

## Esercizio: numeri primi

---

**Scrivere un programma che verifichi se un naturale  $N$  è primo.**

**Specifica di I livello** (ispirata al *Crivello di Eratostene*):  
dividere  $n$  per tutti i naturali  $k \leq \sqrt{n}$ . Se nessuno risulta essere divisore, allora  $n$  è primo.

**Specifica di II livello**: se  $n$  è 1, 2 o 3 allora è primo. Altrimenti, se è un numero pari, allora non è primo. Se invece  $n$  non è pari ed è  $>3$ , allora tentare tutti i possibili divisori da 3 a  $\sqrt{n}$ .

## Esercizio: numeri primi

---

Introduciamo una variabile *risultato* a cui assegnamo il valore 0 se il numero non è primo, 1 altrimenti.

```
main()
{
    int risultato;

    /* corpo del programma */

    if (risultato == 1) printf("\nIl numero %d è primo\n",n);
    else printf("\nIl numero %d non è primo\n",n);
}
```

## Esercizio: numeri primi

---

A parole:

se  $n >= 1$  e  $n <= 3$  allora *risultato* = 1;  
altrimenti:  
se  $n$  è pari allora *risultato* = 0;  
altrimenti:  
*risultato* = 1;  
prova tutti i divisori fino a  $\sqrt{n}$  e, se ne trovi uno, allora *risultato* = 0.

```
#include <stdio.h>
#include <math.h>

main() {
    int n,k,risultato,m;

    printf("\nInserire un numero intero: ");
    scanf("%d",&n);

    if (n>=1 && n<=3) risultato = 1;
    else if (n%2 == 0) risultato = 0;
    else {
        m = sqrt(n);
        risultato = 1;
        for (k=3; k<=m; k+=2) if (n%k == 0) risultato = 0;
    }

    if (risultato == 1) printf("\nIl numero %d è primo\n",n);
    else printf("\nIl numero %d non è primo\n",n);
}
```

## Osservazioni

---

Il ciclo di controllo dei divisori puo' essere arrestato alla prima occorrenza di un divisore:

```
for (k=3; k<=m && risultato != 0; k+=2)
    if (n%k == 0) risultato = 0;
```

Oppure:

```
m = sqrt(n); risultato = 1; k=3;
while (k<=m && risultato != 0){
    if (n%k == 0) risultato = 0;
    k += 2;
}
```

## Commenti

---

La funzione **sqrt** opera su *double* e restituisce un *double*, quindi nel programma "forziamo" una conversione in *int* (e il compilatore genera un *warning*).

Per usare la funzione **sqrt** e' necessario includere il file con le dichiarazioni delle funzioni matematiche:

```
#include <math.h>
```