



**Politecnico di Milano - Dipartimento di Elettronica e informazione**

**Prof. Mauro Negri**

**Fondamenti di Informatica**

**23 novembre 2012**

**I prova in itinere**

Matricola \_\_\_\_\_ Cognome \_\_\_\_\_ Nome \_\_\_\_\_

**Durata prova: 1 ora 30 minuti**

Istruzioni

**LE RISPOSTE DEVONO ESSERE SCRITTE SU QUESTO PLICO (anche sul retro dei fogli)**

Non separare questi fogli. Si può utilizzare la matita.

L'uso di cellulari, libri, eserciziari, appunti o calcolatrici durante lo svolgimento della prova comporta l'annullamento della prova.

Non è possibile uscire dall'aula durante la prova tranne in casi eccezionali.

Esercizio 1 (13 punti) \_\_\_\_\_

Esercizio 2 (7 punti) \_\_\_\_\_

Esercizio 3 (10 punti) \_\_\_\_\_

Punteggio totale (30 punti) \_\_\_\_\_

### Esercizio 1.

1. Riscrivere la seguente porzione di codice C, sostituendo il ciclo for con il ciclo do...while.

```
.....  
int m,sum=0,n=0;  
int main()  
{ for (i=0; i<=100;i++) { scanf("%d", &m); sum=sum+m; n++; }  
  printf("\n sum=%d e n=%d",sum,n);  
}
```

Soluzione:

```
int m,i,sum=0,n=0;  
int main()  
{ i=0;  
  do { scanf("%d", &m); sum=sum+m; n++;i++; } while (i<=100);  
  printf("\n sum=%d e n=%d",sum,n);  
}
```

2. Indicare nella seguente tabella il valore delle variabili B e C dopo l'esecuzione della seguente porzione di codice

if (a<b) if(c>b) b=c; else c=a;

nei seguenti quattro casi

Valori prima degli if			Valori dopo l'esecuzione degli if	
A	B	C	B dopo	C dopo
1	12	98	98	98
0	11	10	11	0
10	22	5	22	10
200	100	-20	100	-20

3. Indicare il valore delle seguenti espressioni logiche con x=70, y=140 e z=210.

Espressione	Valore (falso, vero)
!(x/70>=z/y) && (x<y    x+10>20)	False (1)
y>100 && y<100	False
x>100    x<=100	True
x>z && z>x	False

Specificare inoltre quali sono vere o false indipendentemente dal valore delle variabili con breve motivazione.

x>100 || x<=100 per ogni x è sempre vera

y>100 && y<100 per ogni y sempre falsa perché è impossibile che un valore di y soddisfi le due condizioni logiche opposte; inoltre y=100 non soddisfa entrambe e quindi rimane false..

x>z && z>x sempre false come la precedente

nota (1) i numeri dati non hanno tipo, ma sono integer e quindi o li si considera integer (come sopra) oppure si deve dire che li si considera float affinché si applichi la divisione tra reali.

4. Commentare brevemente la seguente affermazione: “La compilazione di un programma C è quella operazione che traduce un programma sorgente in un programma eseguibile scritto in linguaggio binario”

La compilazione traduce un programma sorgente C in codice oggetto scritto in linguaggio binario, ma il codice oggetto non è eseguibile perché manca ancora il collegamento (linker) alle librerie di sistema e/o dell'utente.

5. Con riferimento al seguente frammento di codice C indicare

```
int x=10,y=20,c;
```

```
...
```

```
int f(int a, int *b) {a=22; *b=100; return 1;}
```

```
int main(){ ... c= f(x, &y);
```

a) quale sia la modalità di passaggio parametri prevista dal C e applicata ad x e y,

-il passaggio parametri in C è per valore sempre

- b ricevendo un indirizzo sta in realtà simulando un passaggio parametri per indirizzo.

b)riportare cosa sarà memorizzato nelle parole di memoria associate ai parametri a e b all'atto dell'esecuzione della funzione f

a= 10

b indirizzo di Y

c) Quali valori assumeranno le variabili x e y dopo l'esecuzione della funzione f

x=10

y=100

6. Indicare la rappresentazione in complemento a 2, utilizzando 10 bit, dei numeri:

A = +111 (base 10)    0001101111

B = - 99 (base 10)    1110011101

7. Eseguire la somma A+B in binario dei numeri interi A=60 e B=30 codificati in complemento a 2, utilizzando 7 bit. Mostrare il risultato in binario come prodotto dall'unità aritmetico-logica e l'equivalente valore in base 10 prodotto. Infine riportare il valore dei bit di carry (riporto) e di overflow dopo l'operazione.

0111100+	A	Premessa: con 7 bit ci sono 128 combinazioni che coprono l'intervallo [-64,+63]
0011110	B	

<u>1011010</u>	Risultato binario
----------------	-------------------

Risultato (base 10): -38

bit carry:            0

bit overflow:        1 (primo bit operandi concorde e primo bit risultato discorde da quello degli operandi)

8. Indicare la rappresentazione in base 8 e 16 di -87 (base 10)

base 8 =    -127

base 16 =   -57

9. Indicare quale numero decimale rappresenta la seguente sequenza di bit codificata in complemento a 2: 1100.1001.0110

Ricalcolo 001101101010 ossia - 874

10. Data la definizione della variabile float v= -79,8, descrivere la sua rappresentazione in virgola mobile in base allo standard IEEE754 semplice precisione FP32 descritto a lezione;

a) Definire prima il numero nel formato normalizzato intermedio (TRONCARE parte decimale alla terza cifra decimale):

- 1,246 \* 2<sup>6</sup>

b) Codificare poi il numero normalizzato usando il formato FP32:

S (1 bit di segno): 1

M (23 bit per la mantissa – fermarsi ai primi 6 bit): 001111

E (8 bit per l'esponente): 10000101

c) Indicare quale sia il vero valore (in base 10) descritto dalla sequenza di bit memorizzata in S, M ed E del punto b) precedente (si consiglia di esprimere la mantissa in termini frazionari).

$$-(1 + 1/8 + 1/16 + 1/32 + 1/64) * 2^6 = -79$$

**Esercizio 2** Dato il seguente programma

```
#include <stdio.h>
```

```
#define m 4
```

```
int *v[m], a=11,b=22,c=33,d=44, *p, **r = &v[2],i;
```

```
int main() { v[0]=&a; v[1]=&b; v[2]=&c;
```

```
v[3]=&d; p=v[2];
```

```
*v[2]=10;
```

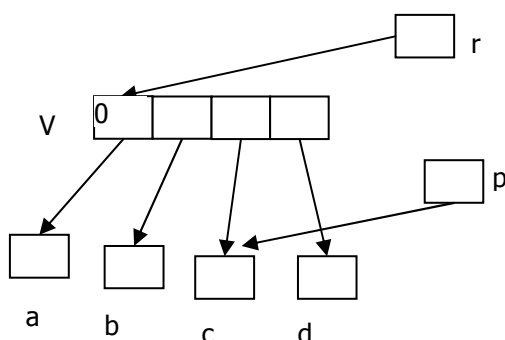
```
*p=*p+15;
```

```
**r=50; r=r-2; *r=v[0];
```

```
for (i=0; i<=3;i++) printf("\n%d ",*v[i]);
```

```
}
```

a) Riportare con un disegno lo stato delle variabili globali v, p ed r prima dell'esecuzione del ciclo finale di stampa; gli indirizzi contenuti in variabili puntatore vanno schematizzati con un arco orientato che parte dalla variabile puntatore e raggiunge l'esplicita variabile puntata.



b) Riportare il risultato prodotto a video dall'esecuzione del programma (rispettare l'esatto formato di stampa)

11  
22  
50  
44

**Esercizio 3.** Scrivere un programma C che definisca una matrice M quadrata NxN con N pari di caratteri e poi proceda a caricare le ultime N/2 righe come mostrato in figura (si supponga che la matrice contenga inizialmente in tutte le caselle il carattere ' '(blank)):

X									X
X	X							X	X
X	X	X					X	X	X
X	X	X	X			X	X	X	X
X	X	X	X	X	X	X	X	X	X

```
#include <stdio.h>
```

```
#define N 10
```

```
char M[N][N]; int r,c;
```

```
int main()
```

```

{ //non richiesta - solo per esecuzione
for (r=0;r<N;r++) for (c=0;c<N;c++) M[r][c]=' ';

for (r=N/2;r<N;r++)

    { for (c=0; c<= r-N/2; c++) M[r][c]='x';
      for (c=(N-1)- (r-N/2);c<=N-1; c++)M[r][c]='x';}

//non richiesta - solo per esecuzione
for (r=0;r<N;r++)
{ for (c=0;c<N;c++) putchar(M[r][c]); printf("\n");}
}

```