Fondamenti di Informatica A

Compito 2 – 19/6/2019

Cognome e Nome _____ matr. _____

Domanda 1. Si consideri la macchina di Turing con alfabeto {*blank*, 0, 1}, stato iniziale q0, e tabella delle istruzioni seguente:

Stato corrente	Simbolo corrente	Nuovo simbolo	Nuovo stato	Spostamento
q0	1	1	q0	right
q0	0	1	halt	---
q0	*blank*	1	halt	---

Allora:

[*V*] [*F*] Se il nastro contiene inizialmente la sequenza 000111, e la testina di lettura-scrittura è posizionata sulla prima cifra a sinistra (quella sottolineata), allora al termine dell'esecuzione il nastro conterrà 100111

[*V*] [*F*] Se la MdT viene fatta partire con il nastro vuoto (cioè in cui tutte le caselle sono *blank*), allora al termine dell'esecuzione il nastro sarà ancora vuoto

[*V*] [*F*] Se il nastro contiene inizialmente la sequenza 111, e la testina di lettura-scrittura è posizionata sulla prima cifra a sinistra (quella sottolineata), allora al termine dell'esecuzione il nastro conterrà la sequenza 110

[*V*] [*F*] Se il nastro contiene inizialmente la sequenza 010, e la testina di lettura-scrittura è posizionata sulla cifra a destra (quella sottolineata), allora al termine dell'esecuzione il nastro conterrà la sequenza 011

Risposte:

- V
- F: il nastro conterrà 1
- F
- V

Domanda 2. Si consideri l'espressione booleana *R* = (*A* **XOR** *B*) **OR** (*A* **AND** *B*). Allora:

[*V*] [*F*] Se *A* e *B* hanno valore diverso (uno dei due è *true* e l'altro *false*) allora il risultato è sempre *true*

[*V*] [*F*] Se *A* e *B* hanno lo stesso valore (entrambi *true* o entrambi *false*) allora il risultato è sempre *true*

[*V*] [*F*] Se *A* e *B* sono entrambi *true*, il risultato è *true*

[*V*] [*F*] Il risultato dell'espressione è sempre *true*, qualunque siano i valori di *A* e *B*

Risposte:

- V
- F: se *A* = *B* = *false*, il risultato è *false*
- V
- F: se *A* = *B* = *false*, il risultato è *false*

Domanda 3. Il seguente frammento di codice in linguaggio C compila ed esegue senza errori:

```
int a[] = {54, 50, 12, -77, 12};
int *p = &a[0];
while ( *p > 0 ) {
    *p = *p - 1;
    p = p + 1;
}
```

Al termine dell'esecuzione:

- [V] [F] `a[0]` ha valore 54
- [V] [F] `a[1]` ha valore 50
- [V] [F] `a[2]` ha valore 12
- [V] [F] `a[3]` ha valore -77

Risposte:
- F: vale 53
- F: vale 49
- F: vale 11
- V

Domanda 4. Si consideri la seguente funzione ricorsiva in linguaggio C:

```
int f(int n) {
      if (n > 5) {
            return n;
      } else {
            return f(n+2);
      }
}
```

- [V] [F] L'espressione `f(0)` ha valore 0
- [V] [F] L'espressione `f(5)` ha valore 5
- [V] [F] L'espressione `f(4)` ha valore 6
- [V] [F] L'espressione `f(-101)` ha valore -1

Risposte:
- F: ha valore 6
- F: ha valore 7
- V
- F: la funzione restituisce sempre un valore strettamente maggiore di 5

Domanda 5. Si considerino due numeri $A = 0100\ 0011_{2C}$ e $B = 0010\ 0010_{2C}$ rappresentati in complemento a due con $N = 8$ bit. Allora:

- [V] [F] *A* rappresenta un valore positivo
- [V] [F] *B* rappresenta un valore positivo
- [V] [F] *B* rappresenta un valore pari
- [V] [F] Il calcolo di (*A* + *B*) (usando 8 bit) non genera overflow

Risposte:
- V
- V
- V
- V: sono due valori positivi ma la loro somma non genera overflow usando 8 bit ($A = 67$, $B = 34$, $A + B = 101$)

Domanda 6. Sapendo che inizialmente vale l'asserzione $x \geq 0$, scrivere negli appositi spazi le asserzioni più specifiche che valgano dopo ciascuna delle istruzioni del seguente frammento di codice, assumendo che tutte le variabili siano di tipo int e che le istruzioni vengano eseguite una di seguito all'altra.

{ $x \geq 0$ }
```
x = x + 1;
```
{ _____ }
```
y = 2 * x;
```
{ _____ }
```
y = y - 1;
```
{ _____ }

Risposta: Una possibilità (non è l'unica) sono le asserzioni seguenti:

{ $x \geq 0$ }
```
x = x + 1;
```
{ $x > 0$ }
```
y = 2 * x;
```
{ $x > 0$ **and** $y = 2x$ }
```
y = y - 1;
```
{ $x > 0$ **and** $y = 2x - 1$ }