

	Politecnico di Milano Facoltà di Ingegneria Industriale <b>INFORMATICA B</b> Appello 18 Febbraio 2016	COGNOME E NOME				
			MATRICOLA			
Spazio riservato ai docenti <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table>						

- Il presente plico contiene 3 esercizi e **deve essere debitamente compilato con cognome e nome, 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!).
- **È vietato utilizzare calcolatrici, telefoni o pc.** Chi tenti di farlo vedrà annullata la sua prova.
- È ammessa la consultazione di libri e appunti, purché con pacata discrezione e senza disturbare.
- **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.**
- L'esame orale è parte integrante dell'esame e deve essere realizzato almeno sufficientemente per il superamento dell'esame complessivo.

### Esercizio 1 (6 punti)

Un programma deve controllare se un numero **N** inserito dall'utente sia contemporaneamente:

- primo,
- dispari,
- compreso tra 3 e 100, estremi inclusi.

Quando il numero **N** inserito non soddisfa tutte le condizioni citate, l'inserimento deve essere ripetuto.

Per realizzare tale programma, un programmatore ha scritto il seguente frammento di codice Matlab, sfruttando la funzione **primo** che restituisce true se il suo argomento è un numero primo, false altrimenti:

```
N = input('Inserire un numero: ');  
while(primo(N) && (3 <= N) && (N <= 100) && (mod(N,2)))  
    N = input('Inserire un numero: ');  
end
```

Si risponda alle seguenti domande:

A. Riportare la tabella di verità della condizione di terminazione del while

A	B	C	D	(B && C)	(B && C && D)	(A && B && C && D)
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

dove A: primo(N), B: (3 <= N), C: (N <= 100) e D: (mod(N,2))

- B. La condizione di terminazione del ciclo while può essere semplificata per evitare delle ridondanze?
- C. Si dica se il frammento di codice soddisfa i requisiti del programma. Nel caso in cui si ritenga che il frammento di codice non sia corretto, se ne indichi la motivazione e se ne fornisca un controesempio (ovvero un valore di N che ne mostri la non correttezza) spiegandolo.
- D. Nel caso in cui si ritenga che il frammento di codice non sia corretto, se ne fornisca una versione in grado di soddisfare i requisiti indicati.

## SOLUZIONE

A.

A	B	C	D	(B && C)	(B && C && D)	(A && B && C && D)
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	1	0	0
0	1	1	1	1	1	0
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	0	0	0
1	1	0	0	0	0	0
1	1	0	1	0	0	0
1	1	1	0	1	0	0
1	1	1	1	1	1	1

- B. La condizione  $\text{mod}(n,2)$  è superflua, perché un numero primo maggiore od uguale a 3 è necessariamente dispari
- C. Non soddisfa i requisiti, infatti inserendo  $N=5$ , il programma continua a ripetere il ciclo invece di riconoscere l'input valido rispetto alle condizioni richieste. Occorre perciò modificare l'espressione, negando quella fornita.

D.

```
N = input('Inserire un numero: ');
while (~ ( primo(N) && (3 <= N) && (N <= 100) ) )
    N = input('Inserire un numero: ');
end
```

## Esercizio 2 (10 punti)

Si scriva un programma in C che permetta l'analisi delle foto delle montagne innevate caricate sul social network "LibroFaccia". Si considerino date le seguenti definizioni:

```
typedef char Stringa[30];

typedef struct
{
    Stringa nome;
    Stringa cognome;
    Stringa username;
    int annoIscrizione;
} Utente;
```

che rappresentano, rispettivamente, una stringa all'interno di LibroFaccia e un utente del social.

Si svolgano i seguenti punti:

- A. Si definisca un tipo di dato `Foto` atto a rappresentare una foto di un utente; Ogni foto è caratterizzata da un'immagine (matrice di `MAX_DIM` x `MAX_DIM` di interi), da un titolo e da un tag (entrambi rappresentabili con una stringa di 30 caratteri).
- B. Si definisca la costante `MAX_DIM` pari a 512. Come indicato al punto precedente, questa rappresenta il numero massimo di righe e colonne di un'immagine.
- C. Si modifichi la definizione di `Utente` affinché sia in grado di
  - contenere fino a 50 `Foto` per ogni utente (un utente potrebbe avere meno di 50 foto ed in numero diverso dagli altri utenti);
  - riportare quante foto siano effettivamente contenute per ogni utente;
- D. Si definisca una variabile `libroFaccia` contenente `MAX_UTENTI` elementi di tipo `Utente`. Si definisca inoltre la costante `MAX_UTENTI` con valore pari a 1000.

Si assuma che la variabile `libroFaccia` sia stata precedentemente popolata interamente e che ogni utente sia stato popolato in tutti i suoi campi (comprese le proprie foto). Si scriva una porzione di codice in linguaggio C che:

- E. Richieda all'utente di inserire un valore soglia `T`, controllando che sia compreso tra 0 e 255 (estremi inclusi).
- F. Tra tutte le foto di tutti degli utenti di `LibroFaccia`, stampi a schermo il titolo della foto la cui matrice immagine ha il maggior numero di elementi che sono maggiori o uguali alla soglia `T` e che ha come tag "Montagna" o "Neve". (Nel caso esistano più foto con tali caratteristiche, è sufficiente considerarne una qualsiasi di esse). Si stampi inoltre a schermo il nome e il cognome dell'utente che possiede tale foto.

## Soluzione

```
#define MAX_DIM 512
#define MAX_UTENTI 1000

typedef struct{
    int immagine[MAX_DIM][MAX_DIM];
    Stringa titolo;
    Stringa tag;
} Foto;

typedef char Stringa[30];
typedef struct{
    Stringa nome;
    Stringa cognome;
    Stringa username;
    int annoIscrizione;
    Foto fotoUtente[50];
    int nFoto;
} Utente;
```

```

Utente libroFaccia[MAX_UTENTI];

int T;
do{
    printf("Inserire una soglia tra 0 e 255");
    scanf("%d",&T);
}while(T < 0 || T > 255)

int i, j, k, h, num_utente, num_foto;
int sum_soglia = 0;
int max_sum_soglia = 0;
Stringa foto_trovata;
Stringa nome_trovato;
Stringa cognome_trovato;
for (i = 0; i < MAX_UTENTI; i++)
    for (j = 0; j < libroFaccia[i].nFoto; j++)
        if( strcmp(libroFaccia[i].fotoUtente[j].tag,"Montagna") == 0
            || strcmp(libroFaccia[i].fotoUtente[j].tag,"Neve") == 0) {
            sum_soglia = 0;
            for (k = 0; k < MAX_DIM; k++)
                for (h = 0; h < MAX_DIM; h++)
                    if (libroFaccia[i].fotoUtente[j].immagine[k][h] > soglia)
                        sum_soglia++;
            if (sum_soglia > max_sum_soglia) {
                max_sum_soglia = sum_soglia;
                num_utente = j;
                num_foto = i;
            }
        }

if (max_sum_soglia > 0) {
    strcpy(foto_trovata,libroFaccia[num_utente].fotoUtente[num_foto].titolo);
    strcpy(nome_trovato,libroFaccia[num_utente].nome);
    strcpy(cognome_trovato,libroFaccia[num_utente].cognome);

    printf("Foto trovata: %s\n",foto_trovata);
    printf("Scattata da: %s %s\n",nome_trovato,cognome_trovato);
}
else
    printf("Nessuna foto trovata");

```

### Esercizio 3 (10 punti)

Si vuole visualizzare su un display contenente **n\_char** caratteri una stringa **text**. Si assuma che **text** abbia più di **n\_char** caratteri e che quindi il testo venga fatto scorrere verso sinistra, come nei display che indicano la prossima fermata all'interno dei treni. Per esempio, dato il testo 'buon viaggio' e un display di 10 caratteri, viene visualizzata la sequenza di stringhe riportate qui sotto:

```
buon viagg
uon viaggi
on viaggio
n viaggio-
viaggio--
viaggio---
iaggio----
aggio-----
ggio-----
gio-----
io-----
o-----
-----
```

Si risponda alle seguenti domande:

- A. Si implementi in linguaggio Matlab un funzione **showRollingTxt** per visualizzare una stringa **text** su un display contenente **n\_char** caratteri, come descritto sopra.

In particolare la funzione **showRollingTxt** opera come segue:

- Inizialmente vengono visualizzati i primi **n\_char** caratteri di **text**; Se la parte di testo da visualizzare in **text** è più corta di **n\_char** caratteri, vengono introdotti caratteri '-' (trattino) fino a riempire il display.
- Il testo quindi *scorre* verso sinistra di un carattere, visualizzando **n\_char** caratteri a partire dal secondo carattere; come nel caso precedente, se la parte di testo da visualizzare è più corta di **n\_char** caratteri, si introducono caratteri '-'.
- La procedura descritta sopra continua fino a quando il display contiene **n\_char** caratteri uguali a '-'.
- Ogni volta che il testo scorre di un carattere, la funzione **showRollingTxt** stampa su una riga diversa il testo visualizzato nel display.

- B. Si scriva un script Matlab che invochi la funzione **showRollingTxt** per mostrare il testo "buon viaggio" su un display avente 10 caratteri.

- C. Si implementi la funzione ricorsiva **showRollingTxtRic** che produce lo stesso risultato di **showRollingTxt** prendendo gli stessi parametri in ingresso ed uscita.

### Soluzione

A)

```
function showRollingTxt(text, n_char)
```

```
lenTXT = length(text);
```

```
txtToShow(1 : lenTXT + n_char) = '-';
txtToShow(1: lenTXT) = text;
```

```
for ii = 1 : lenTXT + 1
    disp(txtToShow(1 : n_char));
    txtToShow(end + 1) = txtToShow(1);
    txtToShow(1) = [];
end
```

B)

```
t = 'buon viaggio';
n = 10;
```

```
showRollingTxt(t, n);
```

**c)**

```
%Versione ricorsiva
function [] = showRollingTxtRic(text, n_char)

lenTXT = length(text);
txtToShow(1 : lenTXT + n_char) = '-';
if lenTXT == 0
    disp(txtToShow);
else
    txtToShow(1: lenTXT) = text;
    disp(txtToShow(1 : n_char));
    showRollingTxtRic(text(2:end), n_char);
```