**SEIVE OF SUNDARAM ALGORITHM…**

This is actually Simple Deterministic Algorithm to find all prime numbers up to a specified number.

To Understand This Algorithm,

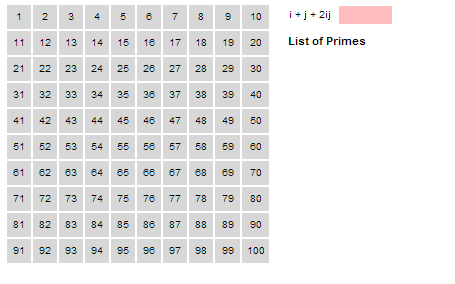
Let’s Consider an Example:

Here we are considering numbers from 1 to 100

First write down all the numbers with the given range.

Its helpful to find all primes numbers less than a given number.

Observe the Below Diagram to get Clarity about this Algorithm.



Select a Specified Number

Select i,j where i,j belongs to the Specified Number and 1<=i<=j;

And the formula identified from the table is i+j+2\*i\*j<Specified Number.

For example,

For i=1,j=1:

i+j+2\*i\*j=1+1+2\*1\*1=4

now mark the number 4 and next increment the value of j;

now j becomes 2;

i+j+2\*i\*j=1+2+2\*1\*2=7

mark the number 7 and increment the value of j;

Like this continue the process of incrementing the value of j but the value should always be less than Specified Number.

Once the Value crossed the Specified Number then Continue to increase the value of i.

Remember that the value obtained should not cross Specified Number.

Next use the formula,

2(p)+1.

Consider p values are Left numbers other than marked numbers.

From Above Diagram, left numbers start from 1 so,

2(1)+1=3

3 is Prime number;

Next p=2;

2(2)+1=5

5 is Prime number;

Next p=3;

2(3)+1=7

7 is Prime number;

P=5;

2(5)+1=11

11 is Prime number;

So, like that continue the process with Left out numbers from the Table

Analysis:

I observed that this is the efficient Algorithm to find all the prime numbers.

It’s Actually easy to understand the nature of prime numbers through this algorithm..

It’s is applicable everywhere.

C PROGRAM

#include <stdio.h>

void main() {

int arraySize, i, j, x;

int numberPrimes = 0;

printf("Input a positive integer to find all the prime numbers up to and including that number: ");

scanf("%d", &arraySize);

int n = arraySize / 2;

int size;

int isPrime[arraySize + 1];

int TheseArePrime = 0;

for (i = 0; i < n; ++i) {

isPrime[i] = i;

}

for (i = 1; i < n; i++) {

for (j = i; j <= (n - i) / (2 \* i + 1); j++) {

isPrime[i + j + 2 \* i \* j] = 0;

}

}

if (arraySize > 2) {

isPrime[TheseArePrime++] = 2;

}

for (i = 1; i < n; i++) {

if (isPrime[i] != 0) {

isPrime[TheseArePrime++] = i \* 2 + 1;

}

}

size = sizeof isPrime / sizeof(int);

for (x = 0; x <= size; x++) {

if (isPrime[x] != 0) {

printf("%d \t", isPrime[x]);

numberPrimes++;

} else {

break;

}

}

printf("\nNumber of Primes: %d", numberPrimes);

}