

Informatica Generale - Primo Appello del 10 giugno 2009

ATTENZIONE: Scrivere nome e cognome su tutti i fogli utilizzati

[Esercizio 1]

(1.a) Scrivere la tavola di verità della seguente funzione booleana in tre variabili:

$$K(A, B, C) = (A \text{ xor } B) \text{ and } (C \text{ or not } B)$$

(1.b) Sfruttando la tavola di verità, disegnare un circuito combinatorio che calcola la funzione K usando solo porte AND, OR e NOT. Se si desidera, si possono usare porte AND e OR con più di due ingressi.

[Esercizio 2]

(2.a) Si mostri la rappresentazione in complemento a 2 su otto bit dei numeri -54 e 73.

(2.b) Qual è la distanza di Hamming delle due rappresentazioni?

(2.c) Qual è il più grande numero positivo rappresentabile in complemento a 2 con otto bit? E il più piccolo numero negativo? Qual è la loro rappresentazione?

(2.d) Si calcoli lo XOR delle due rappresentazioni trovate al punto (2.a): mostrare sia la configurazione di bit risultante, che il suo valore in base 10.

[Esercizio 3]

(3.a) La tabella seguente mostra una porzione della memoria del computer contenente un programma in linguaggio macchina. Si assuma che l'esecuzione cominci con il valore **00** nel registro PC. Scrivere la sequenza di istruzioni che vengono eseguite finché la macchina non si fermi, mostrando il contenuto del PC e dei registri generali dopo ogni istruzione. Rispondere poi alle domande che seguono.

indirizzo	contenuto
00	20
01	02
02	21
03	01
04	22
05	01
06	52
07	12
08	B2
09	0C
0A	B0
0B	06
0C	C0
0D	00

- (3.b) Quale sequenza di bit contiene il registro 2 quando la macchina si ferma?
 (3.c) Quante volte viene eseguita l'istruzione di indirizzo 06 prima che la macchina si fermi?
 (3.d) Se la cella di indirizzo 01 contenesse 04 invece di 02, cosa cambierebbe nell'esecuzione del programma? Quale sarebbero le risposte alle domande (3.b) e (3.c) in questo caso?

[Esercizio 4] Quale sequenza di valori viene stampata se si eseguono le seguenti istruzioni? Motivare la risposta.

```
int X ← 2;
int Y ← 44;
while (X < Y) do
  ( print(X);
    X ← X + X;
  )
print(X-Y);
```

[Esercizio 5] La seguente procedura è progettata per restituire il più grande intero il cui quadrato non è maggiore del parametro **N**, dove si assume che **N** sia maggiore di zero. Per esempio, **RadiceQuadrata(5)** dovrebbe restituire **2**, mentre **RadiceQuadrata(9)** dovrebbe restituire **3**.

La procedura è corretta? In caso contrario, come si può correggere l'errore?

```
int procedure RadiceQuadrata(int N)
  if (N <= 0) then ( print("Errore: N è <=0"); return 0; )
  else (
    int X ← 0;
    while (X * X <= N) do
      (X ← X + 1; )
    return X;
  )
```

Attenzione: Per l'esercizio (6.b) che segue, si chiede di scrivere una procedura usando lo pseudocodice visto a lezione (usando *assegnamenti*, *if-then-else*, *while-do* e *procedure*). Si richiede che ogni variabile utilizzata nella procedura venga prima introdotta in una dichiarazione che ne fissi il tipo e, eventualmente, un valore iniziale. Le sequenze di dati devono essere rappresentate con array. Anche i parametri delle procedure devono essere preceduti dal tipo corrispondente, e la parola chiave **procedure** deve essere preceduta dal tipo del risultato, se si attende un risultato.

[Esercizio 6]

(6.a) Descrivere in modo informale, ma più preciso possibile, un algoritmo che risolva il seguente problema: Data una sequenza di numeri interi $a[1], a[2], \dots, a[n]$, determinare se la sequenza contiene più numeri pari o più numeri dispari.

(6.b) Scrivere una procedura **Paridispari** che realizza l'algoritmo del punto precedente. La procedura **Paridispari** deve avere come parametri un array di interi e la sua lunghezza, e deve restituire **TRUE** se nell'array ci sono più numeri pari che numeri dispari, **FALSE** altrimenti.