



# SUMA ȘI NUMĂRUL DIVIZORILOR INDICATORUL LUI EULER

# CE ESTE INDICATORUL LUI EULER

- Indicatorul lui Euler sau funcția lui Euler se notează cu  $\varphi(n)$  (unde  $n$  este un număr natural nenul) și contorizează numerele întregi pozitive mai mici sau egale cu  $n$  și prime cu acesta

# FORMULĂ

- Dacă  $n = p_1^{k_1} * p_2^{k_2} * \dots * p_r^{k_r}$  este descompunerea în factori primi, distincți, a lui  $n$ , unde  $p_i$  sunt numere prime distincte atunci:
- $\phi(n) = (p_1 - 1) * p_1^{k_1-1} * (p_2 - 1) * p_2^{k_2-1} * \dots * (p_r - 1) * p_r^{k_r-1}$
- Se mai poate scrie:
- $\varphi(n) = n * (1 - \frac{1}{p_1}) * (1 - \frac{1}{p_2}) * \dots * (1 - \frac{1}{p_k}) = n * \frac{p_1-1}{p_1} * \frac{p_2-1}{p_2} * \dots * \frac{p_k-1}{p_k}$

# PROGRAM

```
int n, r, d = 2;
cin >> n; r = n;
while (n > 1)
{ if (n % d == 0)
    { r = r / d * (d - 1);
      while (n % d == 0) n /= d;
    }
  d++;
  if (d * d > n) d = n;
}
cout << r;
```

# CALCUL INDICATOR EULER PENTRU MAI MULTE NUMERE

```
const int nm=1e4; /// numarul de valori pt care calculam
int c[nm], i, j;
/// nr. de numere prime cu mai mici ca i este i
for(i=1; i<nm; i++) c[i] = i;
for(i=2; i<nm; i++)
    if(c[i] == i)
        for(j=i; j<nm; j+=i)
            c[j] = c[j] / i * (i-1);
for(i=1; i<nm; i++) /// afisam
    cout << c[i] << " ";
```

# NUMĂRUL DIVIZORILOR

• Dacă  $n = p_1^{k_1} * p_2^{k_2} * \dots * p_r^{k_r}$

atunci numărul divizorilor lui  $n$  este:  $(k_1 + 1) * (k_2 + 1) * \dots * (k_r + 1)$

# PROGRAM NUMĂRUL DIVIZORILOR

```
int n, d, nrDiv=1, p;  
cin >> n;  
for (d=2; d*d<=n; d++)  
{  
    for (p=1; n%d==0; p++) n/=d;  
    nrDiv*=p;  
}  
if (n>1) nrDiv *= 2;  
cout << nrDiv;
```



# SUMA DIVIZORILOR

• Dacă  $n = p_1^{k_1} * p_2^{k_2} * \dots * p_r^{k_r}$

atunci suma divizorilor lui  $n$  este:  $\frac{p_1^{k_1+1}-1}{p_1-1} * \frac{p_2^{k_2+1}-1}{p_2-1} * \dots * \frac{p_r^{k_r+1}-1}{p_r-1}$



# SUMA DIVIZORILOR

```
int n, d, sumDiv=1, p;  
cin >> n;  
for (d=2; d*d<=n; d++)  
{  
    for (p=d; n%d==0; p*=d) n/=d;  
    sumDiv*= (p-1) / (d-1);  
}  
if (n>1) sumDiv *= (n*n-1) / (n-1);  
cout << sumDiv;
```