

Primera Evaluación I Término 2010-2011. Julio 6, 2010

Tema 2. (25 puntos) Omirp se define como un número primo que al invertir sus dígitos da otro número primo.

Escriba un algoritmo para determinar si un número n tiene la característica de ser un número Omirp.

Ejemplo: **1597** es número primo,
 Se invierte sus dígitos **7951**
 7951 también es primo,
 Entonces el número **1597** es un número omirp

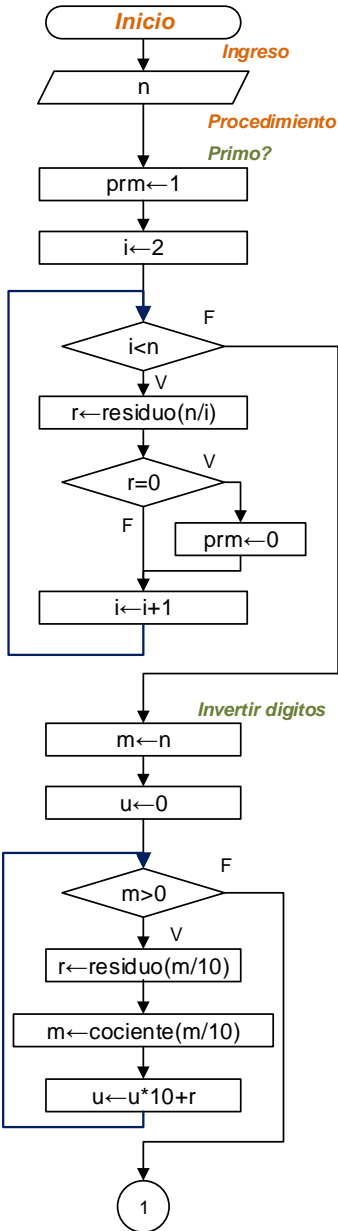
Rúbrica: Validar si n es primo (7 puntos), invertir los dígitos del número (10 puntos), validar si el nuevo número es primo (3 puntos), respuesta y algoritmo integrado (5 puntos)

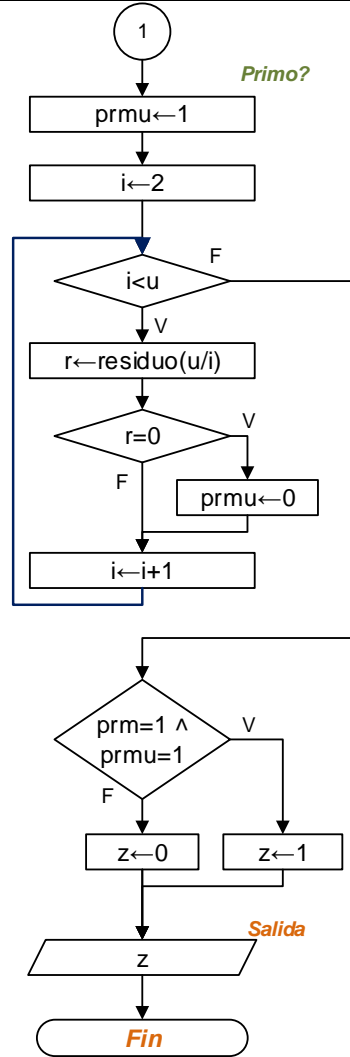
Propuesta de solución:

Nota: Omirp se lee de derecha a izquierda como primo.

Repasar algoritmo de validación de números primos, también el algoritmo para invertir dígitos de un número. En este ejercicio es necesario usar ambos. Primero para verificar si el número a verificar es primo, luego para invertir sus dígitos y finalmente verificar si el número con dígitos invertidos también es primo. Si ambos son primos entonces el número es omirp.

Solución con el lazo mientras-Repita.

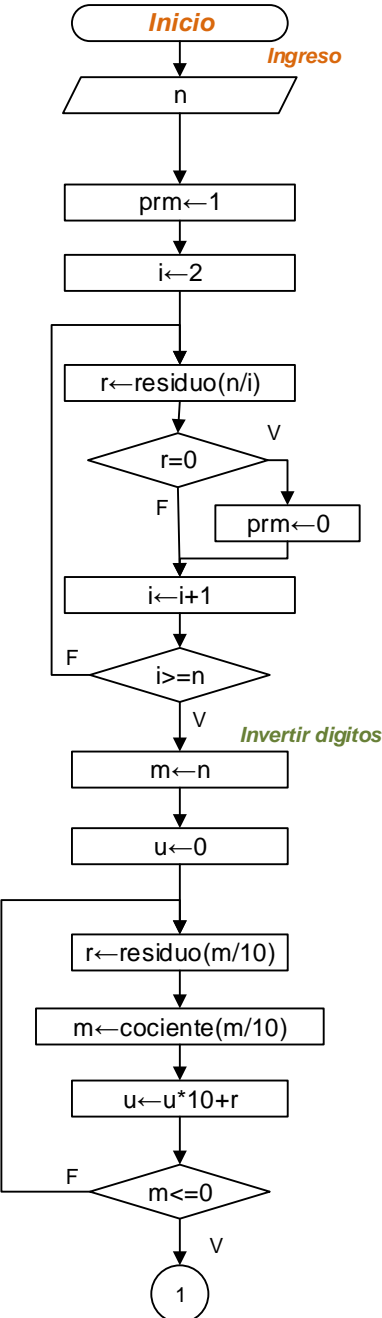
SEUDO CODIGO	DIAGRAMA DE FLUJO	Python
<p>Inicio</p> <p>Ingresar el número a validar como omirp</p> <p>Algoritmo Primo Bandera: Número es primo</p> <p>Analizar divisores desde el 2</p> <p>Mientras divisores sean menores que n</p> <p>Determinar el residuo n/i</p> <p>Si no hay residuo es divisible</p> <p>Bandera: No es primo</p> <p>Siguiente divisor</p> <p>Repita</p> <p>Algoritmo Invertir Dígitos m es el número a invertir sus dígitos</p> <p>número con dígitos invertidos</p> <p>Mientras existan dígitos en m</p> <p>Extraer un dígito (unidad)</p> <p>Eliminar un dígito (unidad)</p> <p>Construye nuevo número con dígito invertido</p> <p>Repita</p>	 <pre> graph TD Inicio([Inicio]) --> Ingreso[/Ingreso n/] Ingreso --> Prm[Procedimiento prm ← 1] Prm --> I[Procedimiento i ← 2] I --> Loop1(()) Loop1 --> Cond1{i < n} Cond1 -- F --> Loop1 Cond1 -- V --> Residuo[r ← residuo(n/i)] Residuo --> Cond2{r = 0} Cond2 -- V --> Prm0[prm ← 0] Prm0 --> Loop1 Cond2 -- F --> IncI[i ← i + 1] IncI --> Loop1 Loop1 --> Invert[Invertir digitos m ← n] Invert --> U[u ← 0] U --> Loop2(()) Loop2 --> Cond3{m > 0} Cond3 -- F --> End((1)) Cond3 -- V --> Residuo2[r ← residuo(m/10)] Residuo2 --> Cociente[m ← cociente(m/10)] Cociente --> U2[u ← u * 10 + r] U2 --> Loop2 </pre>	<pre> # ICM00794-Fund. Computación - FCNM-ESPOL # 1ra Evaluación I Term 2010-2011 # Tema 2. Numero OMIRP # Propuesta: edelros@espol.edu.ec n=int(input('numero a validar omirp:')) # Procedimiento prm=1 i=2 while (i<n): r=n%i if r==0: prm=0 i=i+1 m=n u=0 while (m>0): r=(m%10) m=(m//10) u=u*10+r </pre>

<p>Usar Algoritmo para validar número primo</p> <p>Si el numero original es primo y el número con dígitos invertidos es primo</p> <p>Verdadero: Es omirp Sino Falso: No es omirp</p> <p>Mostrar respuesta</p> <p>Fin</p>		<pre> prmu=1 i=2 while (i<u): r=(u%i) if r==0: prmu=0 i=i+1 if (pru==1 and prmu==1): z=1 else: z=0 # Salida print(z) </pre>
---	--	---

Ejecución del algoritmo: omirp.py

<pre> >>> Número a validar omirp: 4134 0 </pre>	<pre> >>> Número a validar omirp: 1597 1 </pre>
--	--

Solución usando el lazo repita-Hasta

SEUDO CODIGO	DIAGRAMA DE FLUJO	Python
<p>Inicio</p> <p>Ingresar el número a validar como omirp</p> <p>Algoritmo Primo Bandera: Número es primo</p> <p>Analizar divisores desde el 2</p> <p>Repetir</p> <p> Determinar el residuo n/i</p> <p> Si no hay residuo</p> <p> Bandera: No es primo</p> <p> Siguiente divisor</p> <p>Hasta divisor previo a n</p> <p>Algoritmo Invertir Dígitos m es el número a invertir</p> <p>número con dígitos invertidos</p> <p>Repita</p> <p> Extraer un dígito (unidad)</p> <p> Eliminar un dígito (unidad)</p> <p> Construir el nuevo número con dígito invertido</p> <p>Hasta que no queden dígitos</p>	 <pre> graph TD Inicio([Inicio]) -- Ingreso --> n[/n/] n --> prm[prm ← 1] prm --> i[i ← 2] i --> r["r ← residuo(n/i)"] r --> r0{"r=0"} r0 -- V --> prm0[prm ← 0] r0 -- F --> iplus[i ← i+1] iplus --> in["i >= n"] in -- F --> r in -- V --> m["m ← n"] m --> u[u ← 0] u --> r2["r ← residuo(m/10)"] r2 --> m2["m ← cociente(m/10)"] m2 --> u2["u ← u*10+r"] u2 --> m3{"m <= 0"} m3 -- F --> r2 m3 -- V --> 1((1)) </pre>	<pre> # ICM00794-Fund. Computación - FCNM-ESPOL # 1ra Evaluación I Term 2010-2011 # Tema 2. Numero OMIRP # Propuesta: edelros@espol.edu.ec n=int(input('numero a validar omirp:')) #Procedimiento prm=1 i=2 while not(i>=n): r=n%i if r==0: prm=0 i=i+1 m=n u=0 while not(m<=0): r=(m%10) m=(m//10) u=u*10+r </pre>

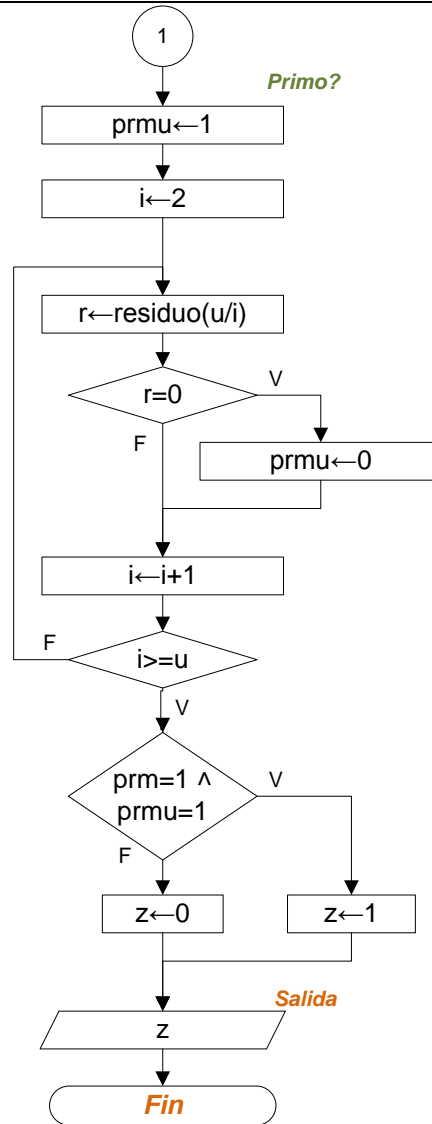
Usar **Algoritmo para validar número primo**

Si el número original es primo y el número con dígitos invertidos es primo

Verdadero: Es omirp
Sino
Falso: No es omirp

Mostrar respuesta

Fin



prmu=1

i=2

while not(i>=u):

r=(u%i)

if r==0:

prmu=0

i=i+1

if (pru==1 and prmu==1):

z=1

else:

z=0

print(z)