



## Esercizio 1 ( Analisi di Codice )

Che output produce l'esecuzione del seguente codice?

```
1 int main()
2 {
3     int i;
4     for(i=20;i<40;i++)
5         if(g(i)) printf("%i\n",i);
6 }
```

A large grid of 20 columns and 10 rows, intended for drawing.

## Esercizio 2 ( Analisi di Codice )

Che output produce l'esecuzione del seguente codice? E' l'unico possibile?

```

1  int main()
2  {
3      int i;
4      printf("%i\n",l(12));
5      printf("%i\n",l(13));
6      printf("%i\n",l(14));
7      printf("%i\n",l(128));
8      printf("%i\n",l(128000));
9  }

```

[illegible]



### Esercizio 4 ( Analisi di Codice )

Che output produce l'esecuzione del seguente codice?

```

1  #include <stdio.h>
2
3  #define DIM (102*102)
4
5  int main()
6  {
7      int m[DIM][DIM];
8      int i,s;
9      for (i=0;i<DIM*DIM;i++)
10     {
11         int a=i%DIM, b=i/DIM;
12         if (a*b==0)
13             m[a][b]=(a+b)%2;
14         else
15             m[a][b]=m[a-1][b-1];
16     }
17
18     s=0;
19     for (i=0;i<DIM;i++)
20         s=s+m[i][i];
21     printf ("%i\n",s);
22
23     s=0;
24     for(i=0;i<DIM-1;i++)
25         s=s+m[i][i+1];
26     printf ("%i\n",s);
27
28     s=0;
29     for (i=0;i<DIM*DIM;i++)
30         s=s+m[i%DIM][i/DIM];
31     printf ("%i\n",s);
32
33 }

```

[illegible]

### Esercizio 5 ( Sintesi di Codice )

Si consideri la funzione  $S(n)$  che, ricevuto in ingresso un numero  $n$ , ritorna la somma degli esponenti della scomposizione di  $n$  in fattori primi. Ad esempio, visto che  $72 = 8 \cdot 9 = 2^3 \cdot 3^2$ ,  $S(72) = 3 + 2 = 5$ . Un archeologo ha trovato un'antica implementazione, in linguaggio C, della funzione  $S$ . Purtroppo, a causa dell'azione impietosa del tempo, il codice è giunto a noi incompleto. Completa il frammento recuperato, utilizzando *rigorosamente* al più un carattere per quadretto. Si possono chiamare le funzioni presenti nel prologo (senza è difficile...)

```
int S(int n)
{
    int i, sum=0;
    for(i=1; i<=1(n); i++)
```

[illegible]

```
    return sum;
}
```