Programmazione
I compitino (Vers. C)

18 novembre 2009

Cognome .... Nome ....
Matricola ....

Note.

- In alcune delle seguenti domande verrà usato il vostro numero di matricola. Scrivete il vostro numero nel seguente schema, una cifra per ogni spazio:

(A)	(B)	(C)	(D)	(E)	(F)

  Negli esercizi, useremo le lettere (A), (B), (C)... per riferirci alle corrispondenti cifre del numero di matricola. Ad esempio, "il numero $(D)(E)(F)$" è il numero costituito dalle ultime tre cifre del vostro numero di matricola.

- Ove non sia esplicitamente indicato, per "programma" si intende semplicemente il corpo della funzione main.

Rispondete alle seguenti domande.

1. Assumete che x e y siano variabili **int**, e che il loro valore *prima di ciascuno dei seguenti assegnamenti* sia rispettivamente $(A)(F)$ e $(E)(D)$. Dite quale sarà il loro valore *dopo* gli assegnamenti indicati:

Assegnamento	x	y
x=(y=3+x)		
x=(x+y>50? x+y : y−x)		
x=y/5		
y∗=1−y		

2. Considerate il seguente programma C:

```
int x, y, z, c;
scanf( "%d%d", &x, &y );
z = 1;
c = 0;
while( z < x ) {
    z *= y;
    c++;
}
printf( "%d\n", c );
```

- Cosa stampa il programma quando $x = (A)(E)$ e $y = (F)$? ....
- Cosa stampa il programma quando $x = (E)(D)$ e $y = (A)$? ....

3. Sia data la seguente dichiarazione di variabili:

```
int x = (A)(F);
int y = (D)(E);
```

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
`y>x && x++>y++`			
`y>x || x++>y++`			

4. Il seguente programma dovrebbe leggere una stringa di caratteri e stamparla al contrario. Il programma contiene degli errori; segnateli con un cerchietto.

```
#include <stdio.h>
#define MAX_STR=500

int main( void ) {
 int i, n;
 char[MAX_STR] x;

 gets( x );
 n = strlen( x );
 for ( i = n-1; i >= 0; i-- )
  printf( "%c", i );
 printf( "\n" );
 return 0;
}
```

5. Scrivete un programma che chieda ripetutamente all'utente di inserire degli interi, e che termini quando l'utente inserisce un valore negativo. Al termine dell'inserimento, il programma deve stampare il prodotto dei multipli di cinque introdotti. Ad esempio, se l'utente inserisce *61 60 5 9 18 -3* il programma deve stampare 300 (che è $60 \cdot 5$).

6. Scrivete un programma che, su un certo input intero (diciamo t), produca un output di t righe, come nell'esempio seguente (su ogni riga vengono stampati i numeri da 1 a t, tranne alcuni che sono sostituiti da asterischi; sulla prima riga si tratta dei multipli di 1, sulla seconda riga dei multipli di 2, sulla terza i multipli di 3 ecc.):

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

7. Scrivete un programma che chieda ripetutamente all'utente di inserire degli interi (al massimo 100), e che termini quando l'utente inserisce il valore 0. Al termine dell'inserimento, il programma deve stampare l'elenco dei valori pari introdotti. Ad esempio, se l'utente inserisce *4 61 300 301 503 0* il programma deve stampare *4 300*.

8. Scrivete un programma che legga un numero intero $n \geq 100$, seguito da n altri interi, e poi ne stampi, per ciascuno, la cifra con cui termina, come nell'esempio seguente:

```
Quanti valori? 6
 153  8123  123122  419  444  6
3
3
2
9
4
6
```

9. Scrivete un programma che, letta una stringa (di al massimo 500 caratteri), la ristampi nel seguente modo: prima tutte le lettere minuscole, poi tutto il resto. Ad esempio:

`Stringa?` **questa stringa 24 non contiene// solo lettere! ma anche altro...**
`questastringanoncontienesololetteremaanchealtro  24  //  !   ...`