_____

**Totale(28)** \_\_\_\_\_

**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 and B or not C) and C or not B

**Risposta:**

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

OUT = ABC + !B

- (b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri A = -64dec e B = 33, 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:**

-64d = 1000000b - 6bit + 1 di segno = 7bit

33d = 0100001b - 6 bit + 1 segno

cp2 = 1011111b - cp2 7bit

A + B = -64 + 33 = -31 = 1100001b - No riporto perduto né overflow

A - B = -64 - 33 = -97 = [1]0011111b - Riporto perduto con overflow

**NOME e COGNOME:** \_\_\_\_\_

7 bit va da -64 a 63

- (c) Si converta il numero 37.59375 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 = 0100101

$0.5 + 0.0625 + 0.03125 = 0.10011$

37.59375 = 0100101.10011 j- Virgola fissa

segno = 0

mantissa = 1.0010110011 non normalizzata

exp = 5 + 127 = 132d = 10000100

IEEE754

segno (1bit) = 0

esponente (8bit) = 10000100

mantissa (23bit) = 100 1011 0011 0000 0000 0000

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

- (a) Spiegare che cosa si intende per iterazione e per ricorsione. Quando è necessario, o conveniente, usarle? In generale è più efficiente l'iterazione o la ricorsione?

- (b) Implementare una semplice funzione di esempio sia in versione iterativa che in versione ricorsiva.

**Risposta:**

**Esercizio 3 - Comprensione del Codice (4 punti)**

Scrivere cosa stampa a video il seguente codice:

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

char* doSomething(char *a, char *b);
int main(void) {
    char *a;
    char *b;

    int n = 7;
    a = (char*)malloc(sizeof(char)*n);
    strcpy(a,"casale");
    b = doSomething(a,b);
    printf("%s %c\n",a,*b);
    return 0;
}

char* doSomething(char *a, char *b){
    int n = 0;
    char c;
    b=a;
    for (b,c='1',n=1; b<&a[strlen(a)]; b++,c++)
    {
        if ((*b)=='a')
        {
            n++;
            *b = c;
        }
        else if ((*b)=='a'+4)
        {
            n++;
            *b = c-'1'+3;
        }
        printf("%s %c\n",a,*b);
    }
    b = (char*)malloc(sizeof(char));
    *b = ('0'+n);
    return b;
}
```

printf:

_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

NOME e COGNOME: \_\_\_\_\_

**Risposta:**

casale c c2sale 2 c2sale s c2s4le 4 c2s4le l c2s4l8 8 c2s4l8 4

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

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

- (a) Creare un vettore colonna A e un vettore colonna B contenente rispettivamente 6 e 5 valori casuali differenti tra loro con valori da 1 a 10; (max 2 riga) (1 punto)

**Risposta:**

```
A = randi([1 10],[6 1]);  
B = randi([1 10],[1 5]);
```

oppure

```
A = randi([1 10],[6 1]);  
B = randi([1 10],[5 1]);
```

- (b) A partire dai vettori A e B, creare una matrice C di 6 righe e 5 colonne (1 punto).

**Risposta:**

```
Se una colonna e una riga  
C = A*B;  
C = A.*B;
```

Se entrambe colonne:

```
C = A*B';  
C = A.*B';
```

- (c) Cancellare le righe che hanno come primo valore un numero inferiore a 10 (max 1 riga) (1 punto).

**Risposta:**

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

- (d) Calcolare la somma di tutte le colonne dispari (max 1 riga) (1 punto).

**Risposta:**

```
sum(sum(C(:,1:2:end)))
```



- (e) Scrivere una funzione che presa in ingresso la matrice C, crea una nuova matrice "cornice", ponendo cioè a zero tutti gli elementi non appartenenti alla prima e ultima riga/colonna. (max 8 righe) (2 punto).

**Risposta:**

```
function [out] = funzione(in)
    out = in;
    for i=2:1:size(in,1)-1
        for j=2:1:size(in,2)-1
            out(i,j) = 0;
        end
    end
end
```

## Esercizio 5 - Programmazione C (12 punti)

Si vuole creare un programma per trasformare una serie di immagini in una GIF animata.

Un'immagine è definita da una struttura che contiene tre vettori bidimensionali 100x100 contenete valori che vanno da 0 a 255. Ogni matrice corrisponde al colore rosso, verde e blue.

Una GIF animata è costituita da una lista i cui elementi sono composti da un'immagine (i tre vettori bidimensionali) ed un numero che corrisponde ai secondi per cui l'immagine viene visualizzata durante l'animazione.

1. Definire le strutture dati necessari per la realizzazione del programma. (1 punto)
2. Scrivere una funzione che prende in ingresso due immagini e controlli se differiscono di almeni N pixel. N viene definito dall'utente ed è un parametro che viene passato alla funzione. I pixel differiscono quando i valori dei vettori bidimensionali nella medesima posizione sono differenti (es. `immRed1[10][10] != immRed2[10][10]`). Se le immagini differiscono la funzione ritorna 1, se non differiscono restituisce 0. (2 punti)
3. Scrivere una funzione che presa in ingresso un'immagine e un tempo t, la aggiunge in coda alla lista contenente tutte le immagini della GIF animata. Prima di aggiugnerle deve controllare che l'immagine differisca di almeno 10 pixel dall'immagine precedente (Si consiglia di usare la funzione del punto 2). (3 punti)
4. Si vuole riprodurre la GIF al contrario. Scrivere la funzione che presa in ingresso la lista della GIF, crei una nuova lista inserendo gli elementi al contrario. (3 punti)
5. Scrivere una funzione che presa in ingresso la GIF ed una variabile tempo T, elimini dalla lista tutte le immagini che hanno durata inferiore a T. Non deve essere creata una nuova lista, ma vanno eliminati gli elementi dalla lista di partenza. (3 punti)

### Risposta:

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

#define M 5

typedef struct immagine{
    int red[M][M];
    int green[M][M];
    int blue[M][M];
} immagine;

typedef struct imgGif{
    immagine img;
    int tempo;
    struct imgGif *next;
} imgGif;

typedef imgGif* ptrGif;

//Funzioni
void generaMatrici(int matrix[][M], int i);
void stampaMatrice(int matrix[][M]);
immagine generaImmagine(int i);
void stampaGif(ptrGif gif);
```

```

int confrontaImmagini(immagine img1, immagine img2, int n);
ptrGif aggiungiGif(ptrGif gif, immagine img, int tempo);
ptrGif contrarioIt(ptrGif gif);
ptrGif contrario(ptrGif gif, ptrGif contrario);
ptrGif velocizza(ptrGif gif, T);

int main(void) {
    ptrGif gif = NULL;
    ptrGif gifContrario = NULL;

    immagine img1;
    immagine img2;
    int c;
    int r;
    int val;

    srand((unsigned)time(0));

    do{
        printf("\n\nMENU\n");
        printf("L'opzione 2 genera una matrice 5x5 e non 100x100 per velocizzare il tutto.
            Il comando genera una matrice uguale o una casuale (per cui diversa)\n\n");
        printf("1) Confronta immagini\n2) Inserisci immagini\n3) Lista al contrario\n4)
            Velocizza GIF\n5) Stampa gif\n6) ESCI\n\n>> ");
        fpurge(stdin);
        scanf("%d",&c);
        switch (c) {
            case 1:
                //Per confrontare immagine differenti mettere 0 e 0 o 0 e 1 nella generazione
                //Per confrontare immagine uguali mettere 1 e 1
                img1 = generaImmagine(0);
                img2 = generaImmagine(1);
                if (confrontaImmagini(img1, img2, 5))
                    printf("Le immagini sono differenti\n");
                else
                    printf("Le immagini non sono differenti\n");

                break;
            case 2:
                printf("Uguale 0, diversa 1: ");
                scanf("%d",&r);
                img1 = generaImmagine(r);
                printf("Inserisci tempo: ");
                scanf("%d",&r);
                gif = aggiungiGif(gif, img1, r);
                break;

            case 3:
                stampaGif(gif);
                printf("\n\nAl contrario: \n\n");
                //gifContrario = contrario(gif, gifContrario);
                gifContrario = contrarioIt(gif);
                stampaGif(gifContrario);
                break;

            case 4:
                printf("Inserire valore minimo tempo immagine: ");
                scanf("%d",&val);
                gif = velocizza(gif, val);

```

```

        break;

        case 5:
            stampaGif(gif);
        default:
            break;
    }
}while(c!=6);
return 0;
}

void generaMatrici(int matrix[][M], int k)
{
    int i,j;
    for (i=0; i<M; i++)
    {
        for (j=0; j<M; j++)
        {
            if (k!=0)
                matrix[i][j]=rand()%256;
            else
                matrix[i][j]=100;
        }
    }
}

void stampaMatrice(int matrix[][M])
{
    int i,j;
    for (i=0; i<M; i++)
    {
        for (j=0; j<M; j++)
        {
            printf("%d ",matrix[i][j]);
        }
        printf("\n");
    }
}

void stampaGif(ptrGif gif)
{
    if (gif==NULL)
        return;
    stampaMatrice(gif->img.blue);
    //stampaMatrice(gif->img.green);
    //stampaMatrice(gif->img.red);
    printf("%d\n",gif->tempo);
    stampaGif(gif->next);
}

immagine generaImmagine(int i){
    immagine img;
    generaMatrici(img.blue, i);
    generaMatrici(img.red, i);
    generaMatrici(img.green, i);
    return img;
}

int confrontaImmagini(immagine img1, immagine img2, int n)

```

```

{
    int i, j;
    int cont = 0;

    for (i=0; i<M; i++)
    {
        for (j=0; j<M; j++)
        {
            if ((img1.blue[i][j] != img2.blue[i][j]) || (img1.red[i][j] != img2.red[i][j])
                || (img1.green[i][j] != img2.green[i][j]))
                cont++;
            // if (img1.blue[i][j] != img2.blue[i][j])
            // cont++;
            // else if (img1.red[i][j] != img2.red[i][j])
            // cont++;
            // else if (img1.green[i][j] != img2.green[i][j])
            // cont++;
        }
    }
    if (cont>n)
        return 1;
    else
        return 0;
}

ptrGif aggiungiGif(ptrGif gif, immagine img, int tempo)
{
    //int i,j;
    if (gif==NULL)
    {
        gif = (ptrGif)malloc(sizeof(imgGif));
    }
    // for (i=0; i<M; i++)
    // {
    // for (j=0; j<M; j++){
    // gif->img.red[i][j] = img.red[i][j];
    // gif->img.blue[i][j] = img.blue[i][j];
    // gif->img.green[i][j] = img.green[i][j];
    // }
    // }
    gif->img = img;
    gif->tempo = tempo;
    gif->next = NULL;
    return gif;
}

if ((gif->next!=NULL) || ((gif->next == NULL) && (confrontaImmagini(gif->img,img,10))))
    gif->next = aggiungiGif(gif->next, img, tempo);
return gif;
}

ptrGif contrarioIt(ptrGif gif)
{
    ptrGif contr = NULL;
    ptrGif temp = NULL;

    while (gif!=NULL){
        temp = (ptrGif)malloc(sizeof(imgGif));
        temp->img = gif->img;
        temp->tempo = gif->tempo;
        temp->next = contr;
    }
}

```

```
        contr = temp;
        gif = gif->next;
    }
    return contr;
}

ptrGif contrario(ptrGif gif, ptrGif contr){
    ptrGif temp;
    int i,j;

    if (gif==NULL)
        return contr;

    temp = (ptrGif)malloc(sizeof(imgGif));
    for (i = 0; i<M; i++)
    {
        for (j=0; j<M; j++)
        {
            temp->img.blue[i][j] = gif->img.blue[i][j];
            temp->img.red[i][j] = gif->img.red[i][j];
            temp->img.green[i][j] = gif->img.green[i][j];
        }
    }
    temp->tempo = gif->tempo;
    temp->next = contr;
    temp = contrario(gif->next, temp);
    return temp;
}

ptrGif velocizza(ptrGif gif, int T){
    ptrGif temp;
    if (gif==NULL)
        return NULL;
    if (gif->tempo<T)
    {
        temp = gif->next;
        free(gif);
        temp = velocizza(temp, T);
        return temp;
    }
    else
    {
        gif->next = velocizza(gif->next, T);
        return gif;
    }
}
```

**NOME e COGNOME:** \_\_\_\_\_

**NOME e COGNOME:** \_\_\_\_\_





Politecnico di Milano

## Dipartimento di Elettronica, Informazione e Bioingegneria

**Informatica A - a.a 2017/2018 - 30 Agosto 2018**

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:

<b>Esercizio 1</b>	<b>3 punti</b>	_____
<b>Esercizio 2</b>	<b>3 punti</b>	_____
<b>Esercizio 3</b>	<b>5 punti</b>	_____
<b>Esercizio 4</b>	<b>6 punti</b>	_____
<b>Esercizio 5</b>	<b>11 punti</b>	_____
<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) and A or !C

**Risposta:**

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

OUT = A + !C

- (b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri A = -33dec e B = 95, 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:**

95d = 01011111b - 7bit + 1 di segno = 8bit

33d = 0100001b - 6 bit + 1 segno

cp2 = 1011111b - cp2 8bit = 11011111

A + B = -33 + 95 = 62 = 00111110b con riporto perduto

**NOME e COGNOME:** \_\_\_\_\_

A-B = -128 = 10000000b c'è riporto perduto;  
8 bit va da -128 a 127

- (c) Si converta il numero 25.3125 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:**

5bit + segno

25d = 011001

0.3125 =  $1/4 + 1/16 = 0.0101b$

25.3125 = 011001.0101 j- Virgola fissa

segno = 0

mantissa = 1.10010101 non normalizzata

exp =  $4 + 127 = 131d = 10000011b$

IEEE754

segno (1bit) = 0

esponente (8bit) = 10000011

mantissa (23bit) = 100 1010 1000 0000 0000 0000

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

- (a) Che differenza c'è tra una struttura dati statica ed una struttura dati dinamica? Perché vengono usate? Quali sono i vantaggi/svantaggi?

**Risposta:**

- (b) Fare un esempio di implementazione di struttura dati statica e dinamica per la memorizzazione di numeri interi, riportando per ognuna la funzione di inserimento di un nuovo dato.

**Risposta:**

### Esercizio 3 - Comprensione del Codice (5 punti)

Scrivere cosa stampa a video il seguente codice:

```
#include <stdio.h>
float doSomething(int val, int d);

int main(){
    int a;
    int b[10];
    int *c;

    a = 2;
    c = b;

    for (c; c<(&b[0])+10; c++)
    {
        *c = (3*(c-b));
    }

    for (int i=0; i<10; i++)
    {
        printf("%d \t %.1f\n",b[i],doSomething(b[i], a));
    }
    return 0;
}

float doSomething(int val, int d){
    if (val%9==0)
        return (float)(val/d);
    else
        return (val/(float)d);
}
```

printf:

_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

**Risposta:**

0 0.0  
3 1.5  
6 3.0

**NOME e COGNOME:** \_\_\_\_\_

9 4.0  
12 6.0  
15 7.5  
18 9.0  
21 10.5  
24 12.0  
27 13.0

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

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

- (a) Creare la matrice A di 5 righe e 6 colonne contenente numeri interi casuali da 10 a 30 (max 1 riga, 1 punto)

**Risposta:**

```
A = randi([10 30], 5,6)
```

- (b) Aggiungere una riga dopo la prima, contenente numeri i ordine da 1 a 6 (max 1 riga) (1 punto)

**Risposta:**

```
A = [A(1,:); [1:6]; A(2:end,:)]
```

oppure

```
A = vertcat(A(1,:), [1:6], A(2:end,:))
```

- (c) Eliminare le colonne con media inferiore alla media totale della matrice (max 1 riga) (1 punto).

**Risposta:**

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

- (d) Calcolare il vettore Z contenente le radici quadrate delle medie delle righe (max 1 riga) (1 punto).

**Risposta:**

```
Z = sqrt(mean(A,2))
```

- (e) Scrivere una funzione che, data in ingresso la matrice A, dia in uscita due vettori contenenti la posizione X e Y dei valori pari della matrice A. (max 12 righe) (2 punti)



**Risposta:**

```
function [X,Y] = trova(A)
X = []
Y = []
for i=1:1:size(A,1)
    for j=1:1:size(A,2)
        if (mod(A(i,j),2)==0)
            X = [X, i];
            Y = [Y, j];
        end
    end
end
end
end
```

### Esercizio 5 - Programmazione C (10 punti)

Si ipotizzi di volere scrivere un programma C per la gestione di un negozio online di alimentari etnici. Il programma utilizza due strutture dati dinamiche.

Nella prima struttura viene immagazzinata mediante una lista l'anagrafica della clientela. Ciascun cliente è dotato di un nome, un cognome, un codice fiscale (alfanumerico), e una data di nascita. All'interno di questa lista i clienti sono ordinati alfabeticamente per cognome e poi per nome. Se due persone si chiamano allo stesso modo sono ordinati per codice fiscale (univoco).

Nella seconda struttura viene immagazzinato mediante una coda lo storico degli acquisti fatti attraverso il sistema. Ciascun acquisto è caratterizzato da un codice univoco, una data di acquisto, il codice fiscale del cliente che ha effettuato l'acquisto, da una descrizione dei contenuti dell'ordine da un totale in euro.

Si definiscano le strutture dati necessarie ad implementare le due liste del programma. (1 punto)

Si implementi una funzione che dati due clienti li confronta e dice quale dei due deve venire prima nell'anagrafica. La funzione restituisce -1 se il primo cliente deve venire prima del secondo, 0 se i due clienti sono identici (e quindi hanno lo stesso codice fiscale), oppure 1 se il primo cliente deve venire dopo il secondo. (1 punto)

Si implementi una funzione di inserimento ordinato di un cliente all'interno dell'anagrafica. La funzione deve restituire 1 se l'inserimento è andato a buon fine, oppure 0 se c'è stato un problema (ad es. il cliente esisteva già). (2 punti)

Si implementi le funzioni di push e pop per la coda (first in - first out) contenente lo storico degli acquisti fatti. (2 punti)

Implementare una funzione che chiede all'utente (mediante printf e scanf) di inserire il nome e il cognome di un cliente. Se ci sono più clienti con quel nome e cognome la funzione deve stampare a video un elenco dei codici fiscali di questi clienti e chiedere all'utente (sempre mediante printf e scanf) di sceglierne uno. A quel punto la funzione deve procedere a stampare a video (i) il numero di acquisti fatti da quel cliente nell'anno solare 2018 e (ii) il totale in euro speso dal cliente nell'anno solare 2018. Deve inoltre stampare a video, per ciascun acquisto fatto nel 2018, la data di acquisto, la descrizione dell'acquisto e il totale dell'acquisto. (4 punti)

**Risposta:**

**NOME e COGNOME:** \_\_\_\_\_

**NOME e COGNOME:** \_\_\_\_\_



Politecnico di Milano

## Dipartimento di Elettronica, Informazione e Bioingegneria

**Informatica A - a.a 2018/2019 - 30 Agosto 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 and B or not C and (A or 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:**  $A + B$

A	B	C	OUT
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
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  $A = 23_{\text{dec}}$  e  $B = 72$ , 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:**

Servono 8 bit: da -128 a 127  $23_{\text{d}} = 10111_{\text{b}}$

$cp2 = 010111_{\text{b}}$

$72_{\text{d}} = 01001000_{\text{b}}$

$-72_{\text{d}} cp2 = 10111000_{\text{b}}$

A + B

riporto		
A +	00010111b	<b>né riporto perduto né overflow</b>
B	01001000b	
A+B	01011111b	

A - B

riporto	11	
A -	00010111b	<b>Nè riporto perduto, né overflow</b>
B	10111000b	
A-B	11001111b	

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

- (c) Si converta il numero 29.625 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:**

5bit + segno

29d = 11101b

$0.5 + 0.125 = 0.101$

37.625 = 011101.101 Virgola fissa

Virgola mobile

segno = 0

mantissa = 1.1101101 non normalizzata

exp = 4 + 127 = 131d = 10000011

IEEE754

segno (1bit) = 0

esponente (8bit) = 10000011

mantissa (23bit) = 110 1101 0000 0000 0000 0000 - normalizzato eliminando il primo 1



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

1. Illustrare cosa sono le funzioni in C e qual'è la loro utilità.
2. Illustrare la differenza tra passaggio per valore e passaggio per puntatore. A cosa serve? Quando deve essere usato?

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

Scrivere la funzione **ricorsiva** *LetterChanges(char str[])* che preso in ingresso un'array di caratteri (lettere minuscole, numeri e segni), lo modifica secondo il seguente algoritmo:

- sostituisce ogni lettera nell'array con la lettera successiva; la z diventa a.
- trasforma in maiuscolo le vocali minuscole.

Esempi:

Input: "hello\*3"

Output: "lfmmp\*3"

Input: "fun times!"

Output: "gvO Ujnft!"

*suggerimento: ascii 'a' = 97; ascii 'A' = 65*

È possibile aggiungere parametri alla funzione se necessari.

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>

void LetterChanges(char str[]) {
    if (*str == '\0')
        return;
    else{
        if ((*str>='a') && (*str<='z')){
            *str = *str+1;
            if (*str>'z')
                *str = *str-'z'+ 'a'-1;
            switch(*str){
                case 'a':
                case 'e':
                case 'i':
                case 'o':
                case 'u':
                    *str = *str - 'a' + 'A';
                    break;
            }
        }
        LetterChanges(str+1);
    }
}

int main(void) {

    // keep this function call here
    char stringa[20] = "fun times!z";

    LetterChanges(stringa);
    printf("%s\n",stringa);
    return 0;
}
```

#### 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
- 0 se i tre numeri sono sia divisibili per 2 che per 3
  - 1 altrimenti (max 10 riga) (2 punto)

##### Risposta:

```
function out = agosto4_19(a,b,c)
    out = rem(a,2) + rem(b,2) + rem(c,2) + rem(a,3) + rem(b,3) + rem(c,3);
    if (out~=0)
        out = 1;
    end
```

- (b) Generare una matrice M con 6 righe e 4 colonne, contenente numeri casuali tra 0 e 100 (max 1 riga) (1 punto).

##### Risposta:

```
randi([0 100],[6 4])
```

- (c) Eliminare la colonne con media inferiore a 10 (max 2 riga) (1 punto).

##### Risposta:

```
M(:,mean(M,1)<10)=[]
```

- (d) Sostituire l'ultima riga con la radice quadrata dei valori della prima riga(max 1 riga) (1 punto).

**Risposta:**

```
M(end,:) = sqrt(M(1,:))
```

- (e) Inserire una nuova colonna contenente la media delle righe (max 2 righe) (1 punto).

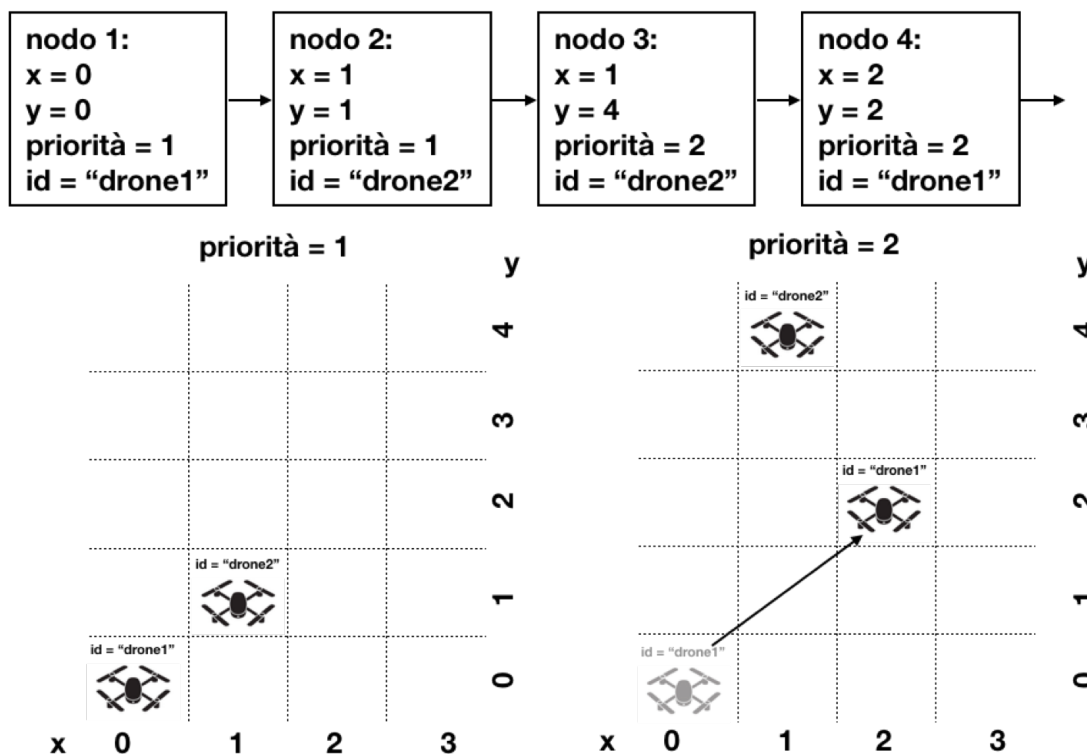
**Risposta:**

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

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

Si vuole creare un programma per la gestione di coreografie con droni. Il programma è basato su **UNA SOLA LISTA** che contiene:

- le coordinate cartesiane X e Y che identificano i punti nel piano nei quali si trovano i droni nei differenti passaggi;
- il nome/identificativo del drone;
- la priorità di esecuzione (un numero incrementale che identifica la posizione temporale).



**La distanza percorsa dal "drone1" è:**  $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{(2 - 0)^2 + (2 - 0)^2} = \sqrt{8} = 2,83$

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 *"ptrNodo inserisciOrdinato(ptrNodo lista, int priorit , int x, int y, char id[])"* per inserire in modo ordinato (in ordine crescente rispetto alla priorit  di esecuzione) un nuovo nodo. I valori da inserire sono letti nel main e passati come parametro. Non occorre effettuare controlli su eventuali nodi duplicati. (3 punti)
3. Scrivere una funzione *"int checkCollisione(ptrNodo lista, int priority)"* che preso in ingresso la lista e la priorit  di esecuzione, restituisca 1 se esiste pi  di un drone nella stessa posizione X,Y con la medesima priorit  (c'  collisione), 0 altrimenti (4 punti)
4. Scrivere la funzione *"int distanza(ptrNodo lista, char id[])"* che preso in ingresso la lista e l'identificativo di un drone, restituisca lo spostamento totale del drone (la somma totale della distanza tra i punti appartenenti al drone). La lista   gi  ordinata. (4 punti)

### Risposta:

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

typedef struct nodo{
    int x;
    int y;
    int priorit ;
    char id[10];
    struct nodo *next;
} nodo;

typedef nodo *ptrNodo;

void stampa(ptrNodo lista);

ptrNodo sortInsert(ptrNodo lista, int priorit , int x, int y, char id[]);
int checkCollisione(ptrNodo lista, int priority);
int distanza(ptrNodo lista, char id[]);

int main(int argc, const char * argv[]) {
    ptrNodo pianoVolo = NULL;
    int r,x,y,priorit ;
    char id[10];

    do{
        printf("MENU\n");
        printf("1) Inserisci nodo\n2) Stampa lista\n3) Check collisione\n4) Calcola spostamento\n5) ESCI\n\n>> ");
        scanf("%d",&r);
        switch (r) {
            case 1:
                printf("Inserisci x: ");
                scanf("%d",&x);
                printf("Inserisci y: ");
                scanf("%d",&y);
                printf("Inserisci priorit : ");
                scanf("%d",&priorit );
                printf("Inserisci id: ");
                scanf("%s",id);
                pianoVolo = sortInsert(pianoVolo, priorit , x, y, id);
                stampa(pianoVolo);
                break;
            case 2:
                stampa(pianoVolo);
                break;
            case 3:
                if (checkCollisione(pianoVolo,1)==1)
                    printf("C' collisione\n");
                else
                    printf("Nessuna collisione\n");
                break;
            case 4:
                printf("La distanza totale del drone 1 : %d\n\n",distanza(pianoVolo, "1"));
                break;
```

```

        default:
            break;
    }
}while(r!=6);
return 0;
}

void stampa(ptrNodo lista)
{
    if (lista == NULL)
        return;
    printf("X: %d\nY: %d\nPriority: %d\nID: %s\n\n",lista->x,lista->y,lista->priorita,lista->id);
    stampa(lista->next);
    return;
}

int checkCollisione(ptrNodo lista, int priority){
    ptrNodo savedLista = lista;
    ptrNodo l;

    int x,y;
    int first = 1;
    if (lista==NULL)
        return 0;
    for (l = savedLista; l!=NULL; l=l->next){
        first = 1;
        for (lista = l; lista!=NULL; lista=lista->next)
        {
            if (lista->priorita == priority){
                if (first==1){
                    x = lista->x;
                    y = lista->y;
                    first = 0;
                }
                else{
                    if ((x==lista->x) && (y==lista->y))
                        return 1;
                }
            }
        }
    }
    return 0;
}

ptrNodo sortInsert(ptrNodo lista, int priorita, int x, int y, char id[]) {
    ptrNodo n;
    if (lista != NULL && lista->priorita < priorita) {
        lista->next = sortInsert(lista->next, priorita, x, y, id);
        return lista;
    } else {
        n = (ptrNodo) malloc(sizeof(nodo));
        n->priorita = priorita;
        n->x = x;
        n->y = y;
        strcpy(n->id,id);
        if (lista != NULL) {
            n->next = lista;
        } else {
            n->next = NULL;
        }
    }
}

```

```

    }
    return n;
}
}

int distanza(ptrNodo lista, char id[]){
    ptrNodo n=NULL;
    int somma = 0;

    if ((lista == NULL) || (lista->next==NULL))
        return 0;

    for (lista; lista!=NULL; lista = lista->next)
    {
        if (strcmp(lista->id,id)==0){
            if (n==NULL)
                n = lista;
            else
            {
                somma = somma + sqrt(pow(lista->x-n->x,2)+pow(lista->y-n->y,2));
                n = lista;
            }
        }
    }

    return somma;
}

```











Politecnico di Milano

## Dipartimento di Elettronica, Informazione e Bioingegneria

**Informatica A - a.a 2018/2019 - 19 Febbraio 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	4 punti	_____
Esercizio 3	3 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 (not C or B) or A

**Risposta:**

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

OUT = A+B+B!C

- (b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri  $A = -23_{dec}$  e  $B = 41$ , 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:**

-23d = 101001 - 5bit + 1 di segno = 6bit  
 41d = 0101001b - 6 bit + 1 segno = 7bit  
 cp2 = 1010111b - cp2 7bit

$A + B = -23 + 41 = 18 = 0010010$  - riporto perduto no overflow  
 $A - B = -23 - 41 = -64 = [1]1000000b$  - Riporto perduto no overflow

**NOME e COGNOME:** \_\_\_\_\_

7 bit va da -64 a 63

- (c) Si converta il numero 47.375 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

47d = 0101111

$0.25 + 0.125 = 0.011$

37.59375 = 0101111.011 j- Virgola fissa

segno = 0

mantissa = 01.01111.011 non normalizzata

exp = 5 + 127 = 132d = 10000100

IEEE754

segno (1bit) = 0

esponente (8bit) = 10000100

mantissa (23bit) = 101 1110 1100 0000 0000 0000

**Esercizio 2 - Domanda di teoria (4 punti)**

(a) Che cosa si intende per puntatore? A cosa serve?

(b) Fare un esempio per dichiarare:

- un puntatore ad una lista. Specificare anche la struttura base della lista in modo che contenga solo un campo di tipo intero.

- un puntatore ad una variabile char

(c) Dato c, puntatore di tipo char:

- allocare la memoria necessaria per la variabile puntata da c (malloc)

- riempire la memoria puntata da c con il carattere 'a'



**Esercizio 3 - Comprensione del Codice (3 punti)**

Scrivere cosa stampa a video il seguente codice:

```
#include <stdio.h>
#include <string.h>
#define N 6

char doFirstThing(char *a);
int thenTheSecond(char a);

int main(void) {
    char *a;
    char c;
    int d;

    a = (char*)malloc(sizeof(char)*N);
    strcpy(a,"abcde");
    c = doFirstThing(a);

    printf("%c\n",c);

    d = thenTheSecond(*a);
    printf("%c\n",*a);

    printf("%d\n",d);

    return 0;
}

char doFirstThing(char *a){
    int i,med = 0, n=0;

    for (i=0; *a!='\0'; i++)
    {
        med = med + *a;
        a = a+1;
        n = n+1;
    }
    a = a-1;
    med = med / n;
    return med;
}

int thenTheSecond(char a){
    a = a - 'a';
    return a;
}
```

**Risposta:**

c  
a  
0

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

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

- (a) Creare la matrice M con dimensione di righe e colonne casuali (diverse) da 5 a 10 contenente solo valori pari a 1; (max 1 riga) (1 punto)

**Risposta:**

```
M = ones(randi([5 10],[1 2]));
```

- (b) Inserire nelle righe dispari valori casuali da 1 a 10 (max 1 riga) (1 punto).

**Risposta:**

```
M(1:2:end,:) = randi([1,10],size(M(1:2:end,:)))
```

- (c) Sommare 1 alle celle della matrice che contengono numeri dispari (max 1 riga) (1 punto).

**Risposta:**

```
M(mod(M,2)==1) = M(mod(M,2)==1)+1
```

- (d) Calcolare la media delle celle con valore maggiore di 3 (max 1 riga) (1 punto).

**Risposta:**

```
mean (M(M>3))
```

- (e) Scrivere una funzione che presa in ingresso la matrice M, crei una nuova matrice "specchio" N con la prima colonna in ultima posizione, la seconda in penultima... (max 8 righe) (2 punto).

**Risposta:**

```
function [N] = funzione(M)
    for i=1:size(M,2)
        N(:,size(M,2)-i+1) = M(:,i);
    end
end
```

## Esercizio 5 - Programmazione C (12 punti)

Si vuole progettare un software che permetta di registrare trasazioni di criptovalute. Ogni record del database costituisce una coin della criptovaluta ed è costituito da un identificativo univoco di 10 cifre , il valore in euro al quale è stato acquistato, il valore in euro al quale è stato venduto, il nome del servizio utilizzato per lo scambio. Ogni coin ha un solo proprietario che è identificato da nome, cognome, identificativo del blocco posseduto e anno di acquisto. Sia i coin che i proprietari sono definiti per mezzo di una lista.

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

1. Definire le strutture dati necessari per la realizzazione del programma. (1 punto)
2. Scrivere la funzione per inserire un nuovo coin ed un nuovo proprietario. L'id è calcolato in automatico partendo da 0 e aggiungendo 1 ad ogni nuovo nodo. (2 punti)
3. Scrivere la funzione che preso in ingresso un identificativo, restituisca la differenza tra prezzo di vendita e prezzo d'acquisto. Restituisce 0 se l'identificativo non esiste o la lista è vuota. (3 punti)
4. Scrivere la funzione che, preso in ingresso nome e cognome del proprietario, permette di calcolare il guadagno/perdita facendo la differenza tra prezzo d'acquisto e prezzo di vendita di tutti i coin posseduti. (3 punti)
5. Scrivere la funzione che preso in ingresso un anno, salva e restituisce un array (lunghezza massima 100) con i valori di vendita di tutte i blocchi che sono stati scambiati per mezzo della piattaforma "coin-base". (3 punti)

### Risposta:

```
/*
Si vuole progettare un software che permette di registrare trasazioni di criptovalute. Ogni record del database
costituisce una coin della criptovaluta ed è costituito da un identificativo univoco di 10 cifre , il valore in euro
al quale è stato acquistato, il valore in euro al quale è stato venduto, il nome del servizio utilizzato per lo
scambio. Ogni coin ha un solo proprietario che è identificato da nome, cognome, identificativo del blocco
posseduto e anno di acquisto. Sia i coin che i proprietari sono definiti per mezzo di una lista.
Svolgere l'esercizio attenendosi a quanto richiesto. NON RICHiesto SCRIVERE IL MAIN.
1. Definire le strutture dati necessari per la realizzazione del programma. (1 punto)
2. Scrivere la funzione per inserire un nuovo coin. L'id è calcolato in automatico partendo da 0 e aggiungendo 1
ad ogni nuovo nodo. (2 punti)
3. Scrivere la funzione che preso in ingresso un identificativo, restituisca la differenza tra prezzo di vendita e
prezzo d'acquisto. Restituisce 0 se l'identificativo non esiste o la lista è vuota. (3 punti)
4. Scrivere la funzione che, preso in ingresso nome e cognome del proprietario, permette di calcolare il guadagno/
perdita facendo la differenza tra prezzo d'acquisto e prezzo di vendita di tutti i coin posseduti. (3 punti)
5. Scrivere la funzione che preso in ingresso un anno, salva e restituisce un array (lunghezza massima 100) con
i valori di vendita di tutte i blocchi che sono stati scambiati per mezzo della piattaforma "coin-base". (3 punti)
*/

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

typedef struct coin{
    int id;
    float acquisto;
    float venduto;
    char nomeServizio[20];
    struct coin *next;
} coin;
```

```

typedef coin *p_coin;

typedef struct proprietari{
    char nome[20];
    char cognome[20];
    int id;
    int anno;
    struct proprietari *next;
} proprietari;

typedef proprietari *p_proprietari;

p_coin inserisciNode(p_coin lista, p_proprietari *prop, float acq, float ven, char nomeServizio[], char nomeProprietario[], float guadagnoSuVendita(p_coin lista, int id);
float guadagnoTotale(p_coin listaCoin, p_proprietari listaProp, char nome[], char cognome[]);
float creaArray (p_coin listaCoin, p_proprietari listaProp, int anno, float array[], int *);
void stampaListe(p_coin coins, p_proprietari props);

int main(int argc, const char * argv[]) {

    int c;

    float acq, ven;
    char nome[20],cognome[20], nomeServizio[20];
    int id, anno;
    float gain, media;
    float arrayVen[100];
    int k;

    p_coin coins = NULL;
    p_proprietari proprietari = NULL;

    do{
        printf("\n\nMENU\n");
        printf("1) Inserisci nuova coin\n");
        printf("2) Calcola guadagno per ID coin\n");
        printf("3) Calcola guadagno per nome e cognome\n");
        printf("4) Crea array coinbase\n");
        printf("5) Stampa liste\n");
        printf("6) ESCI\n\n");
        printf(">> ");
        scanf("%d",&c);

        switch (c) {
            case 1:
                printf("Prezzo acquisto: ");
                scanf("%f",&acq);
                printf("Prezzo vendita: ");
                scanf("%f",&ven);
                printf("Nome servizio: ");
                scanf("%s",nomeServizio);

                printf("Nome proprietario: ");
                scanf("%s",nome);
                printf("Cognome proprietario: ");
                scanf("%s",cognome);
                printf("Anno: ");
                scanf("%d",&anno);

```

```

        coins = inserisciNodo(coins,&proprietari,acq,ven,nomeServizio,nome,cognome,anno);
        break;

    case 2:
        printf("ID: ");
        scanf("%d",&id);
        gain = guadagnoSuVendita(coins, id);
        printf("Il guadagno su %d %.2f\n",id,gain);
        break;
    case 3:
        printf("Nome: ");
        scanf("%s",nome);
        printf("Cognome: ");
        scanf("%s",cognome);
        printf("Il guadagno totale dell'utente %s %s : %.f",nome,cognome,guadagnoTotale(coins, propri
        break;
    case 4:
        printf("Anno: ");
        scanf("%d",&anno);
        media = creaArray (coins, proprietari, anno, arrayVen, &k);
        printf("Media %.2f\n\n",media);
        for(int i=0; i<k; i++)
            printf("%d: %.2f\n",i,arrayVen[i]);
        break;
    case 5:
        stampaListe(coins, proprietari);
        break;
    default:
        break;
}
}while(c != 6);
return 0;
}

p_coin inserisciNodo(p_coin lista, p_proprietari *prop, float acq, float ven, char nomeServizio[], char nom
{
    p_coin temp;
    p_proprietari tempProp;

    temp = (p_coin)malloc(sizeof(coin));
    temp->acquisto = acq;
    temp->venduto = ven;
    strcpy(temp->nomeServizio, nomeServizio);
    if (lista == NULL)
        temp->id = 0;
    else
        temp->id = lista->id+1;
    temp->next = lista;
    lista = temp;

    tempProp = (p_proprietari)malloc(sizeof(proprietari));
    strcpy(tempProp->nome,nome);
    strcpy(tempProp->cognome,cognome);
    tempProp->id = lista->id;
    tempProp->anno = anno;
    if (*prop == NULL)
        tempProp->next = NULL;
    else
        tempProp->next = *prop;
    *prop = tempProp;
}

```

```

        return lista;
    }

    float guadagnoSuVendita(p_coin lista, int id){
        if (lista == NULL)
            return 0;
        else
        {
            if (lista->id == id)
                return (lista->venduto-lista->acquisto);
            else
                return guadagnoSuVendita(lista->next, id);
        }
    }

    float guadagnoTotale(p_coin listaCoin, p_proprietari listaProp, char nome[], char cognome[]){
        if (listaProp==NULL)
            return 0;
        else
        {
            if ((strcmp(listaProp->nome,nome)==0) && (strcmp(listaProp->cognome, cognome)==0)) //NOME E COGNOME
                return guadagnoSuVendita(listaCoin,listaProp->id) + guadagnoTotale(listaCoin, listaProp->next, n
            else
                return guadagnoTotale(listaCoin, listaProp->next, nome, cognome);
        }
    }

    float creaArray (p_coin listaCoin, p_proprietari listaProp, int anno, float array[], int *k){
        p_coin testaCoin = listaCoin;
        float med = 0;
        int i=0;
        if (listaProp == NULL)
            return 0;
        else{
            while(listaProp!=NULL)
            {
                if (listaProp->anno == anno)
                {
                    listaCoin = testaCoin;
                    while(listaCoin!=NULL)
                    {
                        if ((listaCoin->id == listaProp->id) && (strcmp(listaCoin->nomeServizio,"coinbase")==0))
                        {
                            array[i] = listaCoin->venduto;
                            med = med + array[i];
                            i++;
                            break;
                        }
                        else
                            listaCoin = listaCoin->next;
                    }
                }
                listaProp = listaProp->next;
            }
        }
        *k = i;
        med = med/i;
        return med;
    }
}

```

```

void stampaListe(p_coin coins, p_proprietari props){
    if (coins!=NULL)
    {
        printf("ID: %d\tAcquisto: %.2f\tVendita: %.2f\tServizio: %s\n", coins->id, coins->acquisto, coins->vendita, coins->servizio);
        stampaListe(coins->next, props);
    }
    else{
        if (props!=NULL)
        {
            printf("ID: %d\tNome: %s\tCognome: %s\tAnno: %d\n", props->id, props->nome, props->cognome, props->anno);
            stampaListe(coins, props->next);
        }
    }
}

```

**NOME e COGNOME:** \_\_\_\_\_



**NOME e COGNOME:** \_\_\_\_\_



Politecnico di Milano

## Dipartimento di Elettronica, Informazione e Bioingegneria

**Informatica A - a.a 2018/2019 - 28 Gennaio 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 and B or not C) and C or not B

**Risposta:**

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

OUT = ABC + !B

- (b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri A = -64dec e B = 33, 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:**

-64d = 1000000b - 6bit + 1 di segno = 7bit

33d = 0100001b - 6 bit + 1 segno

cp2 = 1011111b - cp2 7bit

A + B = -64 + 33 = -31 = 1100001b - No riporto perduto né overflow

A - B = -64 - 33 = -97 = [1]0011111b - Riporto perduto con overflow

7 bit va da -64 a 63

- (c) Si converta il numero 37.59375 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 = 0100101

$0.5 + 0.0625 + 0.03125 = 0.10011$

37.59375 = 0100101.10011 j- Virgola fissa

segno = 0

mantissa = 1.0010110011 non normalizzata

exp = 5 + 127 = 132d = 10000100

IEEE754

segno (1bit) = 0

esponente (8bit) = 10000100

mantissa (23bit) = 100 1011 0011 0000 0000 0000

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

- (a) Spiegare che cosa si intende per iterazione e per ricorsione. Quando è necessario, o conveniente, usarle? In generale è più efficiente l'iterazione o la ricorsione?

- (b) Implementare una semplice funzione di esempio sia in versione iterativa che in versione ricorsiva.

**Risposta:**

### Esercizio 3 - Comprensione del Codice (4 punti)

Scrivere cosa stampa a video il seguente codice:

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

char* doSomething(char *a, char *b);
int main(void) {
    char *a;
    char *b;

    int n = 7;
    a = (char*)malloc(sizeof(char)*n);
    strcpy(a,"casale");
    b = doSomething(a,b);
    printf("%s %c\n",a,*b);
    return 0;
}

char* doSomething(char *a, char *b){
    int n = 0;
    char c;
    b=a;
    for (b,c='1',n=1; b<&a[strlen(a)]; b++,c++)
    {
        if ((*b)=='a')
        {
            n++;
            *b = c;
        }
        else if ((*b)=='a'+4)
        {
            n++;
            *b = c-'1'+3;
        }
        printf("%s %c\n",a,*b);
    }
    b = (char*)malloc(sizeof(char));
    *b = ('0'+n);
    return b;
}
```

printf:

```
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
```

**Risposta:**

casale c c2sale 2 c2sale s c2s4le 4 c2s4le l c2s4l8 8 c2s4l8 4



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

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

- (a) Creare un vettore colonna A e un vettore colonna B contenente rispettivamente 6 e 5 valori casuali differenti tra loro con valori da 1 a 10; (max 2 riga) (1 punto)

**Risposta:**

```
A = randi([1 10],[6 1]);  
B = randi([1 10],[1 5]);
```

oppure

```
A = randi([1 10],[6 1]);  
B = randi([1 10],[5 1]);
```

- (b) A partire dai vettori A e B, creare una matrice C di 6 righe e 5 colonne (1 punto).

**Risposta:**

Se una colonna e una riga

```
C = A*B;  
C = A.*B;
```

Se entrambe colonne:

```
C = A*B';  
C = A.*B';
```

- (c) Cancellare le righe che hanno come primo valore un numero inferiore a 10 (max 1 riga) (1 punto).

**Risposta:**

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

- (d) Calcolare la somma di tutte le colonne dispari (max 1 riga) (1 punto).

**Risposta:**

```
sum(sum(C(:,1:2:end)))
```

- (e) Scrivere una funzione che presa in ingresso la matrice C, crea una nuova matrice "cornice", ponendo cioè a zero tutti gli elementi non appartenenti alla prima e ultima riga/colonna. (max 8 righe) (2 punto).

**Risposta:**

```
function [out] = funzione(in)
    out = in;
    for i=2:1:size(in,1)-1
        for j=2:1:size(in,2)-1
            out(i,j) = 0;
        end
    end
end
```

## Esercizio 5 - Programmazione C (12 punti)

Si vuole creare un programma per trasformare una serie di immagini in una GIF animata.

Un'immagine è definita da una struttura che contiene tre vettori bidimensionali 100x100 contenete valori che vanno da 0 a 255. Ogni matrice corrisponde al colore rosso, verde e blue.

Una GIF animata è costituita da una lista i cui elementi sono composti da un'immagine (i tre vettori bidimensionali) ed un numero che corrisponde ai secondi per cui l'immagine viene visualizzata durante l'animazione.

1. Definire le strutture dati necessari per la realizzazione del programma. (1 punto)
2. Scrivere una funzione che prende in ingresso due immagini e controlli se differiscono di almeni N pixel. N viene definito dall'utente ed è un parametro che viene passato alla funzione. I pixel differiscono quando i valori dei vettori bidimensionali nella medesima posizione sono differenti (es. `immRed1[10][10] != immRed2[10][10]`). Se le immagini differiscono la funzione ritorna 1, se non differiscono restituisce 0. (2 punti)
3. Scrivere una funzione che presa in ingresso un'immagine e un tempo t, la aggiunge in coda alla lista contenente tutte le immagini della GIF animata. Prima di aggiugnerle deve controllare che l'immagine differisca di almeno 10 pixel dall'immagine precedente (Si consiglia di usare la funzione del punto 2). (3 punti)
4. Si vuole riprodurre la GIF al contrario. Scrivere la funzione che presa in ingresso la lista della GIF, crei una nuova lista inserendo gli elementi al contrario. (3 punti)
5. Scrivere una funzione che presa in ingresso la GIF ed una variabile tempo T, elimini dalla lista tutte le immagini che hanno durata inferiore a T. Non deve essere creata una nuova lista, ma vanno eliminati gli elementi dalla lista di partenza. (3 punti)

### Risposta:

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

#define M 5

typedef struct immagine{
    int red[M][M];
    int green[M][M];
    int blue[M][M];
} immagine;

typedef struct imgGif{
    immagine img;
    int tempo;
    struct imgGif *next;
} imgGif;

typedef imgGif* ptrGif;

//Funzioni
void generaMatrici(int matrix[][M], int i);
void stampaMatrice(int matrix[][M]);
immagine generaImmagine(int i);
void stampaGif(ptrGif gif);
```

```

int confrontaImmagini(immagine img1, immagine img2, int n);
ptrGif aggiungiGif(ptrGif gif, immagine img, int tempo);
ptrGif contrarioIt(ptrGif gif);
ptrGif contrario(ptrGif gif, ptrGif contrario);
ptrGif velocizza(ptrGif gif, T);

int main(void) {
    ptrGif gif = NULL;
    ptrGif gifContrario = NULL;

    immagine img1;
    immagine img2;
    int c;
    int r;
    int val;

    srand((unsigned)time(0));

    do{
        printf("\n\nMENU\n");
        printf("L'opzione 2 genera una matrice 5x5 e non 100x100 per velocizzare il tutto.
            Il comando genera una matrice uguale o una casuale (per cui diversa)\n\n");
        printf("1) Confronta immagini\n2) Inserisci immagini\n3) Lista al contrario\n4)
            Velocizza GIF\n5) Stampa gif\n6) ESCI\n\n>> ");
        fpurge(stdin);
        scanf("%d",&c);
        switch (c) {
            case 1:
                //Per confrontare immagine differenti mettere 0 e 0 o 0 e 1 nella generazione
                //Per confrontare immagine uguali mettere 1 e 1
                img1 = generaImmagine(0);
                img2 = generaImmagine(1);
                if (confrontaImmagini(img1, img2, 5))
                    printf("Le immagini sono differenti\n");
                else
                    printf("Le immagini non sono differenti\n");

                break;
            case 2:
                printf("Uguale 0, diversa 1: ");
                scanf("%d",&r);
                img1 = generaImmagine(r);
                printf("Inserisci tempo: ");
                scanf("%d",&r);
                gif = aggiungiGif(gif, img1, r);
                break;

            case 3:
                stampaGif(gif);
                printf("\n\nAl contrario: \n\n");
                //gifContrario = contrario(gif, gifContrario);
                gifContrario = contrarioIt(gif);
                stampaGif(gifContrario);
                break;

            case 4:
                printf("Inserire valore minimo tempo immagine: ");
                scanf("%d",&val);
                gif = velocizza(gif, val);

```

```

        break;

        case 5:
            stampaGif(gif);
        default:
            break;
    }
}while(c!=6);
return 0;
}

void generaMatrici(int matrix[][M], int k)
{
    int i,j;
    for (i=0; i<M; i++)
    {
        for (j=0; j<M; j++)
        {
            if (k!=0)
                matrix[i][j]=rand()%256;
            else
                matrix[i][j]=100;
        }
    }
}

void stampaMatrice(int matrix[][M])
{
    int i,j;
    for (i=0; i<M; i++)
    {
        for (j=0; j<M; j++)
        {
            printf("%d ",matrix[i][j]);
        }
        printf("\n");
    }
}

void stampaGif(ptrGif gif)
{
    if (gif==NULL)
        return;
    stampaMatrice(gif->img.blue);
    //stampaMatrice(gif->img.green);
    //stampaMatrice(gif->img.red);
    printf("%d\n",gif->tempo);
    stampaGif(gif->next);
}

immagine generaImmagine(int i){
    immagine img;
    generaMatrici(img.blue, i);
    generaMatrici(img.red, i);
    generaMatrici(img.green, i);
    return img;
}

int confrontaImmagini(immagine img1, immagine img2, int n)

```

```

{
    int i, j;
    int cont = 0;

    for (i=0; i<M; i++)
    {
        for (j=0; j<M; j++)
        {
            if ((img1.blue[i][j] != img2.blue[i][j]) || (img1.red[i][j] != img2.red[i][j])
                || (img1.green[i][j] != img2.green[i][j]))
                cont++;
            // if (img1.blue[i][j] != img2.blue[i][j])
            // cont++;
            // else if (img1.red[i][j] != img2.red[i][j])
            // cont++;
            // else if (img1.green[i][j] != img2.green[i][j])
            // cont++;
        }
    }
    if (cont>n)
        return 1;
    else
        return 0;
}

ptrGif aggiungiGif(ptrGif gif, immagine img, int tempo)
{
    //int i,j;
    if (gif==NULL)
    {
        gif = (ptrGif)malloc(sizeof(imgGif));
    // for (i=0; i<M; i++)
    // {
    // for (j=0; j<M; j++){
    // gif->img.red[i][j] = img.red[i][j];
    // gif->img.blue[i][j] = img.blue[i][j];
    // gif->img.green[i][j] = img.green[i][j];
    // }
    // }

        gif->img = img;
        gif->tempo = tempo;
        gif->next = NULL;
        return gif;
    }

    if ((gif->next!=NULL) || ((gif->next == NULL) && (confrontaImmagini(gif->img,img,10))))
        gif->next = aggiungiGif(gif->next, img, tempo);
    return gif;
}

ptrGif contrarioIt(ptrGif gif)
{
    ptrGif contr = NULL;
    ptrGif temp = NULL;

    while (gif!=NULL){
        temp = (ptrGif)malloc(sizeof(imgGif));
        temp->img = gif->img;
        temp->tempo = gif->tempo;
        temp->next = contr;
    }
}

```

```

        contr = temp;
        gif = gif->next;
    }
    return contr;
}

ptrGif contrario(ptrGif gif, ptrGif contr){
    ptrGif temp;
    int i,j;

    if (gif==NULL)
        return contr;

    temp = (ptrGif)malloc(sizeof(imgGif));
    for (i = 0; i<M; i++)
    {
        for (j=0; j<M; j++)
        {
            temp->img.blue[i][j] = gif->img.blue[i][j];
            temp->img.red[i][j] = gif->img.red[i][j];
            temp->img.green[i][j] = gif->img.green[i][j];
        }
    }
    temp->tempo = gif->tempo;
    temp->next = contr;
    temp = contrario(gif->next, temp);
    return temp;
}

ptrGif velocizza(ptrGif gif, int T){
    ptrGif temp;
    if (gif==NULL)
        return NULL;
    if (gif->tempo<T)
    {
        temp = gif->next;
        free(gif);
        temp = velocizza(temp, T);
        return temp;
    }
    else
    {
        gif->next = velocizza(gif->next, T);
        return gif;
    }
}

```









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;
}
```





