Matemática

Números Primos

Integrantes

→ Pedro A C Santos
→ João Pimenta
→ Ítalo Rosa
→ Bruna Lima
→ Leonardo Oliveira

Método Convencional

Divisão por números anteriores;

Ex.: 17, 15, 97

17 – 17%2, 17%3, 17%4 ... 17 é primo

15 – 15%2, 15%3 ... 15 não é primo

97 – sqrt(97) ... 97 é primo

Problema

Algoritmos de múltiplas verificações

Primo muito grande

Crivo de Eratóstenes

- Encontre um primo, em seguida analise no intervalo definido marcando todos os múltiplos desse número.

Crivo de Eratóstenes

- Ex.: 2 (10):

  2*2= 4 2*3=6 2*4= 8 2*5=10

  3*3=9

  .·. 4,6,8,9,10 → não são múltiplos de nenhum, por consequência, o restante é primo

Crivo de Eratóstenes: *implementação*

```
typedef unsigned long long ll;
#define TAM 1e7
ll _sieve_size_ = 0;
bitset<(int)TAM> vb;
vector<int> primes;
```

```
void sieve(ll _size_){
    _sieve_size_ = _size_ +1;
    vb.set();
    vb.set(0,0); vb.set(1,0);
    _sieve_size_ += 2;

    for(ll i=2; i<_sieve_size_;
i++)
        if(vb[i]){
            for(ll j=i*i;
j<_sieve_size_; j+=i)
                vb[j]=0;
            primes.push_back(i);
        }
}
```

Exemplo

- URI : 2180

Código: *Uri-2180*

```
// URI: 2180
#include <bits/stdc++.h>
#include <bitset>
using namespace std;   // 2^(4*8) - 1
max int value

#define mp make_pair
#define fst first
#define snd second
#define pb push_back

typedef long long ll;
typedef std::pair<int,int> pii;
typedef std::vector<int> vi;

bitset<100000> mark; //60000
vi primes;

void sieve(ll s){
s+=1;
mark.set(); mark.set(0,0); mark.set(1,0);
for(ll i=0; i<=s; i++){
  if(mark[i]){
  for(ll j=i*i; j<=s; j+=i) mark.set(j,0);
  primes.pb((int) i);
}}
}
```

Código: *Uri-2180*

```
bool isPrime(ll pos){
if(pos <= 60000) return mark[pos];
for(int i=0; i<sqrt(primes.size()); i++)
if(pos%primes[i] == 0) return false;
return true;
}
```

```
int main(){
sieve(60000);
int weight, i=10, sum=0;
scanf("%i", &weight);
while(i){
while(!isPrime(weight)) weight++;
sum+=weight++;
i--;
}
printf("%i km/h\n", sum);
sum=60e6/sum;
printf("%i h / %i d\n", sum, sum/24);
return 0;
}
```