Programmazione
I compitino (Vers. A)

1 dicembre 2010

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

2. Considerate il seguente programma C:

```
int x, y, z, c;
scanf( "%d%d", &x, &y );
z = 0;
c = y;
while ( z < x ) {
    c += x;
    z++;
}
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>2*x && x++>--y`			
`y>2*x || x++>--y`			

4. Il seguente programma dovrebbe leggere una stringa e stamparne i soli caratteri alfabetici minuscoli. Il programma contiene degli errori; segnateli con un cerchietto.

```
#include "stdio.h"
#include "string.h"
#define MAX_STR 500;

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

 gets( x );
 n = strlen();
 for ( i = 0; i <= n; i++ )
  if ( x[ i ] >= 'a' && x[ i ] <='z')
   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 il valore 0. Al termine dell'inserimento, il programma deve stampare la somma dei multipli di dieci introdotti. Ad esempio, se l'utente inserisce *50 33 100 5 40 40 0* il programma deve stampare 230.

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

```
Intero?  10
0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2
3 3 3 3 3 3 3
4 4 4 4 4 4
5 5 5 5 5
6 6 6 6
7 7 7
8 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 introdotti che sono seguiti da un valore maggiore. Ad esempio, se l'utente inserisce *61 300 301 36 34 49 47 0* il programma deve stampare *61 300 34*.

8. Scrivete un programma che legga un numero intero $n \geq 100$, seguito da n altri interi, e poi ne stampi, per ciascuno, il numero di cifre:

```
Quanti valori? 6
 153  8123  123122  419  444  6
3
4
6
3
3
1
```

9. Scrivete un programma che, letta una stringa (di al massimo 500 caratteri), la ristampi al contrario. Ad esempio:

```
Stringa?  questa stringa sarebbe piuttosto lunga
agnul otsottuiup ebberas agnirts atseuq
```