

	Politecnico di Milano Scuola di Ingegneria Industriale e dell'Informazione INFORMATICA B Appello del 7/09/2023		COGNOME E NOME									
	Fila	Colonna	MATRICOLA									
<table border="1" style="display: inline-table;"> <tr> <td style="width: 20px; height: 20px;"></td> </tr> </table>												

- 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 45 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 – Linguaggio C (10 punti)

Un'azienda vende prodotti farmaceutici e possiede un magazzino con 100 armadi, ciascuno contenente 50 cassetti tutti della stessa dimensione che può essere grande, media oppure piccola. Ciascun prodotto farmaceutico è caratterizzato da un codice numerico univoco, un nome, una casa produttrice, le dimensioni definite in termini di altezza, larghezza e lunghezza della scatola, tutte espresse in cm. Quando l'azienda riceve un nuovo prodotto lo immagazzina in uno dei cassetti disponibili in magazzino a seconda delle sue dimensioni.

Valgono i seguenti vincoli:

- Ogni prodotto non può avere nessuna dimensione superiore a 20 cm.
- Se un prodotto ha almeno una dimensione maggiore di 15 cm, deve essere immagazzinato in un cassetto grande
- Se un prodotto ha almeno una dimensione compresa tra 10 cm e 15 cm deve essere immagazzinato in un cassetto medio
- Se un prodotto ha tutte le dimensioni inferiori a 10 cm deve essere immagazzinato in un cassetto piccolo.
- Un cassetto può contenere un unico prodotto. È quindi possibile immagazzinare un prodotto in un cassetto solo se il cassetto è vuoto.

C.1 (4 punti): Si definiscano in linguaggio C le strutture dati idonee a rappresentare:

- un prodotto;
- un cassetto (si ricordi che un cassetto può essere grande, piccolo o medio, può essere pieno o vuoto e, se pieno, contiene le informazioni associate a un prodotto);
- un armadio (costituito dall'insieme dei suoi cassetti e da un campo tipo che può essere grande, medio o piccolo, a seconda della dimensione dei cassetti nell'armadio);
- l'intero magazzino m ;

Si definisca anche ogni altro tipo di dato necessario per completare la definizione delle strutture dati descritte sopra.

Si dichiarino, inoltre, la variabile m che rappresenta l'intero magazzino.

C.2 (4 punti): Si scriva in linguaggio C un frammento di codice che acquisisce dall'utente:

- le informazioni relative a un prodotto,
- il numero dell'armadio e il numero del cassetto da utilizzare per memorizzare il prodotto,

e memorizza le informazioni del prodotto nel cassetto, previo controllo che i vincoli descritti sopra siano verificati.

Si assuma che il magazzino sia parzialmente pieno e che quindi m sia stata già opportunamente inizializzata, contenga armadi con alcuni cassetti già pieni di prodotti e altri vuoti e l'inizializzazione sia corretta, cioè un armadio grande contiene solo cassetti grandi, un armadio medio solo cassetti medi e un armadio piccolo solo cassetti piccoli. Il frammento di codice, quindi, deve calcolare la dimensione del cassetto necessaria per l'immagazzinamento del prodotto, deve verificare che il cassetto prescelto sia vuoto e di dimensione opportuna per il prodotto, e deve poi aggiornare correttamente le strutture dati.

C.3 (2 punti): Si scriva in linguaggio C un frammento di codice che acquisisce dall'utente il codice univoco di un prodotto e, se il prodotto corrispondente è presente in magazzino, svuota il cassetto corrispondente e stampa per l'utente tutte le informazioni relative al prodotto. Nel caso in cui il prodotto non sia presente in magazzino, il frammento di codice deve stampare il messaggio: "Prodotto non presente in magazzino".

Soluzione

```
C.1)
#define N_ARMADI 100
#define N_CASSETTI 50
#define MAX_NUM_CHAR 30
typedef char stringa[MAX_NUM_CHAR];
typedef enum {falso, vero} booleano;
typedef enum {piccolo, medio, grande} dimensione;
typedef struct {
    int codice;
    stringa nome;
    stringa casaProd;
    float lunghezza;
    float larghezza;
    float altezza;
} prodotto;

typedef struct {
    booleano vuoto;
    dimensione d;
    prodotto p;
} cassetto;

typedef struct {
    cassetto c[N_CASSETTI];
    dimensione d;
} armadio;

typedef armadio magazzino[N_ARMADI];

int main()
{
    magazzino m;
}

C.2)
    prodotto p;
    dimensione dProdotto;

    int i, j;

    /* assumiamo che m sia stato correttamente inizializzato */
    /* acquisizione dei dati di un prodotto */
    printf("Inserisci i dati di un prodotto: \n");
    printf("inserisci il codice univoco: ");
    scanf("%d", &(p.codice));
    printf("inserisci il nome del prodotto: ");
    scanf("%s", p.nome);
    printf("inserisci il nome della casa produttrice: ");
    scanf("%s", p.casaProd);
    do {
        printf("inserisci la lunghezza della scatola: ");
        scanf("%f", &(p.lunghezza));
        printf("inserisci la larghezza della scatola: ");
        scanf("%f", &(p.larghezza));
        printf("inserisci l'altezza della scatola: ");
        scanf("%f", &(p.altezza)); }
    while(p.lunghezza > 20 || p.larghezza > 20 || p.altezza > 20);
    /* acquisizione del numero di armadio e di cassetto */
    do {
        printf("inserisci il numero di armadio: ");
        scanf("%d", &i); }
    while(i < 0 || i >= N_ARMADI);
    do {
        printf("inserisci il numero di cassetto: ");
        scanf("%d", &j); }
    while(j < 0 || j >= N_CASSETTI);
```

```

/* individuazione dimensione cassetto */
if (p.lunghezza > 15 || p.larghezza > 15 || p.altezza > 15)
    dProdotto = grande;
else if (p.lunghezza > 10 || p.larghezza > 10 || p.altezza > 10)
    dProdotto = medio;
else dProdotto = piccolo;

/* verifica che il cassetto sia vuoto e di dimensioni adeguate */
if (m[i].c[j].vuoto == vero && m[i].c[j].d == dProdotto)
{
    m[i].c[j].vuoto = falso;
    m[i].c[j].p = p;
}
else
    printf("Il prodotto non può essere immagazzinato\n");

```

C.3)

```

int codice;
booleano trovato;

printf("Inserisci il codice del prodotto da cercare: ");
scanf("%d", &codice);
trovato = falso;
for (i = 0; i < N_ARMADI && !trovato; i++)
    for (j = 0; j < N_CASSETTI && !trovato; j++)
    {
        if (!m[i].c[j].vuoto && m[i].c[j].p.codice == codice)
        {
            printf("Prodotto codice %d trovato", codice);
            printf("Nome: %s\n", m[i].c[j].p.nome);
            printf("casa Produttrice: %s\n", m[i].c[j].p.casaProd);
            printf("Lunghezza: %f, larghezza: %f, altezza: %f\n", m[i].c[j].p.lunghezza,
                m[i].c[j].p.larghezza, m[i].c[j].p.altezza);
            m[i].c[j].vuoto = vero;
            trovato = vero;
        }
    }
if (!trovato)
    printf("Prodotto codice %d non presente in magazzino", codice);
} /* chiude il main */

```

Esercizio 2 – MATLAB (10 punti)

Un'azienda offre un servizio di monitoraggio della pressione arteriosa per 24 ore (Holter pressorio). Il monitoraggio avviene attraverso un apparecchio che esegue una misurazione della pressione massima e minima ogni 15 min e memorizza i dati in un file. Una volta terminato il monitoraggio, l'azienda acquisisce il file e lo elabora fornendo al paziente un referto.

M.1 (5 punti): Si sviluppi una **funzione** in MATLAB che analizzi i dati di pressione e determini se il paziente ha un profilo normale, anomalo o critico. La funzione riceve come parametri i due array, `p_max` e `p_min` contenenti, rispettivamente, i dati di pressione massima e minima di un paziente organizzati secondo l'ordine temporale con cui sono stati acquisiti (per esempio, la cella 1 di `p_max` contiene la prima misurazione di pressione massima e, analogamente, la cella 1 di `p_min` contiene la prima misurazione di pressione minima). I due array hanno la stessa dimensione.

La funzione determina il profilo del paziente secondo le seguenti regole:

- Se esiste almeno un caso in cui 8 misurazioni rilevano una pressione massima superiore al valore 130, il profilo è critico.
- Altrimenti, se il profilo non risulta critico ma il paziente ha una differenza tra pressione massima e minima superiore a 40 per il 50% o più delle misurazioni, allora il profilo è anomalo.
- In tutti gli altri casi, il profilo è normale.

La funzione restituisce al chiamante i seguenti dati:

- Un numero intero, corrispondente al profilo del paziente, che assumerà valore 1 nel caso di profilo normale, 2 nel caso di profilo anomalo e 3 nel caso di profilo critico.
- Un array contenente i valori degli indici corrispondenti alle celle in `p_max` i cui valori hanno determinato la definizione dello specifico profilo per il paziente. Nel caso di paziente con profilo normale, questo array sarà vuoto.

M.2 (5 punti): si sviluppi uno script in MATLAB che svolge le seguenti operazioni:

- Legge dal file MATLAB `holter.mat` i dati relativi alle misurazioni di pressione massima e minima eseguite per un paziente.
- Chiama la funzione definita al punto precedente passando come parametri gli array caricati dal file.
- Mostra all'utente un diagramma con i valori di pressione massima e minima acquisiti da file nei vari momenti di misurazione. Le due linee visualizzate devono avere colore verde nel caso in cui il paziente ha un profilo normale (sulla base di quanto determinato dalla funzione chiamata), colore blu nel caso di profilo anomalo, colore rosso nel caso di profilo critico.
- Nel caso di profilo critico o anomalo, si evidenzino nel diagramma con un asterisco i valori di pressione massima (e minima nel caso anomalo) che risultano problematici.

Soluzione

M. 1)

```
function [profilo, indiciCritici] = analizzaPressione(p_max, p_min)
% Analizza i dati di pressione e determina se il paziente ha un profilo
% normale, anomalo o critico.
%
% p_max e p_min sono array contenenti le misurazioni di pressione massima
% e minima, rispettivamente
% profilo assume valore 1 nel caso di profilo normale, 2 nel caso di
% profilo anomalo e 3 nel caso di profilo critico
% indiciCritici è array che contiene i valori degli indici corrispondenti alle celle
% in p_max i cui valori hanno determinato la definizione dello
% specifico profilo per il paziente. Nel caso di paziente con profilo
% normale, questo array sarà vuoto.

profilo = 1;
indiciCritici = [];
% profilo cambierà valore se una delle condizioni sintomo di problema
% sarà verificata.

% verifica della condizione profilo critico
filtro1 = p_max > 130;
if sum(filtro1) >= 8 % il profilo è critico
    profilo = 3;
    indiciCritici = find(p_max > 130);
else
    % verifica della condizione profilo anomalo
    p_diff = p_max - p_min;
    filtro2 = p_diff > 40;
    if sum(filtro2) >= length(p_max)/2
        profilo = 2;
        indiciCritici = find(p_diff > 40);
    end
end
end

end
```

M. 2)

```
load('holter.mat', 'p1', 'p2');
[p, ind] = analizzaPressione(p1, p2);
if (p == 1)
    colore = 'g';
elseif (p == 2)
    colore = 'b';
elseif (p == 3)
    colore = 'r';
else colore = 'k';
end
plot(1:length(p1), p1, colore);
hold on
plot(1:length(p2), p2, colore);
if (p == 2 || p == 3)
    plot(ind, p1(ind), [colore '*' '']);
end
```

```
    plot(ind, p2(ind), [colore '*' '']);  
end  
hold off
```

Esercizio 3 – Codifica binaria (6 punti)

CB.1 (2 punti) Quanti bit servono per rappresentare in complemento a due i seguenti numeri espressi in base 10? Si motivi la risposta.

A: +32767

B: +32679

C: -32768

CB.2 (2 punti) Si fornisca la rappresentazione in complemento a due per ciascuno dei numeri dati.

CB.3 (2 punti) Si indichi se è possibile o meno svolgere correttamente le seguenti operazioni in complemento a due usando lo stesso numero di bit utilizzato per rappresentare i numeri di partenza. Si motivi la risposta e si eseguano gli eventuali calcoli:

A+B

A+C

Soluzione

CB.1 Un numero N , espresso in base 10, è rappresentabile in complemento a 2 con un numero n di bit se rispetta i limiti di rappresentabilità per i numeri in complemento a due secondo la seguente formula:

$$-2^{n-1} \leq N \leq 2^{n-1}-1$$

Nel caso specifico dei tre numeri considerati, si ha che per $n = 16$ possiamo rappresentare i numeri positivi fino a $2^{15}-1 = 32767$ e i numeri negativi fino a $-2^{15} = -32768$. Di conseguenza, tutti e tre i numeri A, B e C sono rappresentabili in complemento a due con 16 bit.

CB.2

A: $+32767_{\text{dec}} \rightarrow 0111\ 1111\ 1111\ 1111_{\text{C2}}$

32767		1
16383		1
8191		1
4095		1
2047		1
1023		1
511		1
255		1
127		1
63		1
31		1
15		1
7		1
3		1
1		1

B: $+32679_{\text{dec}} \rightarrow 0111\ 1111\ 1010\ 0111_{\text{C2}}$

32679		1
16339		1
8169		1
4084		0
2042		0
1021		1
510		0
255		1
127		1
63		1
31		1
15		1
7		1
3		1
1		1

C: $-32768_{\text{dec}} \rightarrow 1000\ 0000\ 0000\ 0000_{\text{C2}}$

32768		0
16384		0
8192		0
4096		0
2048		0
1024		0
512		0
256		0
128		0

64		0
32		0
16		0
8		0
4		0
2		0
1		1

32768_{dec} -> 1000 0000 0000 0000_{bin}

Invertiamo i bit

0111 1111 1111 1111

Sommiamo 1

1111	1111	1111	111	(riporti)
0111	1111	1111	1111	
			1	

1000 0000 0000 0000_{C2}

Sia per A che per C potremmo fare a meno di sviluppare i calcoli completi e determinare direttamente il valore in complemento a due considerando che questi due numeri corrispondono all'estremo superiore e inferiore dell'intervallo di rappresentabilità.

CB.3

A+B: non è possibile eseguire questo calcolo ottenendo un risultato in 16 bit perché il risultato sarebbe superiore all'estremo dell'intervallo di rappresentabilità per i valori positivi, infatti $+32767+32769 = 65446$. Il calcolo può essere eseguito dopo aver rappresentato entrambi i numeri con un numero di bit maggiore di 16.

A+C: corrisponde a $+32767-32768 = -1$ che è rappresentabile in complemento a due in 16 bit:

0111	1111	1111	1111	+
1000	0000	0000	0000	=
<hr/>				
1111	1111	1111	1111	

Domanda 1 (3 punti)

Si individui l'errore presente nel seguente programma e lo si corregga.

```
#include <stdio.h>

typedef struct {
    int a;
    float b;
} dato;

int main()
{
    scanf("%d", &(dato.a));
    scanf("%f", &(dato.b));
    printf("%d ", dato.a);
    printf("%f\n", dato.b);
}
```

Domanda 2 (3 punti)

Si analizzi la seguente descrizione:

Il sistema operativo determina il passaggio del processo P dallo stato di esecuzione allo stato di attesa nel momento in cui la periferica segnala di essere pronta a fornire il dato richiesto da P.

È una descrizione corretta del comportamento del sistema operativo? Si motivi la risposta.