

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int i, num1, num2, min, hcf=1;

    scanf("%d%d", &num1, &num2);

    min = (num1<num2) ? num1 : num2;

    for(i=1; i<=min; i++){
        if(num1%i==0 && num2%i==0){
            hcf = i;
        }
    }

    printf("HCF of %d and %d = %d\n", num1, num2, hcf);

    return 0;
}

```

/*

Analysis:

Here we can take two numbers a & b gcd of a & b:

While b is not equal 0 $p = b$ $b' = a \bmod b$ $a' = p$

Here, it divides the big number by a small number.

If the remainder is not equal to 0, it goes to next step.

Then divide the divisor by the remainder.

Repeat the process until the remainder is equal to 0,

when the remainder is 0 that divisors are gcd.

Suppose we have 24 & 18 .

At first, dividing the large number 24 by the small number 18.

Here $24/18 = 1$ & remainder 6.

Then dividing the divisor 18 by the remainder 6, $18/6=3$ remainder 0.

So, gcd of 24 & 18 is 6.

Time complexity :

```
int gcd(int a, int b){
```

```
    If (a == 0) return b;
```

```
    return gcd(b%a,a)}
```

So the time complexity is $\log(\min(a,b))$

```
*/
```