

	Politecnico di Milano Scuola di Ingegneria Industriale e dell'Informazione <b>INFORMATICA B</b> Appello del 22/1/2024		COGNOME E NOME
	Fila	Colonna	MATRICOLA

- Il presente plico contiene 3 esercizi e 2 domande e **deve essere debitamente compilato con cognome e nome, e numero di matricola.**
- Il tempo a disposizione è di 1 ora e 30 minuti.
- 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 (o ripudiate) con un tratto di penna.
- Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
- È possibile scrivere a matita (e non occorre ricalcare al momento della consegna!).
- **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.**
- **Per il superamento dell'esame è necessario dimostrare sufficienti competenze sia in C sia in Matlab, e quindi saper impostare correttamente esercizi in entrambi i linguaggi.**
- **MOLTO IMPORTANTE: risposte poco leggibili** (scritte molto piccolo, con calligrafia poco comprensibile, o molto disordinate) **non saranno considerate nella valutazione.**

## **Esercizio 1 (10 punti)**

Si considerino le seguenti strutture dati, in linguaggio C, necessarie per definire una bambola. Ogni bambola ha un nome, un anno di fabbricazione, un valore economico e degli accessori (di diversa tipologia e colore), fino ad un massimo di 5. La struttura ha, inoltre, un campo (binario) per specificare se possediamo o non la bambola (0 non possediamo la bambola, 1 la possediamo).

```
#DEFINE LEN 20
#DEFINE MAX_ACC 5
typedef enum Colore {rosso, blu, giallo, nero, marrone};
typedef enum Tipologia {borsa, cappello, scarpe};
typedef enum Posseduto {false, true};

typedef struct {
    Tipologia tipo;
    Colore colore;
} Accessorio;

typedef struct {
    char nome[LEN+1];
    int anno;
    float valore;
    Accessorio accessori[MAX_ACC];
    int n_acc;
    Posseduto posseduto;
} Bambola;
```

### **Domanda 1.1 (punti 2)**

Dichiarare un vettore `catalogo` per contenere un catalogo di esattamente 500 bambole. Supponendo che il catalogo sia stato precedentemente riempito, scrivere un frammento di codice per calcolare il valore totale delle bambole possedute (quelle non possedute non devono essere considerate nel calcolo).

### **Domanda 1.2 (punti 4)**

Scrivere un frammento di codice che, per ogni anno dal 1950 ad oggi, stampi a video il nome della bambola che ha più valore. Nel caso non ci siano bambole per un anno, si stampi a schermo il messaggio "Nessuna bambola prodotta nell'anno XXXX", sostituendo l'anno a XXXX.

### **Domanda 1.3 (punti 2)**

Modificare le strutture precedentemente date per considerare che, oltre ad accessori, le bambole possano avere **al più** un mezzo e **al più** un edificio associati. Ogni mezzo avrà un colore e potrà essere di tipologia auto, moto, camper o motoscafo. Ogni edificio dovrà essere di tipologia casa al mare, chalet, attico o loft.

### **Domanda 1.4 (punti 2)**

Considerando le strutture dati originali, senza le modifiche introdotte per rispondere al punto 1.3, verificare che ognuna delle bambole possedute non abbia due accessori uguali. Per esempio, se una bambola ha una borsa marrone tra gli accessori, essa può avere altre borse a patto che non siano marroni. Nel caso in cui una bambola abbia più accessori uguali, scrivere a schermo il suo nome e l'anno di fabbricazione. Nel caso in cui non ci siano bambole all'interno del catalogo che abbiano due accessori uguali, stampare a schermo il messaggio "Nessuna bambola ha oggetti duplicati".

## Soluzione

1.1

```
#define NUM_BAMBOLE 500

Bambola catalogo[NUM_BAMBOLE];

float valoreTotalePossedute = 0.0;
int i;

for (i = 0; i < NUM_BAMBOLE; ++i) {
    if (catalogo[i].posseduto) {
        valoreTotalePossedute += catalogo[i].valore;
    }
}
```

1.2

```
float maxValorePerAnno;
char nomeBambola[LEN+1];
int i, j;

for (i = 1950; i <= 2023; i++) {
    maxValorePerAnno = -1;
    for (j = 0; j < NUM_BAMBOLE)
        if(catalogo[j].anno == i && catalogo[j].valore > maxValorePerAnno) {
            maxValorePerAnno = catalogo[j].valore;
            strcpy(nomeBambola, catalogo[j].nome);
        }
}

if (maxValorePerAnno > -1)
    printf("Anno %d: %s\n", i, nomeBambola);
else
    printf("Nessuna bambola prodotta nell'anno %d\n", i);
}
```

1.3

```
typedef enum TipologiaVeicolo {auto, moto, camper, motoscafo};
typedef enum TipologiaEdificio {casa_al_mare, chalet, attico, loft};

typedef struct {
    TipologiaVeicolo tipoVeicolo;
    Colore coloreVeicolo;
} Mezzo;

typedef struct {
    char nome[LEN + 1];
    int anno;
    float valore;
    Accessorio accessori[MAX_ACC];
    int n_acc;
    Posseduto posseduto;
    Mezzo mezzo; // Aggiunto il campo per il mezzo
    Posseduto posseduto_mezzo; // Aggiunto il campo per il mezzo
    TipologiaEdificio edificio; // Aggiunto il campo per l'edificio
    Posseduto posseduto_edificio; // Aggiunto il campo per l'edificio
} Bambola;
```

1.4

```
int i, j, k, flag_duplicati = 0;
```

```

for (i = 0; i < NUM_BAMBOLE; i++) {
    if (catalogo[i].posseduto) {
        for (j = 0; j < catalogo[i].n_acc - 1; j++) {
            for (k = j + 1; k < catalogo[i].n_acc; k++) {
                if(catalogo[i].accessori[j].tipo == catalogo[i].accessori[k].tipo &&
                    catalogo[i].accessori[j].colore == catalogo[i].accessori[k].colore)
                {
                    printf("Bambola con accessori duplicati: %s (Anno: %d)\n",
                        catalogo[i].nome, catalogo[i].anno);
                    flag_duplicati = 1;
                }
            }
        }
    }
}

if (flag_duplicati == 0)
    printf("Nessuna bambola ha oggetti duplicati");

```

## **Esercizio 2 (10 punti)**

Il prof. Cloisenheimer vuole sviluppare una tecnica per ottenere energia dal processo fisico della fusione. Allo stesso tempo, il professore è molto sensibile ai temi ambientali e desidera che tale processo non produca dei sottoprodotti radioattivi. Dopo aver eseguito numerosi esperimenti, vuole capire quanto è affidabile il processo da lui sviluppato.

### **Domanda 2.1 (punti 4)**

Si scriva una funzione **calcolaAffidabili** in linguaggio Matlab che, dato il vettore numerico **energia**, contenente i megawatt ottenuti da differenti esperimenti, e il vettore numerico **sottoprodotto**, contenente il numero atomico del sottoprodotto creato dalla reazione di fusione, restituisca uno scalare **affidabilita**, che contiene la percentuale di esperimenti andati a buon fine, e un vettore logico **indici\_aff**, che abbia valore logico vero in posizione i-sima se l'esperimento che ha determinato i valori delle variabili **energia** e **sottoprodotto** nella stessa posizione i-sima è andato a buon fine. Un esperimento è andato a buon fine se ha prodotto un numero di megawatt inferiore a 50 e se il sottoprodotto non è né tecnezio (numero atomico 43) né promezio (numero atomico 61) e non ha numero atomico maggiore di 82.

Ad esempio, se i vettori in ingresso sono: **energia** = [42 25 70 46 51], e **sottoprodotto** = [22 43 13 84 33] la funzione deve restituire che solo il 20% degli esperimenti è andato a buon fine (**affidabilita** = 20) e il vettore degli indici è **indici\_aff** = [vero, falso, falso, falso, falso].

### **Domanda 2.2 (punti 2)**

Purtroppo, i militari hanno preso il controllo degli esperimenti del prof. Cloisenheimer per sfruttarli per scopi bellici. In questo caso, i militari vogliono selezionare gli esperimenti distruttivi, ossia, quelli che generano molta energia e anche sottoprodotti radioattivi. Scrivere in linguaggio Matlab una funzione **calcolaDistruttivi** che, dati i vettori **energia** e **sottoprodotto**, restituisca la percentuale degli esperimenti distruttivi **distr** e un vettore **indici\_distr** di valori logici corrispondenti agli esperimenti distruttivi. Un esperimento è definito come distruttivo se non è affidabile secondo la definizione data sopra. Ad esempio, con i vettori riportati sopra avremmo come output **distr** = 80 e **indici\_distr** = [falso, vero, vero, vero, vero]. **Nota:** per sviluppare la funzione **calcolaDistruttivi** si utilizzi la funzione **calcolaAffidabili** definita al punto precedente.

### **Domanda 2.3 (punti 4)**

Scrivere uno script in linguaggio Matlab che svolga le seguenti operazioni:

- Chieda all'utente due vettori relativi all'energia e ai sottoprodotti radioattivi, controlli che abbiano dei valori validi (non negativi e, per i sottoprodotti, interi) e che abbiano le stesse dimensioni. In caso contrario, stampi a schermo un messaggio di errore e chieda nuovamente i vettori all'utente, finché dei valori validi siano stati inseriti;
- Nel caso in cui i vettori contengano valori validi, chiami la funzione **calcolaAffidabili**, memorizzi i valori restituiti dalla funzione in opportune variabili, e stampi a schermo la percentuale degli esperimenti andati a buon fine.
- Crei un plot che visualizzi una serie di punti le cui ascisse sono gli indici degli esperimenti andati a buon fine e le cui ordinate sono i valori di energia relativi a tali esperimenti. I punti non devono essere uniti da alcuna linea e il marcatore usato deve essere un cerchio verde.
- Sullo stesso plot, aggiunga i punti corrispondenti agli indici e ai valori di energia relativi agli esperimenti non andati a buon fine, utilizzando come marcatore una croce rossa.

## Soluzione

2.1

```
function [affidabilita, indici_aff] = calcolaAffidabili(energia, sottoprodotto)
    indici_aff = energia < 50 & ~(sottoprodotto == 43 | sottoprodotto == 61 | sottoprodotto > 82);
    affidabilita = mean(indici_aff) * 100;
```

2.2

```
function [distr, indici_distr] = calcolaDistruttivi(energia, sottoprodotto)
    [affidabilita, indici_aff] = calcolaAffidabili(energia, sottoprodotto)
    distr = 100 - affidabilita;
    indici_distr = ~indici_aff;
```

2.3

clear

clc

close all

```
energia = input('Inserisci il vettore dell"energia: ');
```

```
sottoprodotto = input('Inserisci il vettore dei sottoprodotti: ');
```

```
while length(energia) ~= length(sottoprodotto) || any(energia < 0) || any(sottoprodotto < 0) ||
any(sottoprodotto ~= round(sottoprodotto))
```

```
    disp("I valori devono essere non negativi e, per i sottoprodotti, interi.");
```

```
    energia = input('Inserisci il vettore dell"energia: ');
```

```
    sottoprodotto = input('Inserisci il vettore dei sottoprodotti: ');
```

else

```
[affidabilita, indici_aff] = calcolaAffidabili(energia, sottoprodotto);
```

```
disp(['Percentuale esperimenti andati a buon fine: ' num2str(affidabilita) '%']);
```

```
figure();
```

```
plot(find(indici_aff), energia(indici_aff), 'go'); % Punti andati a buon fine (cerchi verdi)
```

```
hold on;
```

```
plot(find(~indici_aff), energia(~indici_aff), 'rx'); % Punti non andati a buon fine (croci rosse)
```

end

### **Esercizio 3 (6 punti)**

#### **Domanda 3.1 (punti 4)**

Si consideri la seguente condizione logica:

$$!((!A \ || \ B \ || \ C) \ \&\& \ (A \ || \ !B \ || \ !C))$$

e se ne calcoli la tabella di verità.

#### **Domanda 3.2 (punti 2)**

Supponendo che in un programma siano definite le variabili booleane `auto_economica`, `auto_basso_consumo`, `auto_ecologica` e che i letterali A, B e C usati nella domanda precedente corrispondano alle seguenti espressioni:

A: `auto_economica = true`

B: `auto_basso_consumo = true`

C: `auto_ecologica = true`

Dire se la precedente espressione è equivalente a quella che segue (due espressioni logiche sono equivalenti se, quando vengono valutate, risultano nello stesso valore di verità):

```
(auto_economica && !auto_basso_consumo && !auto_ecologica) ||  
(!auto_economica && auto_basso_consumo && auto_ecologica).
```

Motivare la risposta.

## Soluzione

3.1

A	B	C		$\neg((\neg A \vee \neg B \vee \neg C) \wedge (A \vee \neg B \vee \neg C))$
1	1	1		0
1	1	0		0
1	0	1		0
1	0	0		1
0	1	1		1
0	1	0		0
0	0	1		0
0	0	0		0

3.2

Si può verificare che le due condizioni sono equivalenti in quanto la tabella di verità è la medesima o utilizzando le formule di De Morgan.



### **Domanda 1 (3 punti)**

Si consideri il numero  $-976$ . Si dica quali tra le seguenti affermazioni sono corrette e perché (è possibile che più affermazioni, anche tutte, siano corrette):

- Il numero è rappresentabile in complemento a due utilizzando 11 bit.
- $11000000101101$  è una rappresentazione in complemento a due corretta per il numero  $-976$ .

Risposte prive di giustificazione non verranno considerate. Nota: per rispondere non è necessario sviluppare i calcoli.

### **Domanda 2 (3 punti)**

Si consideri un sistema operativo multiprocesso. Si scriva un semplice programma il cui processo necessita un passaggio dallo stato di **attesa**, motivando l'esempio.

Esiste la possibilità che questo processo non passi mai dallo stato di **pronto**?