

	Politecnico di Milano Facoltà di Ingegneria Industriale <b>INFORMATICA B</b> Prova in itinere del 10 Novembre 2009		COGNOME E NOME
	RIGA	COLONNA	MATRICOLA
			Spazio riservato ai docenti <div style="text-align: right;"> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> </div>

- Il presente plico contiene 4 esercizi, deve essere debitamente compilato con cognome e nome, numero di matricola, posizione durante lo scritto (comunicata dal docente).
- Il tempo a disposizione è di 2 ore.
- 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.

### Esercizio 1 (4 punti)

Si consideri la seguente espressione booleana:

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

Si costruisca per essa la **tabella di verità** e si risponda alle seguenti domande:

- la formula è falsa per qualsiasi valore delle variabili A, B e C ?
- la formula è vera per qualsiasi valore delle variabili A, B e C ?

Si consideri ora la condizione, scritta in linguaggio C, in cui  $x$  e  $y$  siano due variabili `int`:

$!(x > 4 \ || \ x < 9) \ \&\& \ (x > 4 \ || \ !(x == y))$

ottenuta dalla prima formula sostituendo la variabile A con  $x > 4$ , B con  $x < 9$  e C con  $x == y$ .

Si risponda alle seguenti domande:

- la condizione è vera o falsa quando  $x$  vale 1 e  $y$  vale 2
- la condizione è vera per qualsiasi valore di  $x$  e  $y$
- la condizione è falsa per qualsiasi valore di  $x$  e  $y$

Le risposte ai quesiti devono essere sinteticamente giustificate.

### Soluzione

A	B	C	!	(A	OR	B)	AND	(A	OR	NOT	C)
F	F	F	V	F	F	F	V	F	V	V	F
F	F	V	V	F	F	F	F	F	F	F	V
F	V	F	F	F	V	V	F	F	V	V	F
F	V	V	F	F	V	V	F	F	F	F	V
V	F	F	F	V	V	F	F	V	V	V	F
V	F	V	F	V	V	F	F	V	V	F	V
V	V	F	F	V	V	V	F	V	V	V	F
V	V	V	F	V	V	V	F	V	V	F	V

**La formula non è sempre vera né sempre falsa.**

**L'espressione** scritta in linguaggio C **è sempre falsa perchè**  $!(x > 4 \ || \ x < 9)$  è equivalente a  $(x \leq 4 \ \&\& \ x \geq 9)$ , e  $x$  non può essere simultaneamente minore di 4 e maggiore di 9. Quindi l'espressione non è sempre vera ed è falsa quando  $x$  vale 1 e  $y$  vale 2.

## Esercizio 2 (3 punti)

- a) Si dica qual è l'intervallo di valori interi rappresentabile con la codifica in complemento a due a 9 bit.
- b) Con riferimento a tale codifica indicare, giustificando brevemente le risposte, quali delle seguenti operazioni possono essere effettuate correttamente:
- (i)  $-254 - 255$
  - (ii)  $+254 - 253$
  - (iii)  $-18 + 236$
  - (iv)  $+217 + 182$
- c) Mostrare in dettaglio come avviene il calcolo delle operazioni (i) e (ii), evidenziando il bit di riporto e il bit di overflow così ottenuti.

**Note:** Il bit di overflow è pari ad 1 se si verifica overflow, 0 altrimenti.

## Soluzione

a) I valori rappresentabili vanno da  $-256$  a  $+255$ .

b)

- (i)  $-254 - 255$  no, si ottiene un valore negativo troppo grande in valore assoluto
- (ii)  $+254 - 253$  si, si ottiene un valore piccolo in valore assoluto
- (iii)  $-18 + 236$  si, si ottiene un valore positivo, grande in valore assoluto ma nei limiti
- (iv)  $+217 + 182$  no, si ottiene un valore positivo troppo grande in valore assoluto

c)

$\begin{array}{r} 100000010 \\ \underline{100000001} \\ [1] (1) 000000011 \end{array}$	$\begin{array}{r} 011111110 \\ \underline{100000011} \\ [0] (1) 000000001 \end{array}$
--	--

### Esercizio 3 (4 punti)

Si consideri il seguente frammento di codice C:

```
int s, i, j;
s = 0;
for (i=1; i<=10; i++) {
    j = i * 2;
    while ( j>0 ) {
        s = s + 1;
        j = j - 1;
    }
    if ( s%2 == 0 )
        printf("%d", s);
}
```

Attraverso una sintetica simulazione dell'esecuzione di tale frammento, indicare quali valori vengono stampati.

### Soluzione

Vengono stampati i valori  $n^2+n$ , per  $1 \leq n \leq 10$  : 2, 6, 12, 20, 30, 42, 56, 72, 90, 110.

#### Esercizio 4 (6 punti)

Si considerino le seguenti dichiarazioni di tipi e variabili, che definiscono le strutture dati per rappresentare un semplice orario ferroviario:

```
typedef struct {
    unsigned ora; /* tra 0 e 23*/
    unsigned minuto; /* tra 0 e 59 */
} tempo;
typedef char stazione[30]; /* il nome di una stazione */
typedef struct {
    int numero; /* numero del treno */
    tempo tPartenza, tArrivo;
    stazione da, a; /* stazioni di partenza e di arrivo */
} treno;
typedef struct{
    treno t[200];
    int nTreni; /* il numero dei treni effettivamente presenti nell'orario,
                che occupano le prime nTreni posizioni nell'array*/
} orario;
orario o;    tempo tPart, tDest;    stazione part, dest, scambio;
```

- Scrivere un frammento di codice, dichiarando eventuali variabili aggiuntive, che trovi tutti i treni che partono, a un'ora successiva a quella memorizzata nella variabile `tPart`, dalla stazione `part`, e arrivano alla stazione `dest`. Per ognuno di tali treni si visualizzi a schermo il numero del treno, la stazione di arrivo e l'orario di arrivo (ora e minuti).
- Trovare tutte le coppie di treni in cui il primo dei due treni parte dalla stazione `part` e arriva alla stazione `scambio`, e il secondo dei due parte da `scambio` e arriva alla stazione `dest`. Si visualizzi, per ogni coppia trovata, il numero del primo e del secondo treno.

**Note:** Si assuma che le variabili `o`, `tPart`, `tDest`, `part`, `dest`, `scambio`, siano state inizializzate attraverso istruzioni non mostrate. Per confrontare le stringhe si consiglia di utilizzare la funzione `strcmp(x, y)`, che riceve in ingresso due stringhe (`x` e `y`), e restituisce 0 se e solo se sono uguali.

#### Soluzione

```
int i;
for ( i=0; i<o.nTreni; i++)
    if ( strcmp(o.t[i].da, part)==0 &&
        strcmp(o.t[i].a, dest)==0 &&
        o.t[i].tPartenza.ora >= tPart.ora &&
        (o.t[i].tPartenza.ora > tPart.ora || o.t[i].tPartenza.minuto >= tPart.minuto) )
        printf ("il treno %d arriva a %s alle %d:%d \n", o.t[i].numero, o.t[i].a,
                o.t[i].tArrivo.ora, o.t[i].tArrivo.minuto);
```

```
int i, j;
for ( i=0; i<o.nTreni; i++)
    for ( j=0; j<o.nTreni; j++)
        if (strcmp(o.t[i].da, part)==0 && strcmp(o.t[i].a, int)==0 &&
            strcmp(o.t[j].da, int)==0 && strcmp(o.t[j].a, dest)==0 )
            printf ("%d %d \n", o.t[i].numero, o.t[j].numero);
```