

Prima prova in itinere di Calcolatori Elettronici – 09.11.2007

Corsi di Laurea in Ing. Gestionale e Ing. delle Telecomunicazioni

A.A. 2007-2008

Cognome		Nome		Matricola							
---------	--	------	--	-----------	--	--	--	--	--	--	--

Istruzioni: Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova.

Esercizio 1 – Rappresentazione dell'informazione (7 punti)

Si considerino i seguenti due numeri in base 10, $X_{10} = 25$ e $Y_{10} = -6$; $Z_{10} = 11,33$ calcolare:

- 1) le rappresentazioni in base 2 di X_{10} (sia X_2), Z_{10} (sia Z_2), e in complemento a 2 di Y_{10} (sia Y_2) (1pt);
- 2) la somma $S_2 = X_2 + Z_2$ dei due numeri binari ottenuti al punto 1) (1pt);
- 3) la moltiplicazione $P_2 = X_2 \times |Y_2|$ (1pt);
- 4) la rappresentazione in base 10 del numero binario ottenuto giustapponendo le cifre di X_2 e P_2 (1pt).
- 5) la rappresentazione secondo lo standard IEEE/ANSI 754 dei numeri *floating point* su 4 byte di Z_{10} (2pt);
- 6) la rappresentazione esadecimale del numero calcolato al punto precedente (1pt).

Esercizio 2 – Reti combinatorie (7 punti)

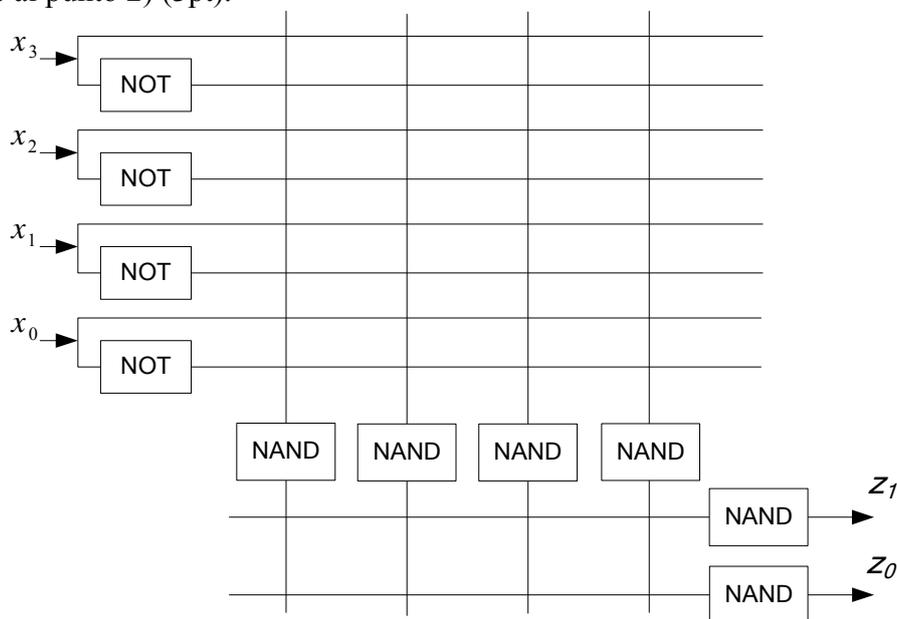
La rete R formata da 4 ingressi x_3, x_2, x_1, x_0 due uscite z_1, z_0 ha un comportamento definito dalla seguente tabella di verità:

$$z_1 = \sum_4(4, 5, 6, 7, 10, 11, 14, 15)$$

$$z_0 = \sum_4(6, 7, 8, 9, 10, 11, 14, 15)$$

Si richiede quanto segue:

- 1) calcolare la forma di tipo SP di costo minimo secondo il criterio di costo a porte per ogni uscita z (2pt);
- 2) per ogni espressione calcolata al punto precedente, individuare eventuali alee e modificare le espressioni delle uscite in modo da eliminare tali alee (2pt);
- 3) programmare la seguente PLA in modo da implementare le uscite z_1, z_0 secondo le espressioni prive di alee calcolate al punto 2) (3pt).



Esercizio 3 – Memorie (5 punti)

Avendo a disposizione dei chip di memoria RAM del tipo 32 MByte x 8 bit, sintetizzare una memoria di tipo 128 MByte x 16 bit, accessibile al byte e alla parola. Disegnare il montaggio, dettagliando la logica combinatoria eventualmente utilizzata e commentando la soluzione scelta (1pt espansione dati, 2pt espansione indirizzi, 2 pt accesso al byte e alla parola).

Esercizio 4 – Reti sequenziali asincrone (7 punti)

Si consideri una *rete sequenziale asincrona* dotata di due ingressi x_1 e x_0 e due uscite z_1 e z_0 , il cui comportamento è specificato come segue:

- se $x_1 = x_0 = 0 \rightarrow z_1 = z_0 = 0$;
- se $x_1 \neq x_0 \rightarrow z_1 = x_0$ e $z_0 = x_1$;
- se $x_1 = x_0 = 1 \rightarrow z_1 = z_0 = 1$.

Per tale rete, pilotata in modo *fondamentale e senza transizione multiple in ingresso*, si richiede di:

- 1) disegnare il diagramma di flusso (2pt);
- 2) indicare il modello strutturale a cui fare riferimento per la sintesi (1pt);
- 3) eseguire la sintesi algebrica e circuitale (3pt);
- 4) discutere il comportamento della rete in caso di transizioni multiple in ingresso (1pt).

Esercizio 5 – Reti sequenziali sincronizzate (7 punti)

Si consideri una rete sequenziale sincronizzata di Moore con due variabili di ingresso x_1 e x_0 e due variabili di uscita z_1 e z_0 . Interpretando le due variabili di uscita come un numero naturale a due cifre in base due, il comportamento della rete è il seguente:

- quando $x_1 = 1$, se $x_0 = 0$ viene conservato il valore dell'uscita, se $x_0 = 1$ la rete incrementa (modulo 2) il numero naturale rappresentato sulle uscite;
- quando $x_1 = 0$ la rete azzerà il valore delle uscite indipendentemente dal valore di x_0 .

Si chiede di:

- 1) disegnare il diagramma di flusso (2pt);
- 2) disegnare il modello strutturale in cui la sottorete sequenziale sia composta da flip-flop J-K (1pt);
- 3) riferendosi al modello strutturale considerato al punto precedente, eseguire la sintesi algebrica e circuitale (4pt).

Prima prova in itinere di Calcolatori Elettronici – 04.12.2007

Corsi di Laurea in Ing. Gestionale e Ing. delle Telecomunicazioni

A.A. 2007-2008

Cognome		Nome		Matricola											
---------	--	------	--	-----------	--	--	--	--	--	--	--	--	--	--	--

Istruzioni: Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova, ad eccezione dell'elenco di istruzioni e degli interrupt del processore 8086.

Esercizio 1 – Traduzione da linguaggio ad alto livello ad Assembler (10 punti)

Dato il seguente listato scritto in linguaggio C in linguaggio **Assembler 8086**

1. *Disegnare* lo stato dello stack per ogni chiamata di funzione *dopo l'allocazione delle variabili locali*, e
2. **Tradurre** il listato *commentando* opportunamente il codice scritto.

```
int g = 0;

int average(int a, int b) {
    int m;

    m = (a + b)/2;

    return m;
}

int fun(int k) {
    int p;

    p = k * 2 - g;

    return p;
}
```

```
int main()
{
    int h, i, s;

    s = 0;
    h = 5;
    for (i = 0; i < 20; i++) {
        s = s + g;
        if (i > 10)
            h = average(s, h);
        else
            h++;
        g = fun(h);
    }

    return g;
}
```

Esercizio 2 – Programmazione Assembler (15 punti)

Utilizzando il linguaggio Assembler 8086 e l'insieme di *interrupt* disponibili sui processori compatibili x86, si realizzi il programma che faccia quanto segue:

1. chiede all'utente di digitare da tastiera un numero compreso tra 01 e 20 (due cifre codificate ASCII) (2pt);
2. memorizza tale numero all'interno di una variabile avente nome simbolico N, ampia 1 byte (1pt);
3. dati due vettori VETT1 e VETT2 di 20 elementi ciascuno, considera i primi N elementi di ciascuno, e per ognuno di essi calcola e stampa a video ogni volta su una linea diversa:
 - a. l'indice dell'elemento considerato seguito dal carattere ':' (1pt);
 - b. quanti bit sono a 1 nell'elemento considerato di VETT1 (1pt),
 - c. quanti bit sono a 0 nell'elemento considerato di VETT2 (1pt),
 - d. quale è il valore numerico decimale codificato ASCII dell'elemento considerato di VETT1 interpretato come numero binario naturale (2pt),
 - e. quale è il valore numerico decimale codificato ASCII dell'elemento considerato di VETT2 interpretato come numero binario naturale (2pt),
 - f. quale è la media intera troncata tra i gli elementi corrispondenti di VETT1 e VETT2 interpretati come numeri binari naturali (1pt);
4. scambia il contenuto dell'elemento *i-esimo* di VETT1 con quello *N-i esimo* di VETT2 (2pt);
5. ripete il punto 3 per i vettori così trasformati (1pt).

Esempio di output.

```
Inserire numero: 02
1: 3 7 5 1 3
2: 2 5 3 10 6
1: 3 7 10 3 6
2: 1 5 1 5 3
```

Esercizio 3 – Questionario Architettura x86 (8 punti)

Rispondere alle seguenti domande in modo conciso ed esauriente.

1. Cosa è l'operazione di fetch ?
2. Descrivere anche tramite esempi assembler il modello architetturale a stack.
3. Descrivere il flag di segno (SF) del registro di stato e fornire un esempio di utilizzo.
4. Descrivere il meccanismo di calcolo dell'indirizzo effettivo del processore i8086.
5. Descrivere e fornire un esempio di indirizzamento relativo per il processore i8086.
6. Descrivere quali sono le informazioni contenute nel record di attivazione (programmazione a moduli).
7. Descrivere lo schema funzionale di una interfaccia con registri di stato evidenziando anche la connessione al bus (schema e commento).
8. Descrivere il ciclo di lettura del bus asincrono (diagramma temporale e commento).

Recupero Prima prova in itinere di Calcolatori Elettronici – 08.01.2008

Corsi di Laurea in Ing. Gestionale e Ing. delle Telecomunicazioni

A.A. 2007-2008

Cognome		Nome		Matricola							
---------	--	------	--	-----------	--	--	--	--	--	--	--

Istruzioni: Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova.

Esercizio 1 – Rappresentazione dell'informazione (7 punti)

Si considerino i seguenti numeri in base 10:

$$X_{10} = 11$$

$$Y_{10} = -7,$$

mostrando tutti i passaggi necessari, calcolare:

- 1) la rappresentazione in base 2 su 8 bit di X_{10} (sia X_2), e in complemento a 2 su 8 bit di Y_{10} (sia Y_2) (1pt);
- 2) la somma $S_2 = X_2 + Y_2$ dei due numeri binari ottenuti al punto 1) (1pt);
- 3) la divisione $D_2 = X_2 / S_2$ ed il resto R_2 (1pt);
- 4) la rappresentazione in base 10 di D_2 (1pt);
- 5) la rappresentazione secondo lo standard IEEE/ANSI 754 dei numeri *floating point* su 4 byte di X_2 / S_2 (2pt);
- 6) la rappresentazione esadecimale del numero calcolato al punto precedente (1pt).

Esercizio 2 – Reti combinatorie (7 punti)

Data la seguente mappa:

	x_3x_2				
x_1x_0		00	01	11	10
00		1	0	0	-
01		-	1	0	-
11		-	1	-	0
10		-	1	0	1
		Z			

- 1) indicare e classificare tutti gli implicanti principali (1pt);
- 2) trovare tutte le possibili liste di copertura cui corrispondono forme di tipo SP di costo minimo secondo il criterio di costo a porte (2pt);
- 3) per ognuna delle liste di copertura trovate nel punto 2), individuare e classificare le eventuali alee del primo ordine presenti, e modificare la corrispondente lista in modo da eliminare le alee (2pt);
- 4) effettuare una nuova sintesi della mappa a costo minimo e priva di alee utilizzando esclusivamente porte NAND (2pt).

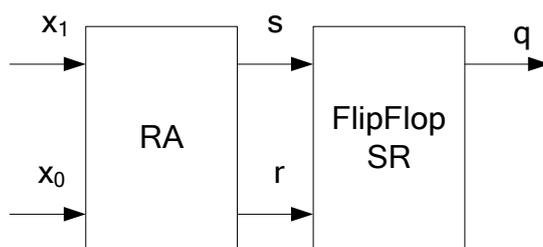
NOTA. Specificare le espressioni utilizzando esclusivamente le variabili e l'ordinamento della mappa.

Esercizio 3 – Memorie (5 punti)

Avendo a disposizione dei chip di memoria RAM del tipo 1 KByte x 8 bit, sintetizzare una memoria di tipo 4KByte x 16 bit, accessibile al byte e alla parola. Disegnare il montaggio, dettagliando la logica combinatoria eventualmente utilizzata e commentando la soluzione scelta (1pt espansione dati, 2pt espansione indirizzi, 2 pt accesso al byte e alla parola).

Esercizio 4 – Reti sequenziali asincrone (7 punti)

Si consideri il seguente sistema



Descrivere tramite diagramma di flusso (2pt), e modello strutturale (1pt), quindi sintetizzare (4pt) la rete sequenziale asincrona RA in modo tale che la variabile q commuti ogni qual volta si presenta in ingresso al sistema lo stato $x_1x_0 = 11$. Sintetizzare le reti combinatorie in forma SP, e calcolarne il costo a porte.

NOTE

1. Si ricordi che RA è una rete sequenziale asincrona e quindi, quando riceve in ingresso $x_1x_0 = 11$, compie un passo e poi si stabilizza per tutto il tempo in cui lo stato di ingresso permane.
2. Non ci si preoccupi che il tutto risponda alle specifiche fin dall'arrivo del primo stato di ingresso $x_1x_0 = 11$ immediatamente successivo all'accensione.

Esercizio 5 – Reti sequenziali sincronizzate (7 punti)

Si consideri una rete sequenziale sincronizzata di Mealy ritardato con una variabile di ingresso e una variabile di uscita il cui compito è di riconoscere le sequenze di ingresso 1,0,0,1 *interallacciate*.

Si chiede di:

- 1) disegnare il diagramma di flusso (2pt);
- 2) disegnare il modello strutturale (1pt);
- 3) eseguire la sintesi algebrica e circuitale (4pt).

Prima prova in itinere di Calcolatori Elettronici – 08.01.2008

Corsi di Laurea in Ing. Gestionale e Ing. delle Telecomunicazioni

A.A. 2007-2008

Cognome		Nome		Matricola											
---------	--	------	--	-----------	--	--	--	--	--	--	--	--	--	--	--

Istruzioni: Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova, ad eccezione dell'elenco di istruzioni e degli interrupt del processore 8086.

Esercizio 1 – Traduzione da linguaggio ad alto livello ad Assembler (10 punti)

Dato il seguente listato scritto in linguaggio C in linguaggio **Assembler 8086**

1. *Disegnare* lo stato dello stack per ogni chiamata di funzione *dopo l'allocazione delle variabili locali*, e
2. **Tradurre** il listato *commentando* opportunamente il codice scritto.

```
int g = 2;

int one(int a, int b) {
    int m;

    m = two(a/b);

    return m;
}

int two(int k) {
    int p;

    p = k * 2;

    return p;
}
```

```
int main() {
    int i;

    for (i = 0; i > 5; i++) {
        g += one(g, i)
    }

    return g;
}
```

Esercizio 2 – Programmazione Assembler (15 punti)

Utilizzando il linguaggio Assembler 8086 e l'insieme di *interrupt* disponibili sui processori compatibili x86, si realizzi il programma che faccia quanto segue:

1. chiede all'utente di digitare da tastiera un numero compreso tra 01 e 99 (due cifre codificate ASCII) (1pt);
2. memorizza tale numero all'interno di una variabile avente nome simbolico N1, ampia 1 byte (1pt);
3. stampa tale numero nei formati binario ed esadecimale codificati con i caratteri ASCII (2pt);
4. chiede all'utente di digitare da tastiera un altro numero compreso tra 01 e 99 (due cifre codificate ASCII) (1pt);
5. memorizza tale numero all'interno di una variabile avente nome simbolico N2, ampia 1 byte (1pt);
6. stampa tale numero nei formati binario ed esadecimale codificati con i caratteri ASCII (2pt);
7. chiede all'utente di digitare da tastiera il simbolo ASCII dell'operazione che intende effettuare, limitato a '+', '-', '*', '/' (1pt);
8. effettua l'operazione corrispondente memorizzando il risultato nella variabile avente nome simbolico R, ampia 2 byte (2pt);
9. stampa a video il risultato dell'operazione corrispondente nei formati decimale, binario ed esadecimale, tutti codificati con i caratteri ASCII (2pt);
10. quindi chiede all'utente se vuole effettuare un altro calcolo (attende la pressione del tasto 'y'), oppure se vuole stampare nuovamente il risultato (attende la pressione del tasto 'a'), oppure se vuole uscire (attende la pressione del tasto 'z') (1pt).

Esercizio 3 – Questionario Architettura x86 (8 punti)

Rispondere alle seguenti domande in modo conciso ed esauriente.

1. Descrivere il meccanismo di *boot* del processore i8086.
2. Descrivere il salto condizionato dal flag OF e fornire un esempio in linguaggio assembler (1pt).
3. Descrivere i meccanismi di costruzione e distruzione del *record di attivazione* (2pt).
4. Descrivere l'interfaccia parallela di ingresso con *handshake* (schema e commento) (2pt).
5. Descrivere il ciclo di scrittura del bus asincrono (diagramma temporale e commento) (1pt).
6. Descrivere l'arbitro a priorità rotante (1pt).

Recupero Prima prova in itinere di Calcolatori Elettronici – 13.12.2007

Corsi di Laurea in Ing. Gestionale e Ing. delle Telecomunicazioni

A.A. 2007-2008

Cognome		Nome		Matricola							
---------	--	------	--	-----------	--	--	--	--	--	--	--

Istruzioni: Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova.

Esercizio 1 – Rappresentazione dell'informazione (7 punti)

Si considerino i seguenti numeri in base 10:

$$X_{10} = 13$$

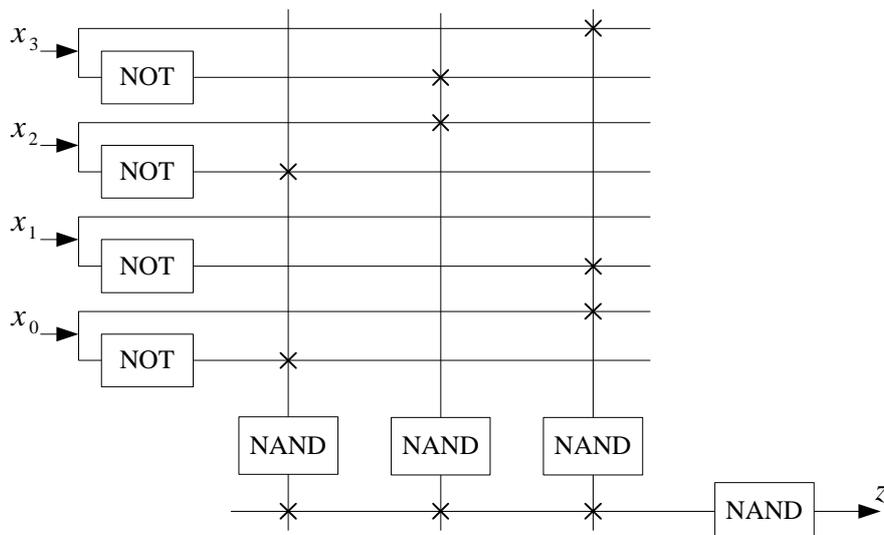
$$Y_{10} = -5,$$

mostrando tutti i passaggi necessari, calcolare:

- 1) la rappresentazione in base 2 di X_{10} (sia X_2), e in complemento a 2 di Y_{10} (sia Y_2) (1pt);
- 2) la somma $S_2 = X_2 + Z_2$ dei due numeri binari ottenuti al punto 1) (1pt);
- 3) la divisione $D_2 = X_2 / S_2$ ed il resto R_2 (1pt);
- 4) la rappresentazione in base 10 di D_2 (1pt);
- 5) la rappresentazione secondo lo standard IEEE/ANSI 754 dei numeri *floating point* su 4 byte di D_2 (2pt);
- 6) la rappresentazione esadecimale del numero calcolato al punto precedente (1pt).

Esercizio 2 – Reti combinatorie (7 punti)

Verificare e giustificare il fatto che il circuito in figura è affetto da Alee del I ordine. Modificare poi il circuito in modo da eliminare dette Alee.



Esercizio 3 – Memorie (5 punti)

Avendo a disposizione dei chip di memoria RAM del tipo 8 MByte x 8 bit, sintetizzare una memoria di tipo 32 MByte x 16 bit, accessibile al byte e alla parola. Disegnare il montaggio, dettagliando la logica combinatoria eventualmente utilizzata e commentando la soluzione scelta (1pt espansione dati, 2pt espansione indirizzi, 2 pt accesso al byte e alla parola).

Esercizio 4 – Reti sequenziali asincrone (7 punti)

Si consideri una *rete sequenziale asincrona* dotata di due ingressi x_1 e x_0 e una uscita z , funzionante nel modo seguente: l'uscita Z , inizialmente a 0, va ad 1 quando si siano verificati un fronte di salita su x_1 ed uno su x_0 , indipendentemente dal loro ordine; Z torna a 0 quando su uno dei due ingressi si presenta un fronte in discesa. Per tale rete, pilotata in modo *fondamentale e senza transizione multiple in ingresso*, si richiede di:

- 1) disegnare il diagramma di flusso (2pt);
- 2) indicare il modello strutturale a cui fare riferimento per la sintesi (1pt);
- 3) eseguire la sintesi algebrica e circuitale (3pt);
- 4) discutere il comportamento della rete in caso di transizioni multiple in ingresso (1pt).

Esercizio 5 – Reti sequenziali sincronizzate (7 punti)

Si consideri una rete sequenziale sincronizzata di Moore con due variabili di ingresso e due variabili di uscita. Interpretando le due variabili di uscita come un numero naturale a due cifre in base due, il comportamento della rete è il seguente:

- quando gli ingressi sono *diversi*, la rete conta in avanti (modulo 4);
- quando gli ingressi sono *uguali*, la rete conta all'indietro (modulo 4).

Si chiede di:

- 1) disegnare il diagramma di flusso (2pt);
- 2) disegnare il modello strutturale in cui la sottorete sequenziale sia composta da flip-flop J-K (1pt);
- 3) riferendosi al modello strutturale considerato al punto precedente, eseguire la sintesi algebrica e circuitale (4pt).

Prima prova in itinere di Calcolatori Elettronici – 13.12.2007

Corsi di Laurea in Ing. Gestionale e Ing. delle Telecomunicazioni

A.A. 2007-2008

Cognome		Nome		Matricola											
---------	--	------	--	-----------	--	--	--	--	--	--	--	--	--	--	--

Istruzioni: Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova, ad eccezione dell'elenco di istruzioni e degli interrupt del processore 8086.

Esercizio 1 – Traduzione da linguaggio ad alto livello ad Assembler (10 punti)

Dato il seguente listato scritto in linguaggio C in linguaggio **Assembler 8086**

1. *Disegnare* lo stato dello stack per ogni chiamata di funzione *dopo l'allocazione delle variabili locali*, e
2. **Tradurre** il listato *commentando* opportunamente il codice scritto.

```
int g = 0;
char c = 'a';

int fun(int a) {
    int m;
    char n;

    n = 'k';
    if ((a % 2) != 0)
        m = run(n);

    return m;
}

int run(char k) {
    int p;

    p = (int) (k - c);

    return p;
}
```

```
int main()
{
    int i;

    for (i = 5; i > 0; i--) {
        g += fun(i)
    }

    return g;
}
```

Esercizio 2 – Programmazione Assembler (15 punti)

Utilizzando il linguaggio Assembler 8086 e l'insieme di *interrupt* disponibili sui processori compatibili x86, si realizzi il programma che faccia quanto segue:

1. chiede all'utente di digitare da tastiera un numero compreso tra 1 e 5 (una cifra codificate ASCII) (1pt);
2. memorizza tale numero all'interno di una variabile avente nome simbolico N, ampia 1 byte (1pt);
3. per N volte chiede all'utente di inserire due cifre codificate ASCII comprese tra 00 e 99 precedendo ogni richiesta dalla stringa posta su una nuova linea dello schermo avente contenuto "Inserire il valore n. #: ", dove al posto di # viene stampato l'indice corrente del numero da inserire (4pt);
4. gli N numeri immessi vengono memorizzati in un vettore VETT1 (1pt);
5. terminata l'immissione degli N numeri, chiama una procedura che riceve in ingresso l'indirizzo del vettore ed il numero N (3pt), e che per ogni numero inserito calcola:
 - a. la rappresentazione binaria e la stampa su schermo in binario codificato ASCII (1pt);
 - b. la rappresentazione binaria del complemento a 1 e la stampa su schermo in binario codificato ASCII (1pt),
 - c. la rappresentazione binaria del complemento a 2 e la stampa su schermo in binario codificato ASCII (1pt),
6. quindi, per moltiplica per 2 ogni numero inserito e lo memorizza in un vettore VETT2 (2pt);
7. ripete il punto 5 per i vettore 2 (1pt).

Esempio di output.

```
Inserire numero: 2
Inserire il valore n.1: 12
Inserire il valore n.2: 07
00001010 11110101 11110100
00000111 11111000 11111001
00010100 11101011 11101000
00001110 11110001 11110010
```

Esercizio 3 – Questionario Architettura x86 (8 punti)

Rispondere alle seguenti domande in modo conciso ed esauriente.

1. Descrivere il meccanismo di memoria segmentata e il meccanismo di traduzione in indirizzo fisico effettivo (1pt) ?
2. Descrivere il salto condizionato da flag ZF e fornire un esempio in linguaggio assembler (1pt).
3. Descrivere e fornire un esempio di indirizzamento indiretto attraverso un registro e indiciato per il processore i8086 (1pt).
4. Descrivere l'interfaccia parallela di uscita con *handshake* (schema e commento) (2pt).
5. Descrivere il ciclo di lettura del bus asincrono (diagramma temporale e commento) (1pt).
6. Descrivere il meccanismo programmazione tramite *interrupt* (2pt).