

Prime

目的:输出100以内的素数及其和

思路:在大于1的自然数中,除了1和它本身以外不再有其他因数的自然数是素数

方法一

- 已知第一个素数是2,那么从2开始,每个自然数 i 除以大于2小于他本身的自然数 j .
- 如果存在可以整除的自然数 j ,那么 i 就不是素数,反之则是.
- 然后输出所有素数.

代码:

```
#include <stdio.h>
int main(void)
{
    printf("100以内的素数表及其和\n");

    int i ;
    int j ;
    int sum = 0;

    for (i = 2; i < 100; i++) {
        int n = 1 ;
        for (j = 2; j < i; j++) {
            if (i % j == 0) {
                n = 0;
                break;
                /*节约循环次数,如果有一个符合条件的i,那么n = 0后可以直接break.*/
            }
        }
        if (n == 1) {
            printf("%d\n", i);
            sum = sum + i;
            /*素数加一起.*/
        }
    }
    printf("100以内所有素数和为%d", sum);
    return 0;
}
```

方法二

使用数组和函数

- 定义100以内的素数数组 `prime[100] = {2}` /*易知`prime[0] = 2`*/
- 用 `isPrime` 函数从3开始,一直到100依次判断每一个数是否是素数
- 如果是,讲这个素数写入素数数组的对应位置

代码块:

```

int prime[100] = {2};
int i = 3;
int count = 1;
/*已知prime[0] = 2,故count从1开始,直到count=99时,得到prime[100]*/

while(count < 100){
    if (isPrime(/*具体先不写*/) ){
        prime[count++] = i; /*小套路写法*/
    }
    i++;
}

```

- 定义 `isPrime` 函数用于判断 `i` 是否是素数

函数定义:

- 判断 `x` 是否是素数只需用 `x` 除以所有比他小的素数
- 如果能整除,则不是素数,反之则是.
- 例如:

i	比 i 小的素数	运算	是/否是素数
3	2	$3 \% 2 \neq 0$	是
4	2,3	$4 \% 2 = 0$	不是
5	2,3	$5 \% 2 \neq 0$ $5 \% 3 \neq 0$	是
6	2,3,5	$6 \% 2 = 0$ $6 \% 3 = 0$	不是

代码块:

```

int isPrime(int x,int knownPrimes[],int numberOfKnownPrimes)
{
    int ret = 1;
    int i;
    for (i = 0;i < numberOfKnownPrimes;i++) {
        if (x % knownPrimes[i] == 0) {
            ret = 0;
            break;
        }
    }
    return ret;
}
/*C 语言把任何非零和非空的值假定为 true, 把零或 null 假定为 false*/

```

函数引用代码块:

```
isPrime(i,prime,count);
```

完整代码

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

```

int isPrime(int x, int knownPrimes[], int numberOfKnownPrimes);

int main(void)
{
    int prime[100] = { 2 };
    int count = 1;
    int i = 3;

    while (count < 100) {
        if (isPrime(i, prime, count)) {
            prime[count++] = i;
        }
        i++;
    }
    for (i = 0; i < 100; i++) {
        printf("%d", prime[i]);
        if ((i + 1) % 5) {
            printf("\t");
        } else { printf("\n"); }
    } /*排版*/
    return 0;
}

int isPrime(int x, int knownPrimes[], int numberOfKnownPrimes)
{
    int ret = 1;
    int i;
    for (i = 0; i < numberOfKnownPrimes; i++) {
        if (x % knownPrimes[i] == 0) {
            ret = 0;
            break;
        }
    }
    return ret;
}

```