

Programmazione

Preparazione al compito (Soluzioni)

11 dicembre 2007

Note.

- Useremo la notazione xxx_b per indicare che il numero xxx è da intendersi espresso in base b . Ad esempio, 100_{10} è il numero cento (è scritto in base 10), mentre 100_2 è il numero 4 (è scritto in base 2).
- Ove non sia esplicitamente indicato, per “programma” si intende semplicemente il corpo della funzione main.

Rispondete alle seguenti domande.

1. Usate lo spazio in fondo alla pagina per giustificare le risposte alle seguenti domande:

- (a) Scrivete in binario il numero esadecimale 31_{16} ;

Risposta: **00110001₂**

- (b) Scrivete in ottale il numero binario 11010110_2 ;

Risposta: **326₈**

- (c) Rappresentate il numero negativo -35 in *complemento a due* su 8 bit;

Risposta: **11011101**

- (d) Sul pianeta di Robulon, si usa un alfabeto che comprende 60 simboli, e la lingua non richiede l'uso di spazi fra le parole o segni di interpunzione. Il più famoso poema epico robulonesco, dal titolo "Ran-ba doi" (che significa "La saga della lasagna"), è composto da 100 capitoli, ognuno di 50 versi, e ciascun verso è costituito da 50 lettere. Quanto spazio servirà per memorizzarlo in forma digitale? (Esprimate la risposta in byte)

Risposta: **187 500**

- (a) $31_{16} = 0011\ 0001_2$.
- (b) $11010110_2 = 011\ 010\ 110_2 = 326_8$.
- (c) $-35 + 128 = 93$ e $93_{10} = 1011101_2$, da cui $-35 = 11011101$.
- (d) Ogni simbolo richiede 6 bit (infatti $2^5 = 32 < 60$ mentre $2^6 = 64$). Quindi $100 \cdot 50 \cdot 50 = 250000$ caratteri richiedono $250000 \cdot 6 = 1500000$ bit, ovvero 187500 byte.

2. Assumete che x e y siano variabili **int**, e che il loro valore *prima di ciascuno dei seguenti assegnamenti* sia rispettivamente 15 e 33. Dite quale sarà il loro valore *dopo* gli assegnamenti indicati:

Assegnamento	x	y
$x=(y-3+x)$	18	18
$x=(x+y>50? x+y : y-x)$	18	33
$x=y/5$	6	33
$y=(\text{int})(1-y)$	15	-32

3. Considerate il seguente programma C:

```

int x, y, z;
scanf( "%d%d", &x, &y );
z = 0;
for ( z = x; z < y; z += x );
printf( "%d", z );

```

- Cosa stampa il programma quando $x = 5$ e $y = 12$? **15**
- Cosa stampa il programma quando $x = 31$ e $y = 5$? **31**

4. Scrivete un programma che chieda ripetutamente all'utente di inserire degli interi, e che termini quando l'utente inserisce un valore pari. Al termine dell'inserimento, il programma deve stampare il numero di valori negativi introdotti. Ad esempio, se l'utente inserisce *61 -301 1 503 -2* il programma deve stampare 1 (visto che c'è un unico valore negativo introdotto).

```
int x, n;
n = 0;
do {
    scanf( "%d", &x );
    if ( x % 2 != 0 && x < 0 )
        n++;
} while ( x % 2 != 0 );
printf( "%d\n", n );
```

5. Sia data la seguente definizione di variabili:

```
int x = 15;  
int y = 31;
```

Dire qual è il valore di verità delle seguenti espressioni (indicate *vero* se l'espressione ha un valore numerico diverso da zero, *falso* se l'espressione ha un valore numerico uguale a zero) e qual è il contenuto delle variabili x e y dopo la valutazione delle espressioni:

Espressione	Valore di verità	x	y
<code>y > x && x++ > y++</code>	falso	16	32
<code>x++ > y++ y > x</code>	vero	16	32

6. Scrivete un programma che, su un certo input intero (diciamo t), produca un output di t righe e t colonne come nell'esempio seguente:

```
Intero? 10
10 9 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 10
```

Potete assumere che l'input sia un numero pari. (Scrivete il programma nella facciata che segue)

```
int t, i, j;
printf( "Intero?_" );
scanf( "%d", &t );
for ( i = 0; i < t / 2; i++ ) {
    for ( j = t; j > 0; j-- )
        printf( "%d_", j );
    printf( "\n" );
    for ( j = 1; j <= t; j++ )
        printf( "%d_", j );
    printf( "\n" );
}
```

7. Scrivete un programma che chieda ripetutamente all'utente di inserire degli interi, e che termini quando l'utente inserisce il valore 0. Al termine dell'inserimento, il programma deve stampare la media dei valori dispari introdotti; la media deve essere stampata come valore floating point. Ad esempio, se l'utente inserisce *61 300 301 503 0* il programma deve stampare *288.3333333333* (che è $(61 + 301 + 503)/3$).

```
int x, s, n;
s = n = 0;
do {
    scanf( "%d", &x );
    if ( x % 2 != 0 ) {
        s += x;
        n++;
    }
} while ( x != 0 );
printf( "%f\n", (float)s / n );
```


8. Scrivete una funzione che, su un certo input intero (diciamo t), produca un disegno di t righe e t colonne, come nell'esempio seguente (qui $t = 10$):

```
*****
****  ****
***   ***
**    **
*     *
*     *
**    **
***   ***
****  ****
*****
```

Potete assumere che l'argomento sia un numero pari.

```
void paint( int t ) {
    int x, i;
    for ( x = t/2; x >= 1; x-- ) {
        for ( i = 0; i < x; i++ ) putchar( '*' );
        for ( i = 0; i < t - 2*x; i++ ) putchar( ' ' );
        for ( i = 0; i < x; i++ ) putchar( '*' );
        printf( "\n" );
    }
    for ( x = 1; x <= t/2; x++ ) {
        for ( i = 0; i < x; i++ ) putchar( '*' );
        for ( i = 0; i < t - 2*x; i++ ) putchar( ' ' );
        for ( i = 0; i < x; i++ ) putchar( '*' );
        printf( "\n" );
    }
}
```

9. Scrivete una funzione C che implementi la seguente funzione (definita su tutti i numeri naturali):

$$f(x) = \begin{cases} x & \text{se } x \text{ è pari} \\ f(x-1) + f(x+1) & \text{altrimenti.} \end{cases}$$

```
void f( int x ) {  
    if ( x % 2 == 0 )  
        return x;  
    else  
        return f(x-1)+f(x+1);  
}
```