Programmazione
I compitino (Vers. A)

14 novembre 2013

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+2)		
x=y%y? x+1 : y++		
x+=(y*2)/3		
y+=−x		

2. Considerate il seguente programma C:

```
int x, y, z, c;
scanf( "%d", &x );
z = c = 0;
while ( z < x ) {
    c++;
    z += 2;
}
printf( "%d\n", z+c );
```

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

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 && y++<x++`			
`y>2*x+1 || x++>(y-=3)`			

4. Il seguente programma dovrebbe leggere una stringa e contarne gli spazi. Il programma contiene degli errori; segnateli con un cerchietto.

```
#include <stdio>
#include <string>
#define MAX_STR=500

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

  x = gets();
  n = strlen( x );
  c = 0;
  for ( i = 1; i <= n; i++ )
    if ( x[ i ] = '␣' )
      c++;
  printf( c, "\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 delle cifre delle unità. Ad esempio, se l'utente inserisce *51 33 107 5 40 41 0* il programma deve stampare 17 (che è $1 + 3 + 7 + 5 + 0 + 1$).

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

```
Intero?  10
**********<<<<<<<<<<
 **********<<<<<<<<<
  **********<<<<<<<<
   **********<<<<<<<
    **********<<<<<<
     **********<<<<<
      **********<<<<
       **********<<<
        **********<<
         **********<
```

7. Scrivete un programma che chieda ripetutamente all'utente di inserire degli interi e che termini quando l'utente inserisce un valore inferiore a 10. Al termine dell'inserimento, il programma deve stampare la somma delle cifre dei valori inseriti escluso l'ultimo. Ad esempio, se l'utente inserisce *61 61 95 97 99 300 336 337 7* il programma deve stampare *90* (che è $6+1+6+1+9+5+9+7+9+9+3+0+0+3+3+6+3+3+7$).

8. Scrivete un programma che legga un numero intero $n \geq 100$, seguito da n altri interi compresi fra 1 e 26, e poi ne stampi, per ciascuno, la lettera corrispondente (1=a, 2=b, ...).

```
Quanti valori? 6
 3  5  1  3  6  6
ceacff
```

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

```
Stringa? ***questa stringa conti*ene degli ast*rischi!
questa stringa conti*ene degli ast*rischi!
```