



Politecnico di Milano

Dipartimento di Elettronica, Informazione e Bioingegneria

Informatica A - Prof. A. Fuggetta - a.a 2021/2022 - 27 Agosto 2022

Cognome: _____ Matricola: _____
Nome: _____ Firma: _____

Istruzioni

- Non separate questi fogli. Scrivete la soluzione **solo sui fogli distribuiti**, utilizzando il retro delle pagine in caso di necessità. **Cancellate le parti di brutta** con un tratto di **penna**.
- Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
- **NON è possibile scrivere a matita.**
- Scrivere nome e cognome su tutti i fogli. Non verranno corretti compiti non firmati o con nome illeggibile
- È **vietato** utilizzare **calcolatrici, telefoni o pc**. Chi tenti di farlo vedrà **annullata** la sua prova.
- **Non è ammessa la consultazione di libri e appunti.**
- Qualsiasi **tentativo** di comunicare con altri studenti comporta **l'espulsione** dall'aula.
- È possibile ritirarsi senza penalità.
- Non è possibile lasciare l'aula conservando il tema della prova in corso.
- Tempo a disposizione: **2.30h**

Valore indicativo degli esercizi, voti parziali e voto finale:

Esercizio 1	5 punti	_____
Esercizio 2	6 punti	_____
Esercizio 3	7 punti	_____
Esercizio 4	5 punti	_____
Esercizio 5	9 punti	_____
Totale(32)		_____

Esercizio 1 - Algebra di Boole, Aritmetica Binaria, Codifica delle Informazioni (5 punti)

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

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

Risposta:

Raccolgo subito A

$A (B + \text{not } B + A \dots)$

$B + \text{NOT } B = 1$

L'equazione equivale ad A

$A \text{ AND } B \text{ OR } A \text{ AND } (\text{NOT } B \text{ OR } A \text{ AND NOT } (B \text{ AND NOT } C))$

$$AB + A(\bar{B} + A\bar{B}\bar{C})$$

$$AB + A(\bar{B} + A(\bar{B} + \bar{C}))$$

$$AB + A(\bar{B} + A\bar{B}\bar{C})$$

$$AB + A\bar{B}(\bar{C})$$

$$A(\bar{B} + B) = A$$

$$A(\bar{B} + B + A\bar{B}\bar{C})$$

$$\begin{matrix} | \\ \hline \\ \hline \end{matrix} = A$$

- (b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri $A = -32_{\text{dec}}$ e $B = 21_{\text{dec}}$ li si converta, se ne calcolino la somma $(A+B)$ e la differenza $(A-B)$ in complemento a due e si indichi se si genera riporto sulla colonna dei bit più significativi e se si verifica overflow. Non si accetteranno soluzioni senza il procedimento. (2 punto)

-32
↑
6 bit

21

6 bit

100000

100000
010101

110101

NO OF
NO CR

21 | 1
10 | 0 010101
5 | 1
2 | 0
1 | 1
0
110000
10101

001011

OF

- (c) Si converta il numero 12,625 in virgola fissa e in virgola mobile con codifica IEEE 754 con precisione singola. Non si accetteranno soluzioni senza il procedimento. (2 punto)

$$\begin{array}{l}
 12,625 \\
 \swarrow \quad \searrow \\
 1100 \quad 0,5 + 0,125 \\
 \quad \quad 0,101
 \end{array}$$

$$01100,101$$

$$S = 0$$

$$P = 100101 \dots \quad 236 \sim$$

$$E = 127 + 3 = 130$$

Esercizio 2 - Teoria (6 punti)

- (a) Data una CPU a 8 bit in complemento a 2, si vuole eseguire la somma tra i numeri ESADECIMALI 10hex e 70hex

Quale delle seguenti affermazioni sono vere

- ☐ viene visualizzato il valore decimale 128
- ☒ viene visualizzato il valore esadecimale 80
- ☐ viene visualizzato il valore decimale 100
- ☐ viene visualizzato il valore esadecimale 100
- ☐ viene visualizzato il valore decimale 80

- (b) Data l'operazione precedente, quale delle seguenti affermazioni è corretta:

- ☒ È attivo solo il bit di overflow
- ☐ È attivo il bit di carry
- ☒ È attivo il bit di segno
- ☐ Nessuna delle precedenti

- (c) L'intervallo di valori rappresentabili con 6 bit, utilizzando la rappresentazione in complemento a 2 è:

- ☒ da -32dec a 31dec
- ☐ da -31dec a 32dec
- ☒ da -20hex a 1Fhex
- ☐ da -16hex a 15hex

- (d) Una variabile di tipo *double*?

- ☐ ha un ingombro di memoria pari a quello di una variabile di tipo float
- ☐ non ammette un indirizzo di memoria
- ☐ ha un valore massimo pari alla metà di quello del long double
- ☒ ha ingombro di memoria pari a 8 byte
- ☐ è in gradi di memorizzare senza approssimazione il valore 0.6 in binario

- (e) Dato il prototipo di funzione

`float* f(float uno, int *due);`

e le dichiarazioni di variabili

`float *x;`

`int *y;`

stabilire quale delle seguenti invocazioni di funzione rispetta il prototipo.

- ☐ `x=f(y,*y)`
- ☒ `x=f(*x,y)`
- ☐ `*y = f(x, x)`
- ☐ `y = f(*y, y)` *è corretto.*
- ☐ `y=f(y,y)`

(f) Si consideri il seguente programma.

```
#include <stdio.h>
void func (int x, int *y) {
    x++;
    (*y)++;
}
int main () {
    int a = 1, b = 2;
    func (a, &b);
    printf ("%d", a + b);
    return 0;
}
```

Cosa stampa la funzione printf?

☐ 2

☐ 3

☒ 4

☐ 5

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

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

1. Creare una matrice quadrata di dimensione 6 contenente numeri casuali da 1 a 100. (1 riga - 1 punto)

```
A = randi([1 100],6);
```

2. Dividere per due tutte le celle pari. (1 riga - 1 punto)

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

3. Cancellare le colonne con media inferiore a 5 (1 riga - 1 punto)

```
A(mean(A,1)<5) = []
```

4. Aggiungere una riga all'inizio della matrice contenente in ogni cella il valore della somma totale della matrice (1 riga - 1 punti)

```
A = [sum(A,'all')*ones(1,size(a,2));A]
```

5. Creare un array contenente la media delle colonne (1 riga - 1 punti)

```
B = mean(A,1)
```

6. Scrivere una funzione Matlab che presa in ingresso la matrice e l'array delle medie calcolato al punto 5, restituisca l'array A contente il numero di valori di ogni colonna superiore alla media della colonna. (2 punti)

Function x = funzioneMedia(A,B)

```
x = [];
for (i=1:1:size(A,2))
    For (j=1:1:size(A,1))

        if (A(i,j)>B(i))
            x=[x S(i,j)]
        end
    End
End
```

(a) **Programmazione C (5 punti)**

Scrivere un sottoprogramma baricentro che ricevuto in ingresso un array di valori interi e qualsiasi altro parametro ritenuto strettamente necessario restituisce l'indice dell'elemento che fa da baricentro agli elementi dell'array, -1 se non esiste. Definiamo baricentro di un array di h elementi, l'elemento in posizione i tale per cui:

$$v[0] + v[1] + \dots + v[i] = v[i+1] + v[i+2] + \dots + v[h-1]$$

ovvero la somma degli elementi dell'array da 0 a i è uguale alla somma degli elementi dell'array da i+1 fino alla fine.

Risposta:

```
#include <stdio.h>

int baricentro(int array[], int l);

int main(void) {
    int a[] = {1,2,4,5,9,2,1};

    printf("Baricentro = %d", baricentro(a,7));
    return 0;
}

int baricentro(int array[], int l){
    int i,j;
    int somma;
    for (i=1; i<l-1;i++){
        somma=0;
        for (j=0; j<i; j++)
            somma+=array[j];
        for (j=i;j<l;j++)
            somma-=array[j];
        if (somma==0)
            return i;
    }
    return -1;
}
```

(b) Programmazione C Liste (9 punti)

Si ipotizzi di dovere realizzare un software per la gestione delle letture realizzate da più sensori di temperatura e umidità all'interno di serre per la floricoltura.

Una lettura è composta da l'orario in cui è stata effettuata (HH:MM:SS), una stringa che indica il nome del sensore ovvero la posizione in cui è stata effettuata la lettura (es. "Serra1", "Tulipani"...), ed i valori stessi delle letture in gradi Celsius e percentuale di umidità.

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 7200 misurazioni; una volta raggiunto il massimo delle 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. (2 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 7200 letture, la prima lettura viene automaticamente cancellata. (3 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 "anteriore" il secondo orario. `ptrLista rimuoviLetture(ptrLista lista, int daOra, int aOra)` (3 punti)
5. (SOLO PER CHI NON HA SUPERATO IL LABORATORIO)
Si definisca la funzione `elimina_nodo` che presa in ingresso la lista, il nome del sensore ed un numero `n`, elimini dalla lista le prime `n` registrazioni.

```
ptrNode rimuoviLetture(ptrNode lista, char daOra[], char aOra[]){
    ptrNode temp;
    if (lista==NULL)
        return lista;
    if ((strcmp(lista->orario,daOra)>=0) && (strcmp(lista->orario,aOra)<=0)){
        temp = lista;
        lista = lista->next;
        free(temp);
        lista = rimuoviLetture(lista, daOra, aOra);
    }
    else {
        lista->next = rimuoviLetture(lista->next, daOra, aOra);
    }
    return lista;
}
```

```
void stampaLista(ptrNode lista){
    if (lista==NULL)
        return;
    else
    {
        printf("%s %s %.2f %.2f\n",lista->nome, lista->orario, lista->gradi, lista->umidita);
        stampaLista(lista->next);
    }
}
```



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

#define MAX 7200

typedef struct nodo {
    char orario[9];
    char nome[20];
    float gradi;
    float umidita;
    struct nodo *next;
} nodo;

typedef nodo *ptrNode;

int contaNodi(ptrNode lista);
ptrNode aggiungiLettura(ptrNode lista, int ora, int minuti, int secondi, char nome[], float valC,
float valH);
ptrNode rimuoviLettture(ptrNode lista, char daOra[], char aOra[]);
void stampaLista(ptrNode lista);

int main(void) {
    ptrNode lista = NULL;
    char nome[20];
    int r;
    do{
        printf("Menu\n");
        printf("1) Inserisci\n");
        printf("2) Rimuovi\n");
        printf("3) Stampa\n");
        printf("4) ESCI\n");
        scanf("%d",&r);
        switch(r){
            case 1:
                lista = aggiungiLettura(lista,1,1,1,nome,1,1);
                lista = aggiungiLettura(lista,1,1,2,nome,1,1);
                lista = aggiungiLettura(lista,1,1,3,nome,1,1);
                lista = aggiungiLettura(lista,1,1,4,nome,1,1);
                lista = aggiungiLettura(lista,1,1,5,nome,1,1);
                lista = aggiungiLettura(lista,1,1,6,nome,1,1);
                break;
            case 2:
                lista = rimuoviLettture(lista,"1:1:2","1:1:4");
                break;
            case 3:
                stampaLista(lista);
```

break;

```
    }  
    }while(r<3);  
    return 0;  
}
```

NOME e COGNOME: _____

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

```
ptrNode aggiungiLettura(ptrNode lista, int ora, int minuti, int secondi, char nome[], float  
valC, float valH){  
    ptrNode temp = lista;  
    if (lista == NULL){  
        lista = (ptrNode)malloc(sizeof(nodo));  
        sprintf(lista->orario, "%d:%d:%d",ora,minuti,secondi);  
        strcpy(lista->nome,nome);  
        lista->gradi = valC;  
        lista->umidita = valH;  
        lista->next = NULL;  
        return lista;  
    }
```

```
    if (contaNodi(lista)>=MAX){  
        temp = lista;  
        lista = lista->next;  
        free(temp);  
    }
```

```
    while(temp->next != NULL){  
        temp = temp->next;  
    }  
    temp->next = (ptrNode)malloc(sizeof(nodo));  
    temp = temp->next;  
    sprintf(temp->orario, "%d:%d:%d",ora,minuti,secondi);  
    strcpy(temp->nome,nome);  
    temp->gradi = valC;  
    temp->umidita = valH;  
    temp->next = NULL;
```

```
    return lista;  
}
```



Politecnico di Milano

Dipartimento di Elettronica, Informazione e Bioingegneria

Informatica A - a.a 2020/2021 - 24 giugno 2021

Cognome: _____ **Matricola:** _____
Nome: _____ **Firma:** _____

Istruzioni

- Non separate questi fogli. Scrivete la soluzione **solo sui fogli distribuiti**, utilizzando il retro delle pagine in caso di necessità. **Cancellate le parti di brutta** con un tratto di **penna**.
- Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
- **NON è possibile scrivere a matita.**
- Scrivere nome e cognome su tutti i fogli. Non verranno corretti compiti non firmati o con nome illeggibile
- È **vietato** utilizzare **calcolatrici, telefoni o pc**. Chi tenti di farlo vedrà **annullata** la sua prova.
- **Non è ammessa la consultazione di libri e appunti.**
- Qualsiasi **tentativo** di comunicare con altri studenti comporta **l'espulsione** dall'aula.
- È possibile ritirarsi senza penalità.
- Non è possibile lasciare l'aula conservando il tema della prova in corso.
- Tempo a disposizione: **2h30m**

Valore indicativo degli esercizi, voti parziali e voto finale:

Esercizio 1	6 punti	_____
Esercizio 2	7 punti	_____
Esercizio 3	6 punti	_____
Esercizio 4	13 punti	_____
Totale(32)		_____

Esercizio 1 - Algebra di Boole, Aritmetica Binaria, Codifica delle Informazioni (7 punti)

Punteggio totale quesiti a risposta multipla: 4 punti. Ogni domanda può avere da 0 a 4 risposte corrette. Risposte errate comporteranno rimozione di 0,5 punti.

- (a) Qual è il risultato dell'espressione logica NOT(A AND B) OR (A AND B)

- ☒ Un valore logico VERO
☐ A AND B
☐ A OR B
☐ NOT A or NOT B or NOT (A OR B)

$$\overline{AB} + AB$$

$$\overline{C} + C = 1$$

- (b) Qual è il risultato dell'espressione logica NOT A AND B OR A AND NOT B AND (A OR C)

- ☒ Non dipende da C
☐ Un valore logico FALSO
☒ NOT A AND B OR A AND NOT B
☐ A OR NOT B AND NOT A OR B

$$\overline{A}B + A\overline{B}(A+C) = \overline{A}B + \overline{A}B\cancel{A} + A\overline{B}C = \overline{A}B + A\overline{B}(1+C)$$

- (c) Qual è il risultato dell'espressione logica NOT (NOT (NOT A OR B) OR NOT B AND NOT C)

- ☐ Un valore logico che dipende solo da A e B
☐ B
☒ NOT A AND C OR B
☐ Nessuna delle precedenti

$$\overline{[(\overline{A+B}) + \overline{B}C]} = \overline{(\overline{A+B})} \cdot \overline{\overline{B}C} = (A+B) \cdot B\overline{C} = (A+B) \cdot (B+C) = \overline{A}B + B\overline{A} + \overline{A}C + BC$$

$B(\overline{A}+1+C) + \overline{A}C$

$= \overline{A}C + B$

- (d) Ho un insieme contenente n=32 numeri interi casuali tra 0 e 128. Quanti bit mi servono per codificarne gli elementi?

- ☐ log2 n-1 bit
☒ 5 bit
☐ 8 bit
☐ 9 bit

$$h = 32$$

$$2^5 = 32$$

- (e) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri

$$A = -21_{\text{dec}}$$

$$B = 43_{\text{dec}}$$

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.

Formattare le risposte con spazi in modo che siano leggibili.

Non verranno accettate risposte senza procedimento. (1 punto)

$\begin{array}{r l} +21 & 1 \\ 10 & 0 \\ 5 & 1 \\ 2 & 0 \\ 1 & 1 \\ 0 & \end{array}$	$\begin{array}{c} 010101 \\ \downarrow \\ -21 = 101011 \\ \\ A+B = \begin{array}{r} 1101011 \\ 0101011 \\ \hline 0010110 \\ \text{OK} \end{array} \end{array}$	$\begin{array}{r l} 43 & 1 \\ 21 & 1 \\ 10 & 0 \\ 5 & 1 \\ 2 & 0 \\ 1 & 1 \\ 0 & \end{array}$	$\begin{array}{c} 43_2 = 0101011 \\ -21_2 = 101011 \\ \\ A-B = \begin{array}{r} 1101011 \\ 1010101 \\ \hline 1000000 \\ \text{OK} \end{array} \end{array}$
--	--	---	--

- (f) 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. Riportare le operazioni di conversione (1 punto).

$\begin{array}{r l} 29 & 1 \\ 14 & 0 \\ 7 & 1 \\ 3 & 1 \\ 1 & 1 \\ 0 & \end{array}$	$\begin{array}{c} 11101 \\ 11101,101 \end{array}$	$\begin{array}{c} 0,625 = 0,5 + 0,125 \\ 0,101 \\ \\ n = 1101101 \dots \\ s = 0 \\ E = 127 + 6 = 133 \quad 10000011 \end{array}$
---	---	--

Esercizio 2 - Programmazione C (7 punti)

Si scriva un sottoprogramma che, ricevendo in ingresso un vettore ove ogni cella e' di tipo

```
struct sq {  
    char nome[15];  
    int ptn;  
}
```

restituisce la stessa struttura in ordine decrescente relativamente al campo nome. Si puo' fare uso della funzione *int strcmp(char *, char *)* che restituisce un valore negativo se la prima stringa e' minore -in termini alfabetici- della seconda, 0 se uguale e un valore positivo se maggiore.

Il sottoprogramma restituisce inoltre il valore massimo di ptn all'interno del vettore.

Risposta:

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

typedef struct sq {
    char name[15];
    int ptn;
} sq;

int funzione(sq array[], int len);
int main(int argc, const char * argv[]) {
    sq vettore[5];

    strcpy(vettore[0].name, "Ciccio");
    vettore[0].ptn = 1;
    strcpy(vettore[1].name, "Ananas");
    vettore[1].ptn = 12;
    strcpy(vettore[2].name, "Topolino");
    vettore[2].ptn = 23;
    strcpy(vettore[3].name, "Pluto");
    vettore[3].ptn = -10;
    strcpy(vettore[4].name, "Banana");
    vettore[4].ptn = 4;

    printf("Max = %d\n", funzione(vettore, 5));
    for (int i=0; i<5; i++){
        printf("%s %d\n", vettore[i].name, vettore[i].ptn);
    }
    return 0;
}

int funzione(sq array[], int len){
    int i,j;
    int ptn;
    sq temp;
    int swap = 1;

    //Calcolo valore massimo
    for (i=0; i<len; i++)
    {
        if (i==0)
            ptn = array[i].ptn;
        else{
            if (array[i].ptn>ptn)
                ptn = array[i].ptn;
        }
    }

    //Ordina array - bubblesort
    for (i=0; i<len-1 && swap == 1; i++){
        swap = 0;
        for (j=0; j<len-i-1; j++){
            if (strcmp(array[j].name, array[j+1].name)<0){
                temp = array[j];
                array[j] = array[j+1];
                array[j+1] = temp;
                swap = 1;
            }
        }
    }
    return ptn;
}
```

Esercizio 3 - Matlab (6 punti)

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

1. Creare una matrice quadrata A di dimensione 4 x 7 contenente lo stesso numero casuale tra 1 e 10 (1 riga - 1 punto)

$$A = \text{randi}([1, 10]) \cdot \text{ones}(4, 7)$$

2. Moltiplicare a tutte le celle della matrice, un numero casuale tra -5 e 5. Ogni cella va moltiplicata per un numero differente (1 riga - 1 punto)

$$A = A \cdot \text{rand}([-5, 5], \text{size}(A))$$

3. Calcolare la somma totale della sottomatrice matrice 4x4 (1 riga - 1 punto)

$$\text{sum}(\text{sum}(A)) \quad \text{oppure} \quad \text{sum}(A, 'all')$$

4. Eliminare le colonne della matrice A la cui media delle componenti è minore di 0. (1 riga - 2 punti)

$$A(:, \text{mean}(A, 1) < 0) = []$$

5. Scrivere una funzione Matlab che presa in ingresso la matrice, restituisca due vettori. Il primo vettore contiene la prima e l'ultima riga della matrice, il secondo vettore contiene la seconda e la terza riga. (2 punti)

$$\begin{aligned} &\text{FUNCTION } x, y = \text{FUNZIONE}(A) \\ &x = [A(1, :) \ A(\text{end}, :)]'; \\ &y = [A(2, :) \ A(3, :)]' \\ &\text{END} \end{aligned}$$

Esercizio 4 - Programmazione C - Liste (13 punti)

Si vuole creare un software per la gestione dei pazienti in attesa di essere operati. Ogni paziente è descritto da: codice paziente (decida lo studente come strutturarli), urgenza di operazione (in un range da 0 a 10), tempo richiesto dall'operazione in minuti, età del paziente.

La lista è ordinata in base all'urgenza e a parità di urgenza in base all'età.

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

B. Scrivere una funzione che presa in ingresso la lista, restituisca l'età media dei pazienti in attesa di operazione e tutti i dati relativi al paziente più anziano (3 punti).

C. Scrivere una funzione che, presi in ingresso i dati del paziente, inserisca in modo ordinato un nuovo paziente nella lista. Si ricorda che la lista è ordinata in base all'urgenza e a parità di urgenza in base all'età (4 punti).

D. Scrivere una funzione ricorsiva che, presa in ingresso la lista dei pazienti, il tempo in cui è libera la sala operatoria, restituisca il codice del paziente con priorità più alta, che può essere operato nel tempo specificato, e cancelli l'elemento del paziente selezionato dalla lista (5 punti).

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

typedef struct node{
    int id_paziente;
    int urgenza;
    int tempo_operazione;
    int eta;
    struct node *next;
} node;

typedef node* ptrNode;

float mediaEtaPaziente(ptrNode lista, ptrNode *anziano);
ptrNode aggiungiPaziente(ptrNode lista, int id_paziente, int urgenza, int tempo, int eta);
ptrNode trovaPaziente(ptrNode lista, int *id, int tempoSala);
void stampaLista(ptrNode lista);

int main(int argc, const char * argv[]) {
    ptrNode lista = NULL;
    ptrNode anziano;
    int id;

    lista = aggiungiPaziente(lista, 1, 2, 10, 80);
    lista = aggiungiPaziente(lista, 2, 4, 20, 70);
    lista = aggiungiPaziente(lista, 3, 10, 30, 50);
    lista = aggiungiPaziente(lista, 4, 10, 40, 20);
    lista = aggiungiPaziente(lista, 5, 7, 50, 40);
    stampaLista(lista);

    printf("Media eta: %.2f\n", mediaEtaPaziente(lista, &anziano));
    printf("%d eta: %d %d %d\n", anziano->id_paziente, anziano->eta, anziano->tempo_operazione, anziano->urgenza);

    lista = trovaPaziente(lista, &id, 30);
    printf("Id da operare: %d\n", id);
    stampaLista(lista);

    return 0;
}
```

```

void stampaLista(ptrNode lista){
    if (lista==NULL)
        return;
    printf("ID: %d\nEta: %d\nUrgenza: %d\nTempo: %d\n\n", lista->id_paziente, lista->eta, lista->urgenza, lista->tempo_operazione);
    stampaLista(lista->next);
}

float mediaEtaPaziente(ptrNode lista, ptrNode *anziano){
    float media = 0;
    int cont = 0;
    int etaMax;

    if (lista!=NULL){
        etaMax = lista->eta;
        *anziano = lista;
    }
    for (; lista!=NULL; lista = lista->next){
        media+=lista->eta;
        cont++;
        if (lista->eta>etaMax){
            etaMax = lista->eta;
            *anziano = lista;
        }
    }
    return media/cont;
}

ptrNode aggiungiPaziente(ptrNode lista, int id_paziente, int urgenza, int tempo, int eta){
    ptrNode temp;

    if ((lista==NULL) || (lista->urgenza<urgenza) || ((lista->urgenza == urgenza) && (lista->eta<eta))){
        temp = (ptrNode)malloc(sizeof(node));
        temp->id_paziente = id_paziente;
        temp->eta = eta;
        temp->tempo_operazione = tempo;
        temp->urgenza = urgenza;
        temp->next = lista;
        return temp;
    }
    else{
        lista->next = aggiungiPaziente(lista->next, id_paziente, urgenza, tempo, eta);
        return lista;
    }
}

ptrNode trovaPaziente(ptrNode lista, int *id, int tempoSala){
    ptrNode temp;
    if (lista == NULL)
    {
        *id = -1;
        return NULL;
    }
    if (lista->tempo_operazione>=tempoSala)
    {
        *id = lista->id_paziente;
        temp = lista;
        lista = lista->next;
        free(temp);
        return lista;
    }
    else{
        return trovaPaziente(lista->next, id, tempoSala);
    }
}

```

NOME e COGNOME: _____



Politecnico di Milano

Dipartimento di Elettronica, Informazione e Bioingegneria

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

Cognome: _____ Matricola: _____
Nome: _____ Firma: _____

Istruzioni

- Non separate questi fogli. Scrivete la soluzione **solo sui fogli distribuiti**, utilizzando il retro delle pagine in caso di necessità. **Cancellate le parti di brutta** con un tratto di **penna**.
- Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
- **NON è possibile scrivere a matita.**
- Scrivere nome e cognome su tutti i fogli. Non verranno corretti compiti non firmati o con nome illeggibile
- È **vietato** utilizzare **calcolatrici, telefoni o pc**. Chi tenti di farlo vedrà **annullata** la sua prova.
- **Non è ammessa la consultazione di libri e appunti.**
- Qualsiasi **tentativo** di comunicare con altri studenti comporta **l'espulsione** dall'aula.
- È possibile ritirarsi senza penalità.
- Non è possibile lasciare l'aula conservando il tema della prova in corso.
- Tempo a disposizione: **2h**

Valore indicativo degli esercizi, voti parziali e voto finale:

Esercizio 1	6 punti	_____
Esercizio 2	7 punti	_____
Esercizio 3	7 punti	_____
Esercizio 4	12 punti	_____
Totale(32)		_____

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

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

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

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

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

$$AB + C(\overline{A}\overline{B}) + C$$

$$AB + C(\overline{A}\overline{B} + 1)$$

$$AB + C$$

- (b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri $A = 35_{\text{dec}}$ e $B = -47$, li si converta, se ne calcolino la somma ($A+B$) e la differenza ($A-B$) in complemento a due e si indichi se si genera riporto sulla colonna dei bit più significativi e se si verifica overflow. Non si accetteranno soluzioni senza il procedimento. (2 punto)

7 bit minimo

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

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

35		1
17		1
8		0
4		0
2		0
1		1
0		

$$\begin{array}{r} 0100011 \\ 1011101 \\ \hline \end{array}$$

47		1
23		1
11		1
5		1
2		0
1		1
0		

$$\begin{array}{r} 0101111 \\ 1010001 \\ \hline \end{array}$$

$A+B$

$$\begin{array}{r} 0100011 \\ 1010001 \\ \hline 1110100 \end{array}$$

$A-B$

$$\begin{array}{r} 0100011 \\ 0101111 \\ \hline 1010010 \end{array}$$

OF

NO CR

- (c) Si converta il numero 24,3 in virgola fissa e in virgola mobile con codifica IEEE 754 con precisione singola. Non si accetteranno soluzioni senza il procedimento. (2 punto)
 $24 = 11000$ $0.3 = 0.01001$ con 1001 parte periodica
 virgola fissa = $24.0100110011001\dots$ 1001 periodico virgola mobile IEEE754 esponente = $127+4$
 $= 131 = 10000011$ mantissa = 10000100110011001100110 segno = 0

24		0
12		0
6		0
3		1
1		1
0		

0,3		0,6
0,6		1,2
0,2		0,4
0,4		0,8
0,8		1,6
0,6		

11000

0,01001

11000,01001

S = 0

P = 10000100110011001100110

E = $127+4 = 131 = 10000011$

Esercizio 2 - Teoria (7 punti)

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

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

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

    funz(val);
    return 0;
}
```

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

160 10 10

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

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

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

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

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

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

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

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

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

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

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

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

oppure

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

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

NOME e COGNOME: _____

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

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

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

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

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

Ad esempio:

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

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

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

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

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

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

a = 7

e = 2

i = 1

o = 0

u = 8

Le stringhe devono essere anche convertite tutte in caratteri maiuscoli.

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

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

typedef node* ptrNode;

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

```

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

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

    if (testa == NULL)
        return nuovo;

    ptrNode cursor = testa;
    ptrNode prec = NULL;

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

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

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

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

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

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

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

```

```

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

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

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

```



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	_____
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)

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

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: $(a \text{ and } b) \text{ or } (\text{not } a \text{ and } c)$

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 $A = 46_{\text{dec}}$ 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:

$46_{\text{d}} = 101110_{\text{b}}$ - 6bit + 1 di segno = 7bit

cp2 = 0101110b

$-17_{\text{d}} = 10001_{\text{b}}$ - 5 bit + 1 segno = 6bit

cp2 = 1101111b - cp2 7bit

A + B

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

A - B

riporto		Né riporto perduto, né overflow
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)x4+3]x(2+4)+(3-6)/4	[(3-2)x4+3]x(2+4) + (3-6)/4
[(3-2)x4+3]x(2+4))+(3-6)/4	5+6-[8x3]
5+6-[8x3]	7x8-(3x2)
7x8-(3x2)	

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);

```

}

