

# Resumen del desarrollo de la práctica

La práctica nos pide hallar los números primos entre el 2 y el 65535. Primero de todo, al hacer un programa básico, descubrimos que hay 6541 números primos en ese intervalo. Con este dato inicializamos un array como variable global, que será recorrido por una variable, también global “k”, introduciendo en la posición correspondiente los números primos debidamente calculados.

Para calcular debidamente los números primos, disponemos de un bucle “for” que irá recorriendo todos los números del 2 al 65535; dentro de este “for” habrá un “if” que, si cumple la sentencia, guarda el numero hallado en el array antes descrito.

La función que se encarga de obtener los números primos se llama esPrimo. En esta función, tenemos otro “for” que recorre las posiciones del array donde hemos guardado los números primos ya evaluados. Por definición un numero primo se puede obtener al dividir por factores que son números primos menores a la raíz cuadrada del número evaluado.

Dentro de este “for” tenemos la condición `if (numero % primos[i]==0)`; con esta sentencia vemos si el numero evaluado es divisible por los números primos, ya hallados anteriormente, y el resto de esa división da cero, si ese fuera el caso el número no sería primo y por tanto no se guardaría en el array.

Si, por el contrario, ninguna división cumple esa condición, entonces, el numero evaluado es un numero primo, y la función esPrimo devolvería un “1”, con lo que el número se guardaría en el array (y se incrementaría el valor de k, con la sentencia `k++`). Con esto se pasaría a evaluar el siguiente número.

La ejecución del programa, para hallar todos los números primos del 2 al 65535, tarda 1,5502