

```

1  #include <stdio.h>
2
3  int g(int a)
4  {
5      int i;
6      for (i=a-1;i>1 && a%i!=0 ;i--)
7          ;
8      return i==1 ? 1 : 0;
9
10 }
11
12 int h(int n, int k)
13 {
14     int i;
15     for(i=2;i<=n;i++)
16     {
17         if (g(i) && n%i==0) k--;
18         if (k==0) return i;
19     }
20     return 0;
21 }
22
23 int l(int a)
24 {
25     int i;
26     for(i=1;h(a,i)!=0;i++)
27         ;
28     return i-1;
29 }
30
31 int q(int n, int d)
32 {
33     if(n%d!=0) return 0;
34     return 1+q(n/d,d);
35 }

```

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 graph paper with 20 columns and 10 rows. The grid is composed of small squares, with a slightly larger square at the top left corner, likely for a title or header. The grid is empty and ready for use.

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(12800));
9  }
```

[illegible]

Esercizio 3 (Analisi di Codice)

Che output produce l'esecuzione del seguente codice?

```

1  int main()
2  {
3      int i;
4      for(i=2;i<10;i++)
5          printf("%i:%i ",i,q(120,i));
6
7      printf("\n");
8      printf("%i\n",q(810,3));
9      printf("%i\n",q(13*29*31*5*3,31));
10     printf("%i\n",q(32*32*16,2));
11 }

```

[illegible]

Esercizio 4 (Analisi di Codice)

Che output produce l'esecuzione del seguente codice?

```

1  #include <stdio.h>
2
3  #define DIM (100*100)
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<=n; i++)
```

[illegible]

```
    return sum;
}
```