

Segunda Tarea

Criptografía y Seguridad

José Demian Jimenez
314291707

Carlos Cruz Rangel
312285823

Victor Hugo Gallegos Mota
316160456

9 de noviembre del 2022

1 Problema 1

Dado el siguiente número $n = 1, 148, 289, 976, 600, 001$ aplique una prueba de primalidad en la cual se ocupe testigo (testigo de Fermat, testigo de Euler, testigo fuertes,...) y cite cual es.

1.1 A

Determina si el número $n = 1, 148, 289, 976, 600, 001$ es primo con una prueba de primalidad probabilística vista en clase. Para el caso de ser primo explique como llega a tal conclusión.

Para determinar esto usaremos el test de fermat. Explicare en que consiste esta prueba antes de pasar al código del cual me apoye para hacer esto: Primero para calcular $((a^n) \bmod p)$ iniciando con un resultado de 1, ahora actualizamos a si se cumple que $a \geq p$. ahora mientras $n > 0$ si n es impar entonces actualizamos el resultado al valor de multiplicar el resultado $*a$. de lo contrario a toma el valor de $a^2 \bmod n$.

```
def potencia(a, n, p):  
    res = 1  
    a = a % p  
    while n > 0:  
        if n % 2:  
            res = (res * a) % p  
            n = n - 1  
        else:  
            a = (a ** 2) % p
```

```

        n = n // 2
    return res % p

```

función para calcular $((a^n) \bmod p)$ en python

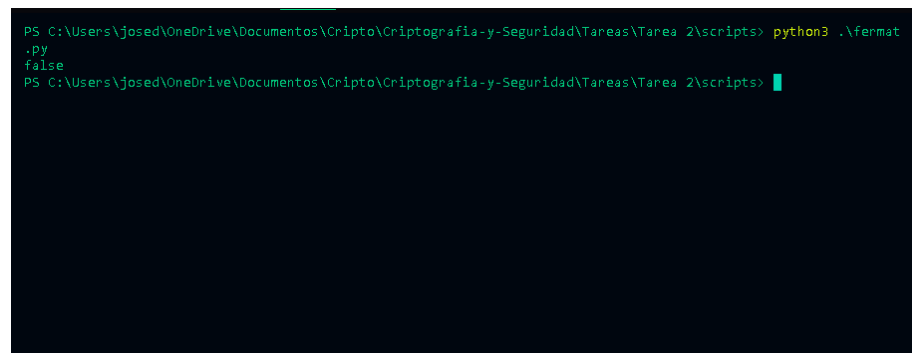
Ahora se codifico una función para comprobar si el numero es primo o compuesto en el que recibimos el numero a operar y el numero de k iteraciones a realizar, nota entre mas iteraciones sean mas probabilidad habrá de obtener el resultado correcto. Esta función hace uso de la función potencia que es la que emplea el algoritmo de fermat.

```

def esPrimo(n, k):
    if n == 1 or n == 4:
        return False
    elif n == 2 or n == 3:
        return True
    else:
        for i in range(k):
            a = random.randint(2, n - 2)
            if potencia(a, n - 1, n) != 1:
                return False
        return True

```

Al ejecutar este numero obtenemos el siguiente resultado:



```

PS C:\Users\josed\OneDrive\Documentos\Cripto\Criptografia-y-Seguridad\Tareas\Tarea 2\scripts> python3 .\fermat
.py
false
PS C:\Users\josed\OneDrive\Documentos\Cripto\Criptografia-y-Seguridad\Tareas\Tarea 2\scripts>

```

Figure 1: Resultado de ejecución

El algoritmo indica que n no es primo por lo tanto es un numero compuesto.

1.2 B

En caso de ser compuesto de explícitamente la iteración y su testigo determina que es compuesto.

Para sacar los factores de n usaremos el algoritmo de Pollard ya que se nos permite operar con numeros grandes , para esto primero se codifico una funcion auxiliar para sacar el mcd en python.

```
def mcd(a, b):  
    if b == 0:  
        return a  
    else:  
        return mcd(b, a % b)
```

Despues para descomponer el numero usamos el algoritmo de Ro de Polard y se obtubieron los siguientes resultados:

```
def pollard():  
  
    n = 1148289976600001 # Valor de n, cambiarlo si  
    se desea  
    x = 2 # Valor inicial de x  
    y = 2 # Valor inicial de y  
    d = 1 # Valor inicial de d  
    iteraciones = 0  
    while d == 1:  
        x = (x**2 + 1) % n # f(x) = x^2 + 1  
        y = (y**2 + 1) % n # f(y) = y^2 + 1  
        y = (y**2 + 1) % n # f(y) = y^2 + 1  
        d = mcd(abs(x-y), n) # d = mcd(|x-y|, n)  
        iteraciones += 1  
        print("Iteraciones realizadas:",  
              iteraciones, d)  
        global p  
        p = d  
        global q  
        q = n//d  
        print("p =", d)  
        print("q =", n//d)  
        print("f(x) = x^2 + 1")
```

Al ejecutar obtenemos los siguientes resultados:

```
Iteraciones realizadas: 1 1  
Iteraciones realizadas: 2 1  
Iteraciones realizadas: 3 1  
Iteraciones realizadas: 4 1  
Iteraciones realizadas: 5 1
```

```
Iteraciones realizadas: 6 1
Iteraciones realizadas: 7 1
Iteraciones realizadas: 8 1
Iteraciones realizadas: 9 1
Iteraciones realizadas: 10 1
Iteraciones realizadas: 11 1
Iteraciones realizadas: 12 1
Iteraciones realizadas: 13 1
Iteraciones realizadas: 14 1
Iteraciones realizadas: 15 1
Iteraciones realizadas: 16 1
Iteraciones realizadas: 17 1
Iteraciones realizadas: 18 1
Iteraciones realizadas: 19 1
Iteraciones realizadas: 20 1
Iteraciones realizadas: 21 1
Iteraciones realizadas: 22 1
Iteraciones realizadas: 23 1
Iteraciones realizadas: 24 1
Iteraciones realizadas: 25 1
Iteraciones realizadas: 26 1
Iteraciones realizadas: 27 1
Iteraciones realizadas: 28 1
Iteraciones realizadas: 29 1
Iteraciones realizadas: 30 1
Iteraciones realizadas: 31 1
Iteraciones realizadas: 32 1
Iteraciones realizadas: 33 1
Iteraciones realizadas: 34 1
Iteraciones realizadas: 35 1
Iteraciones realizadas: 36 1
Iteraciones realizadas: 37 1
Iteraciones realizadas: 38 1
Iteraciones realizadas: 39 1
Iteraciones realizadas: 40 1
Iteraciones realizadas: 41 1
Iteraciones realizadas: 42 1
Iteraciones realizadas: 43 1
Iteraciones realizadas: 44 1
Iteraciones realizadas: 45 1
Iteraciones realizadas: 46 1
Iteraciones realizadas: 47 1
Iteraciones realizadas: 48 1
Iteraciones realizadas: 49 1
Iteraciones realizadas: 50 1
Iteraciones realizadas: 51 1
```

```
Iteraciones realizadas: 52 1
Iteraciones realizadas: 53 1
Iteraciones realizadas: 54 1
Iteraciones realizadas: 55 1
Iteraciones realizadas: 56 1
Iteraciones realizadas: 57 1
Iteraciones realizadas: 58 1
Iteraciones realizadas: 59 1
Iteraciones realizadas: 60 1
Iteraciones realizadas: 61 1
Iteraciones realizadas: 62 1
Iteraciones realizadas: 63 1
Iteraciones realizadas: 64 1
Iteraciones realizadas: 65 1
Iteraciones realizadas: 66 1
Iteraciones realizadas: 67 1
Iteraciones realizadas: 68 1
Iteraciones realizadas: 69 1
Iteraciones realizadas: 70 1
Iteraciones realizadas: 71 1
Iteraciones realizadas: 72 1
Iteraciones realizadas: 73 1
Iteraciones realizadas: 74 1
Iteraciones realizadas: 75 1
Iteraciones realizadas: 76 1
Iteraciones realizadas: 77 1
Iteraciones realizadas: 78 1
Iteraciones realizadas: 79 1
Iteraciones realizadas: 80 1
Iteraciones realizadas: 81 1
Iteraciones realizadas: 82 1
Iteraciones realizadas: 83 1
Iteraciones realizadas: 84 1
Iteraciones realizadas: 85 1
Iteraciones realizadas: 86 1
Iteraciones realizadas: 87 1
Iteraciones realizadas: 88 1
Iteraciones realizadas: 89 1
Iteraciones realizadas: 90 1
Iteraciones realizadas: 91 1
Iteraciones realizadas: 92 1
Iteraciones realizadas: 93 1
Iteraciones realizadas: 94 1
Iteraciones realizadas: 95 1
Iteraciones realizadas: 96 1
Iteraciones realizadas: 97 1
```

```
Iteraciones realizadas: 98 1
Iteraciones realizadas: 99 1
Iteraciones realizadas: 100 1
Iteraciones realizadas: 101 1
Iteraciones realizadas: 102 1
Iteraciones realizadas: 103 1
Iteraciones realizadas: 104 1
Iteraciones realizadas: 105 1
Iteraciones realizadas: 106 1
Iteraciones realizadas: 107 1
Iteraciones realizadas: 108 1
Iteraciones realizadas: 109 1
Iteraciones realizadas: 110 1
Iteraciones realizadas: 111 1
Iteraciones realizadas: 112 1
Iteraciones realizadas: 113 1
Iteraciones realizadas: 114 1
Iteraciones realizadas: 115 1
Iteraciones realizadas: 116 1
Iteraciones realizadas: 117 1
Iteraciones realizadas: 118 1
Iteraciones realizadas: 119 1
Iteraciones realizadas: 120 1
Iteraciones realizadas: 121 1
Iteraciones realizadas: 122 1
Iteraciones realizadas: 123 1
Iteraciones realizadas: 124 1
Iteraciones realizadas: 125 1
Iteraciones realizadas: 126 1
Iteraciones realizadas: 127 1
Iteraciones realizadas: 128 1
Iteraciones realizadas: 129 1
Iteraciones realizadas: 130 1
Iteraciones realizadas: 131 1
Iteraciones realizadas: 132 1
Iteraciones realizadas: 133 1
Iteraciones realizadas: 134 1
Iteraciones realizadas: 135 1
Iteraciones realizadas: 136 1
Iteraciones realizadas: 137 1
Iteraciones realizadas: 138 1
Iteraciones realizadas: 139 1
Iteraciones realizadas: 140 1
Iteraciones realizadas: 141 1
Iteraciones realizadas: 142 1
Iteraciones realizadas: 143 1
```

```
Iteraciones realizadas: 144 1
Iteraciones realizadas: 145 1
Iteraciones realizadas: 146 1
Iteraciones realizadas: 147 1
Iteraciones realizadas: 148 1
Iteraciones realizadas: 149 1
Iteraciones realizadas: 150 1
Iteraciones realizadas: 151 1
Iteraciones realizadas: 152 1
Iteraciones realizadas: 153 1
Iteraciones realizadas: 154 1
Iteraciones realizadas: 155 1
Iteraciones realizadas: 156 1
Iteraciones realizadas: 157 1
Iteraciones realizadas: 158 1
Iteraciones realizadas: 159 1
Iteraciones realizadas: 160 104711
p = 104711
q = 10966278391
f(x) = x^2 + 1
None
```

Por lo tanto los factores de n son : $1, 148, 289, 976, 600, 001 = 104711 \cdot 10966278391$

2 Problema 2

Mediante el algoritmo de ro de Pollard para enteros descomponga $n = 7784099$

NOTA PARA EL AYUDANTE : Este problema se solucionó con 2 scripts realizados en python (ambos documentados), al final del ejercicio se adjunta las salidas en consola de la ejecución de ambos, con todo el procedimiento del algoritmo, también se adjunta en la carpeta junto con este pdf los archivos para probar su funcionamiento

2.1 A

Dé la función semi-aleatoria empleada

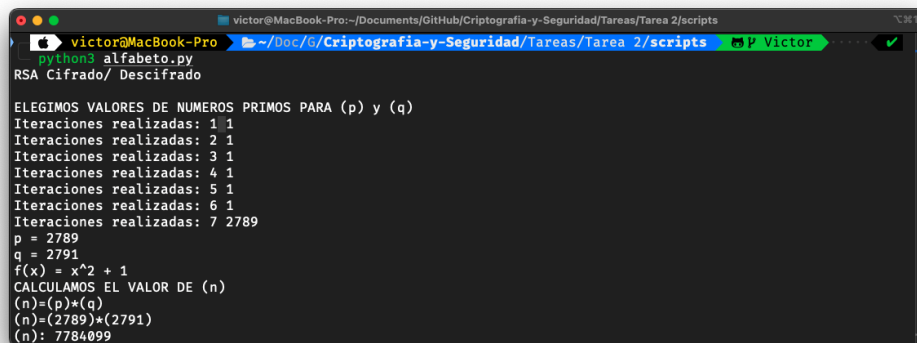
Respuesta:

1. Empezamos con los valores iniciales de $(x, y) = (2, 2)$ y $d = 1$.
2. Calculamos el valor de $x = f(x) = (x^2 + 1) \bmod n$.
3. Calculamos el valor de $y = f(f(y)) = ((y^2 + 1)^2 + 1) \bmod n$.
4. Calculamos el valor de $d = \text{mcd}(|x - y|, n)$, donde $\text{mcd}(a, b)$ es el máximo común divisor de a y b .
5. Si $d = 1$, volvemos al paso 2. De lo contrario, hemos encontrado un factor de n y podemos detener el algoritmo.
6. Imprimimos los valores de p y q , que son los factores de n .

2.2 B

Número de iteración en el cual fue exitoso el algoritmo y factor encontrado.

Respuesta: Iteración #7 contando de 1 a 7 fue exitosa para encontrar el factor



```
victor@MacBook-Pro:~/Documents/GitHub/Criptografia-y-Seguridad/Tareas/Tarea 2/scripts
python3 alfabeto.py
RSA Cifrado/ Descifrado

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
Iteraciones realizadas: 1 1
Iteraciones realizadas: 2 1
Iteraciones realizadas: 3 1
Iteraciones realizadas: 4 1
Iteraciones realizadas: 5 1
Iteraciones realizadas: 6 1
Iteraciones realizadas: 7 2789
p = 2789
q = 2791
f(x) = x^2 + 1
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2789)*(2791)
(n): 7784099
```

Figure 2: Resultado de ejecución

2.3 C

Descifre el siguiente mensaje RSA, el cual esta en unicode:

Llave pública RSA = (7784099, 7), mensaje cifrado = 6308199

Respuesta: E = 6308199

Para este caso se descifró con el archivo `alfabetoVerificacion.py` para descifrar la letra E directamente y se obtuvo el siguiente resultado

```
ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2791)*(2789)
(n): 7784099

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2791-1)*(2789-1)
(phi): 7778520

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=7

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(7)^-1 mod (7778520)
(d): 6667303

Generando la llave publica y privada . . .
Llave publica: (7, 7784099)
Llave privada: (6667303, 7784099)

Ingrese el mensaje a encriptar: E
Mensaje a encriptar: E

Mensaje encriptado: 6308199

para descifrar mensaje con llave privada (6667303,
7784099) . . .
Mensaje descifrado: E

El cifrado y descifrado funciona correctamente!
```

Llave pública RSA = (7784099, 11), mensaje cifrado = 5536286

Respuesta: $s = 5536286$

Para este caso se descifró primero con el archivo alfabeto.py para encontrar en que letra del alfabeto coincide posteriormente se utilizo alfabetoVerificacionCriba.py para verificar, para descriptar la letra s directamente y se obtuvo el siguiente resultado

```
ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2791)*(2789)
(n): 7784099

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2791-1)*(2789-1)
(phi): 7778520

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=11

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(11)^-1 mod (7778520)
(d): 3535691

Generando la llave publica y privada . . .
Llave publica: (11, 7784099)
Llave privada: (3535691, 7784099)

Ingrese el mensaje a encriptar: s
Mensaje a encriptar: s

Mensaje encriptado: 5536286

para descriptar mensaje con llave privada (3535691,
7784099) . . .
Mensaje descriptado: s

El cifrado y descifrado funciono correctamente!
```

lave pública RSA = (7784099, 13), mensaje cifrado = 159060

Respuesta: Espacio " " = 159060

Para este caso se descifró con el archivo alfabetoVerificacion.py para desencriptar la el símbolo espacio " " directamente y se obtuvo el siguiente resultado

```
ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2791)*(2789)
(n): 7784099

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2791-1)*(2789-1)
(phi): 7778520

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=13

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(13)^-1 mod (7778520)
(d): 5983477

Generando la llave publica y privada . . .
Llave publica: (13, 7784099)
Llave privada: (5983477, 7784099)

Ingrese el mensaje a encriptar:
Mensaje a encriptar:

Mensaje encriptado: 159060

para desencriptar mensaje con llave privada (5983477,
7784099) . . .
Mensaje desencriptado:

El cifrado y descifrado funciona correctamente!
```

Llave pública RSA = (7784099, 19), mensaje cifrado = 6724396
Respuesta: r = 6724396

Para este caso se descifró primero con el archivo alfabeto.py para encontrar en que letra del alfabeto coincide posteriormente se utilizo alfabetoVerificacion.py para verificar, para desencriptar la letra s directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2791)*(2789)
(n): 7784099

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2791-1)*(2789-1)
(phi): 7778520

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=19

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(19)^-1 mod (7778520)
(d): 2046979

Generando la llave publica y privada . . .
Llave publica: (19, 7784099)
Llave privada: (2046979, 7784099)

Ingrese el mensaje a encriptar: r
Mensaje a encriptar: r

Mensaje encriptado: 6724396

para desencriptar mensaje con llave privada (2046979,
7784099) . . .
Mensaje desencriptado: r

El cifrado y descifrado funciona correctamente!

Llave pública RSA = (7784099, 23), mensaje cifrado = 26176
Respuesta: u = 26176

```

Para este caso se descifró primero con el archivo alfabeto.py para encontrar en que letra del alfabeto coincide posteriormente se utilizo alfabetoVerificacion.py para verificar, para desencriptar la letra s directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2791)*(2789)
(n): 7784099

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2791-1)*(2789-1)
(phi): 7778520

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=23

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(23)^-1 mod (7778520)
(d): 7102127

Generando la llave publica y privada . . .
Llave publica: (23, 7784099)
Llave privada: (7102127, 7784099)

Ingrese el mensaje a encriptar: u
Mensaje a encriptar: u

Mensaje encriptado: 26176

para desencriptar mensaje con llave privada (7102127,
7784099) . . .
Mensaje desencriptado: u

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (7784099, 29), mensaje cifrado = 1117219
Respuesta: $t = 1117219$

Para este caso se descifró primero con el archivo alfabeto.py para encontrar en que letra del alfabeto coincide posteriormente se utilizó alfabetoVerificacion.py para verificar, para desencriptar la letra s directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)

```

```

CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2791)*(2789)
(n): 7784099

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2791-1)*(2789-1)
(phi): 7778520

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=29

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(29)^-1 mod (7778520)
(d): 1609349

Generando la llave publica y privada . . .
Llave publica: (29, 7784099)
Llave privada: (1609349, 7784099)

Ingrese el mensaje a encriptar: t
Mensaje a encriptar: t

Mensaje encriptado: 1117219

para desencriptar mensaje con llave privada (1609349,
7784099) . . .
Mensaje desencriptado: t

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (7784099, 37), mensaje cifrado = 6925326
Respuesta: i = 6925326

Para este caso se descifró primero con el archivo alfabeto.py para encontrar en que letra del alfabeto coincide posteriormente se utilizó alfabetoVerificacion.py para verificar, para desencriptar la letra s directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)

```

```

(n)=(2791)*(2789)
(n): 7784099

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2791-1)*(2789-1)
(phi): 7778520

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=37

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(37)^-1 mod (7778520)
(d): 2312533

Generando la llave publica y privada . . .
Llave publica: (37, 7784099)
Llave privada: (2312533, 7784099)

Ingrese el mensaje a encriptar: i
Mensaje a encriptar: i

Mensaje encriptado: 6925326

para desencriptar mensaje con llave privada (2312533,
7784099) . . .
Mensaje desencriptado: i

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (7784099, 43), mensaje cifrado = 7550806
Respuesta: $n = 7550806$

Para este caso se descifró primero con el archivo alfabeto.py para encontrar en que letra del alfabeto coincide posteriormente se utilizó alfabetoVerificacion.py para verificar, para desencriptar la letra s directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2791)*(2789)
(n): 7784099

```

```

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2791-1)*(2789-1)
(phi): 7778520

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=43

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(43)^-1 mod (7778520)
(d): 4884187

Generando la llave publica y privada . . .
Llave publica: (43, 7784099)
Llave privada: (4884187, 7784099)

Ingrese el mensaje a encriptar: n
Mensaje a encriptar: n

Mensaje encriptado: 7550806

para desencriptar mensaje con llave privada (4884187,
7784099) . . .
Mensaje desencriptado: n

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (7784099, 47), mensaje cifrado = 1525454
Respuesta: a = 1525454

Para este caso se descifró primero con el archivo alfabeto.py para encontrar en que letra del alfabeto coincide posteriormente se utilizó alfabetoVerificacion.py para verificar, para desencriptar la letra s directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2791)*(2789)
(n): 7784099

CALCULAMOS EL VALOR DE (phi)

```



```

(phi)=(p-1)*(q-1)
(phi)=(2791-1)*(2789-1)
(phi): 7778520

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=47

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(47)^-1 mod (7778520)
(d): 1158503

Generando la llave publica y privada . . .
Llave publica: (47, 7784099)
Llave privada: (1158503, 7784099)

Ingrese el mensaje a encriptar: a
Mensaje a encriptar: a

Mensaje encriptado: 1525454

para desencriptar mensaje con llave privada (1158503,
7784099) . . .
Mensaje desencriptado: a

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (7784099, 49), mensaje cifrado = 4142333

Respuesta: Punto " . " = 4142333

Para este caso se descifró con el archivo `alfabetoVerificacion.py` para desencriptar el signo de puntuación "." directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2791)*(2789)
(n): 7784099

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2791-1)*(2789-1)
(phi): 7778520

```

```
ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=49

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(49)^-1 mod (7778520)
(d): 2063689

Generando la llave publica y privada . . .
Llave publica: (49, 7784099)
Llave privada: (2063689, 7784099)

Ingrese el mensaje a encriptar: .
Mensaje a encriptar: .

Mensaje encriptado: 4142333

para desencriptar mensaje con llave privada (2063689,
7784099) . . .
Mensaje desencriptado: .

El cifrado y descifrado funciona correctamente!
```

Respuesta completa: Es rutina.

2.4 Flujo de ejecucion de los scripts con funcion Pollard rho



```
1 def pollard():
2     """
3     Algoritmo de Pollard para factorizar números grandes.
4     return: Tupla con los valores de p y q
5     rtype: tuple
6     """
7     n = 7784099 # Valor de n, cambiarlo si se desea
8     x = 2 # Valor inicial de x
9     y = 2 # Valor inicial de y
10    d = 1 # Valor inicial de d
11    iteraciones = 0
12    while d == 1:
13        x = (x**2 + 1) % n # f(x) = x^2 + 1
14        y = (y**2 + 1) % n # f(y) = y^2 + 1
15        y = (y**2 + 1) % n # f(y) = y^2 + 1
16        d = mcd(abs(x-y), n) # d = mcd(|x-y|, n)
17        iteraciones += 1
18        print("Iteraciones realizadas:", iteraciones, d)
19
20    # Agregar los factores a las variables globales p y q
21    global p
22    p = d
23    global q
24    q = n//d
25    # Imprimir los valores de p y q
26    print("p =", d)
27    print("q =", n//d)
```

Figure 3: Función pollard rho utilizada para factorizar n

Código 1: La siguiente salida en consola es el ejemplo de ejecucion del primer script llamado alfabeto.py desarrollamos este script y ejecutamos primero en todas las letras que estaban en minusculas pues es importante para ver el posible descifrado sustituyendo por todas las letras del alfabeto en este caso la letra a coincide con la llave publica que estamos buscando

```
python3 alfabeto.py
RSA Cifrado/ Descifrado

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
Iteraciones realizadas: 1 1
Iteraciones realizadas: 2 1
```

```

Iteraciones realizadas: 3 1
Iteraciones realizadas: 4 1
Iteraciones realizadas: 5 1
Iteraciones realizadas: 6 1
Iteraciones realizadas: 7 2789
p = 2789
q = 2791
f(x) = x^2 + 1
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2789)*(2791)
(n): 7784099

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2789-1)*(2791-1)
(phi): 7778520

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=47

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(47)^-1 mod (7778520)
(d): 1158503

Generando la llave publica y privada . . .
Llave publica: (47, 7784099)
Llave privada: (1158503, 7784099)

Cifrado de cada letra del alfabeto
a [1525454]
b [5820680]
c [6213204]
d [1702520]
e [613915]
f [3082985]
g [6128450]
h [3281868]
i [4660903]
j [3573068]
k [7663697]
l [2786183]
m [2541186]
n [3679525]

```

```
o [3925724]
p [6797061]
q [386440]
r [132921]
s [6848703]
t [1127235]
u [5756690]
v [2925404]
w [1535969]
x [4692267]
y [413079]
z [6256560]
```

Código 2: La siguiente salida en consola es de otro script llamado alfabetoVerificacion.py este escript es el utilizado para verificar todos los mensajes como ejemplo se puede notar como verifica que el mensaje anterior si corresponde a la letra a

```
RSA Cifrado/ Descifrado

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
Iteraciones realizadas: 1 1
Iteraciones realizadas: 2 1
Iteraciones realizadas: 3 1
Iteraciones realizadas: 4 1
Iteraciones realizadas: 5 1
Iteraciones realizadas: 6 1
Iteraciones realizadas: 7 2789
p = 2789
q = 2791
f(x) = x^2 + 1
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2789)*(2791)
(n): 7784099

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2789-1)*(2791-1)
(phi): 7778520

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=47

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
```

```
(d)=e^-1 mod phi
(d)=(47)^-1 mod (7778520)
(d): 1158503

Generando la llave publica y privada . . .
Llave publica: (47, 7784099)
Llave privada: (1158503, 7784099)

Ingrese el mensaje a encriptar: a
Mensaje a encriptar: a

Mensaje encriptado: 1525454

para desencriptar mensaje con llave privada (1158503,
7784099) . . .
Mensaje desencriptado: a

El cifrado y descifrado funciono correctamente!
```

En resumen con ayuda del primer script alfabeto.py se descifraron la mayoría de mensajes, en este caso se identifico que este script ayudaba a descifrar las letras minusculas que componen el mensaje completo, posteriormente con ayuda del script alfabetoVerificacio.py o alfabetoVerificacion.py se justificaron estas respuestas y tambien ayudo para encontrar la letra faltante la cual estaba en Mayuscula, un signo de espacio y el punto final

3 Problema 3

Mediante el algoritmo de la criba cuadrática descomponga $n = 4245221$ y descifre el mensaje en RSA que se proporciona mas adelante.

NOTA PARA EL AYUDANTE : Este problema se soluciono con 2 scripts realizados en python (ambos comentados), al final del ejercicio se muestra el flujo de ejecución de ambos, también se adjunta en la carpeta junto con este pdf los archivos para probar su funcionamiento

3.1 A

De las cotas de base e intervalo, escriba la base

Respuesta:

- Cotas de base e intervalo

1. $x = 2061$

2. $y = 50$

- Base

1. $b = 2111$

3.2 B

Proporcione las i de $q(i)$ con las cuales se obtiene la solución, x , y tales que $(x - y, n)$ donde n es un factor primo de n , describa de manera clara y metódica como obtiene y .

Respuesta: Para obtener el valor de las i de $q(i)$ debemos de obtener el resultado de la suma de $x + y$ y el resultado de la resta de $x - y$ en este caso seria $i = 2111$ $i = 2011$ respectivamente. Para obtener el valor de y primero definiremos 2 variables: (x, y) x sera la raíz cuadrada de el valor de n en este caso n corresponde a 4245221 y vamos a inicializar la variable y en 0 para que aumente su valor posteriormente. Utilizaremos un ciclo while el cual se ejecuta mientras $x^2 - y^2 \neq n$. Esto significa que nuestra variable x se incrementara hasta que encontremos un valor tal que $x^2 - y^2 == n$, en la segunda linea, el valor de x se incrementara en 1. Y por ultimo para obtener el valor de la variable y sera la raíz cuadrada de $x^2 - n$ Utilizaremos la función `math.sqrt()` para calcula la raíz cuadrada. Esta función devuelve un número de punto flotante. usamos el función `int()` para convertir este número en un entero y así tener el valor de $y = 50$

3.3 C

Descifre el siguiente mensaje cifrado en RSA:

Llave pública RSA = (4245221, 7), mensaje cifrado = 2787825

Para este caso se descifró con el archivo `alfabetoVerificacionCriba.py` para descryptar el signo de admiración ¡ directamente y se obtuvo el siguiente resultado

Respuesta: Signo de admiración que abre "¡" = 1550905

```
ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=7

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(7)^-1 mod (4241100)
(d): 1211743

Generando la llave publica y privada . . .
Llave publica: (7, 4245221)
Llave privada: (1211743, 4245221)
jajaj
Ingrese el mensaje a encriptar:
Mensaje a encriptar: !

Mensaje encriptado: 2787825

para descryptar mensaje con llave privada (1211743,
4245221) . . .
Mensaje descryptado: !

El cifrado y descifrado funciona correctamente!
```

Llave pública RSA = (4245221, 11), mensaje cifrado = 2055284

Respuesta: B = 2055284

Para este caso se descifró con el archivo `alfabetoVerificacionCriba.py` descifrando la letra B directamente y se obtuvo el siguiente resultado


```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=11

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(11)^-1 mod (4241100)
(d): 3469991

Generando la llave publica y privada . . .
Llave publica: (11, 4245221)
Llave privada: (3469991, 4245221)

Ingrese el mensaje a encriptar: B
Mensaje a encriptar: B

Mensaje encriptado: 2055284

para desencriptar mensaje con llave privada (3469991,
4245221) . . .
Mensaje desencriptado: B

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (4245221, 13), mensaje cifrado = 2061537
Respuesta: i = 2061537

Para este caso se descifró primero con el archivo alfabetoCriba.py para descifrar todas las letras minúsculas del alfabeto posteriormente se utilizó el script alfabetoVerificacionCriba.py pasando i directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=13

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(13)^-1 mod (4241100)
(d): 652477

Generando la llave publica y privada . . .
Llave publica: (13, 4245221)
Llave privada: (652477, 4245221)

Ingrese el mensaje a encriptar: i
Mensaje a encriptar: i

Mensaje encriptado: 2061537

para desencriptar mensaje con llave privada (652477,
4245221) . . .
Mensaje desencriptado: i

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (4245221, 17), mensaje cifrado = 4003203
Respuesta: e = 4003203

Para este caso se descifró primero con el archivo alfabetoCriba.py para descifrar todas las letras minúsculas del alfabeto posteriormente se utilizó script alfabetoVerificacionCriba.py pasando e directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)

```

```

CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=17

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(17)^-1 mod (4241100)
(d): 498953

Generando la llave publica y privada . . .
Llave publica: (17, 4245221)
Llave privada: (498953, 4245221)

Ingrese el mensaje a encriptar: e
Mensaje a encriptar: e

Mensaje encriptado: 4003203

para desencriptar mensaje con llave privada (498953,
4245221) . . .
Mensaje desencriptado: e

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (4245221, 19), mensaje cifrado = 3833015
Respuesta: n = 3833015

Para este caso se descifró primero con el archivo alfabetoCriba.py para descifrar todas las letras minúsculas del alfabeto posteriormente se utilizó script alfabetoVerificacionCriba.py pasando n directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)

```

```

(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=19

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(19)^-1 mod (4241100)
(d): 1116079

Generando la llave publica y privada . . .
Llave publica: (19, 4245221)
Llave privada: (1116079, 4245221)

Ingrese el mensaje a encriptar: n
Mensaje a encriptar: n

Mensaje encriptado: 3833015

para desencriptar mensaje con llave privada (1116079,
4245221) . . .
Mensaje desencriptado: n

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (4245221, 23), mensaje cifrado = 504464
Respuesta: Espacio " "

Para este caso se descifró primero con el script `alfabetoVerificacionCriba.py` pasando espacio " " directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)

```

```

(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=23

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(23)^-1 mod (4241100)
(d): 553187

Generando la llave publica y privada . . .
Llave publica: (23, 4245221)
Llave privada: (553187, 4245221)

Ingrese el mensaje a encriptar:
Mensaje a encriptar:

Mensaje encriptado: 504464

para desencriptar mensaje con llave privada (553187,
4245221) . . .
Mensaje desencriptado:

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (4245221, 29), mensaje cifrado = 1181333
Respuesta: d = 1181333

Para este caso se descifró primero con el archivo alfabetoCriba.py para descifrar todas las letras minúsculas del alfabeto posteriormente se utilizó script alfabetoVerificacionCriba.py pasando d directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)

```

```

(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=29

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(29)^-1 mod (4241100)
(d): 877469

Generando la llave publica y privada . . .
Llave publica: (29, 4245221)
Llave privada: (877469, 4245221)

Ingrese el mensaje a encriptar: d
Mensaje a encriptar: d

Mensaje encriptado: 1181333

para desencriptar mensaje con llave privada (877469,
4245221) . . .
Mensaje desencriptado: d

El cifrado y descifrado funciono correctamente!

```

Llave pública RSA = (4245221, 31), mensaje cifrado = 3063352
Respuesta: e = 3063352

Para este caso se descifró primero con el archivo alfabetoCriba.py para descifrar todas las letras minúsculas del alfabeto posteriormente se utilizó script alfabetoVerificacionCriba.py pasando e directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

```

```

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=31

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(31)^-1 mod (4241100)
(d): 3830671

Generando la llave publica y privada . . .
Llave publica: (31, 4245221)
Llave privada: (3830671, 4245221)

Ingrese el mensaje a encriptar: e
Mensaje a encriptar: e

Mensaje encriptado: 3063352

para desencriptar mensaje con llave privada (3830671,
4245221) . . .
Mensaje desencriptado: e

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (4245221, 37), mensaje cifrado = 1145481
Respuesta: s = 1145481

Para este caso se descifró primero con el archivo alfabetoCriba.py para descifrar todas las letras minúsculas del alfabeto posteriormente se utilizó script alfabetoVerificacionCriba.py pasando s directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1

```

```

(e)=37

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(37)^-1 mod (4241100)
(d): 343873

Generando la llave publica y privada . . .
Llave publica: (37, 4245221)
Llave privada: (343873, 4245221)

Ingrese el mensaje a encriptar: s
Mensaje a encriptar: s

Mensaje encriptado: 1145481

para desencriptar mensaje con llave privada (343873,
4245221) . . .
Mensaje desencriptado: s

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (4245221, 41), mensaje cifrado = 899155
Respuesta: $c = 899155$

Para este caso se descifró primero con el archivo alfabetoCriba.py para descifrar todas las letras minúsculas del alfabeto posteriormente se utilizó script alfabetoVerificacionCriba.py pasando c directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=41

```



```

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(41)^-1 mod (4241100)
(d): 2896361

Generando la llave publica y privada . . .
Llave publica: (41, 4245221)
Llave privada: (2896361, 4245221)

Ingrese el mensaje a encriptar: c
Mensaje a encriptar: c

Mensaje encriptado: 899155

para desencriptar mensaje con llave privada (2896361,
4245221) . . .
Mensaje desencriptado: c

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (4245221, 43), mensaje cifrado = 1046164
Respuesta: i = 1046164

Para este caso se descifró primero con el archivo alfabetoCriba.py para descifrar todas las letras minúsculas del alfabeto posteriormente se utilizó script alfabetoVerificacionCriba.py pasando i directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=43

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi

```

```

(d)=(43)^-1 mod (4241100)
(d): 2958907

Generando la llave publica y privada . . .
Llave publica: (43, 4245221)
Llave privada: (2958907, 4245221)

Ingrese el mensaje a encriptar: i
Mensaje a encriptar: i

Mensaje encriptado: 1046164

para desencriptar mensaje con llave privada (2958907,
4245221) . . .
Mensaje desencriptado: i

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (4245221, 47), mensaje cifrado = 1315170
Respuesta: f = 1315170

Para este caso se descifró primero con el archivo alfabetoCriba.py para descifrar todas las letras minúsculas del alfabeto posteriormente se utilizó script alfabetoVerificacionCriba.py pasando f directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=47

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(47)^-1 mod (4241100)
(d): 3699683

```

```

Generando la llave publica y privada . . .
Llave publica: (47, 4245221)
Llave privada: (3699683, 4245221)

Ingrese el mensaje a encriptar: f
Mensaje a encriptar: f

Mensaje encriptado: 1315170

para desencriptar mensaje con llave privada (3699683,
4245221) . . .
Mensaje desencriptado: f

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (4245221, 49), mensaje cifrado = 1878863
Respuesta: $r = 1878863$

Para este caso se descifró primero con el archivo alfabetoCriba.py para descifrar todas las letras minúsculas del alfabeto posteriormente se utilizó script alfabetoVerificacionCriba.py pasando r directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=49

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(49)^-1 mod (4241100)
(d): 1384849

Generando la llave publica y privada . . .

```

```

Llave publica: (49, 4245221)
Llave privada: (1384849, 4245221)

Ingrese el mensaje a encriptar: r
Mensaje a encriptar: r

Mensaje encriptado: 1878863

para desencriptar mensaje con llave privada (1384849,
4245221) . . .
Mensaje desencriptado: r

El cifrado y descifrado funciona correctamente!

```

Llave pública RSA = (4245221, 53), mensaje cifrado = 2088416
Respuesta: $a = 2088416$

Para este caso se descifró primero con el archivo alfabetoCriba.py para descifrar todas las letras minúsculas del alfabeto posteriormente se utilizó script alfabetoVerificacionCriba.py pasando a directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=53

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(53)^-1 mod (4241100)
(d): 3921017

Generando la llave publica y privada . . .
Llave publica: (53, 4245221)
Llave privada: (3921017, 4245221)

```

```
Ingrese el mensaje a encriptar: a
Mensaje a encriptar: a

Mensaje encriptado: 2088416

para desencriptar mensaje con llave privada (3921017,
4245221) . . .
Mensaje desencriptado: a

El cifrado y descifrado funciona correctamente!
```

Llave pública RSA = (4245221, 59), mensaje cifrado = 2571920
Respuesta: d = 2571920

Para este caso se descifró primero con el archivo alfabetoCriba.py para descifrar todas las letras minúsculas del alfabeto posteriormente se utilizó script alfabetoVerificacionCriba.py pasando d directamente y se obtuvo el siguiente resultado

```
ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=59

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(59)^-1 mod (4241100)
(d): 2803439

Generando la llave publica y privada . . .
Llave publica: (59, 4245221)
Llave privada: (2803439, 4245221)

Ingrese el mensaje a encriptar: d
```

Mensaje a encriptar: d

Mensaje encriptado: 2571920

para desencriptar mensaje con llave privada (2803439,
4245221) . . .

Mensaje desencriptado: d

El cifrado y descifrado funciono correctamente!

Llave pública RSA = (4245221, 61), mensaje cifrado = 2621019

Respuesta: o = 2621019

Para este caso se descifró primero con el archivo alfabetoCriba.py para descifrar todas las letras minúsculas del alfabeto posteriormente se utilizó script alfabetoVerificacionCriba.py pasando o directamente y se obtuvo el siguiente resultado

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)

(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=61

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e⁻¹ mod phi
(d)=(61)⁻¹ mod (4241100)
(d): 903841

Generando la llave publica y privada . . .
Llave publica: (61, 4245221)
Llave privada: (903841, 4245221)

Ingrese el mensaje a encriptar: o
Mensaje a encriptar: o

```

Mensaje encriptado: 2621019

para desencriptar mensaje con llave privada (903841,
4245221) . . .
Mensaje desencriptado: o

El cifrado y descifrado funciona correctamente!

Llave pública RSA = (4245221, 71), mensaje cifrado = 1550905
Respuesta: Signo de admiración que cierra "!" = 1550905

```

Para este caso se descifró primero con el archivo alfabetoCriba.py para descifrar todas las letras minúsculas del alfabeto posteriormente se utilizó script alfabetoVerificacionCriba.py pasando el signo de admiración que cierra ! directamente y se obtuvo el siguiente resultado

```

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=71

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(71)^-1 mod (4241100)
(d): 3942431

Generando la llave publica y privada . . .
Llave publica: (71, 4245221)
Llave privada: (3942431, 4245221)

Ingrese el mensaje a encriptar: !
Mensaje a encriptar: !

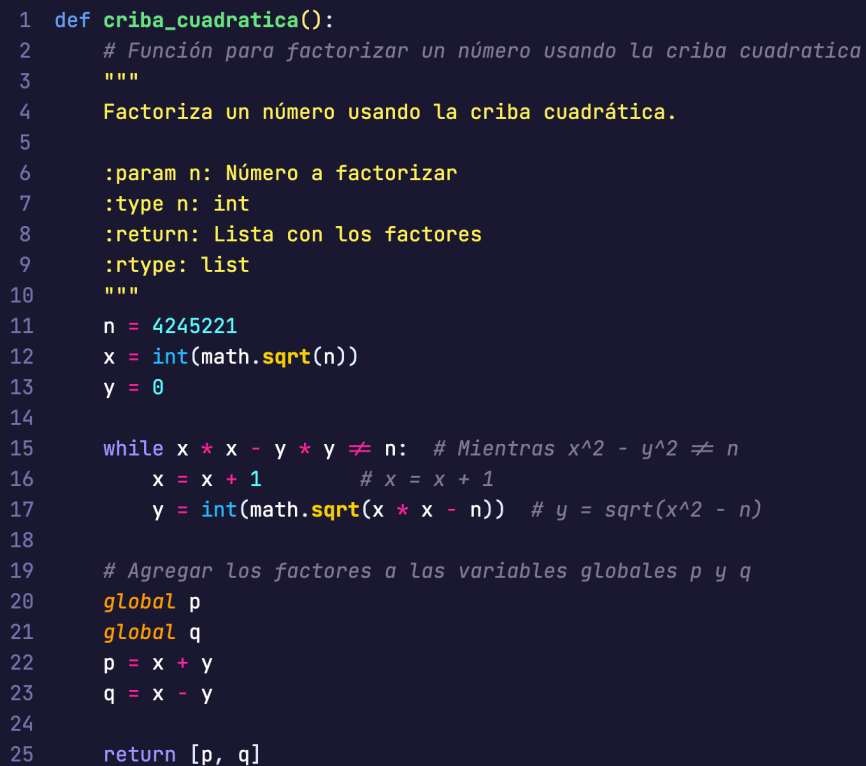
Mensaje encriptado: 1550905

```

```
para desencriptar mensaje con llave privada (3942431,
4245221) . . .
Mensaje desencriptado: !
El cifrado y descifrado funciono correctamente!
```

Respuesta completa: ¡Bien descifrado!

3.4 Flujo de ejecución de los scripts con función Criba Cuadrática



```
1 def criba_cuadratica():
2     # Función para factorizar un número usando la criba cuadratica
3     """
4     Factoriza un número usando la criba cuadrática.
5
6     :param n: Número a factorizar
7     :type n: int
8     :return: Lista con los factores
9     :rtype: list
10    """
11    n = 4245221
12    x = int(math.sqrt(n))
13    y = 0
14
15    while x * x - y * y != n: # Mientras  $x^2 - y^2 \neq n$ 
16        x = x + 1             #  $x = x + 1$ 
17        y = int(math.sqrt(x * x - n)) #  $y = \text{sqrt}(x^2 - n)$ 
18
19    # Agregar los factores a las variables globales p y q
20    global p
21    global q
22    p = x + y
23    q = x - y
24
25    return [p, q]
```

Figure 4: Función criba cuadrática utilizada para factorizar n

Código 3: La siguiente salida en consola es de un script llamado alfabetoCriba.py desarrollamos este script como el primero y ejecutamos pues es importante para ver el posible descifrado de las letras minúsculas del texto que debemos descifrar en este ejemplo desciframos a la letra o

```
python3 alfabetoCriba.py
RSA Cifrado/ Descifrado

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
Cotas de base e intervalo
x = 2061
y = 50
Proporcione las i de q(i)
i = 2111
i = 2011

Base
b = 2111
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=61

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(61)^-1 mod (4241100)
(d): 903841

Generando la llave publica y privada . . .
Llave publica: (61, 4245221)
Llave privada: (903841, 4245221)

Cifrado de cada letra del alfabeto
a [3992719]
b [2726503]
c [1580525]
```

```

d [1651182]
e [3076022]
f [978043]
g [3187499]
h [3204417]
i [3079087]
j [2362530]
k [2371550]
l [662728]
m [3276174]
n [3394893]
o [2621019]
p [508311]
q [693599]
r [1939000]
s [3457681]
t [3161686]
u [1383688]
v [3767530]
w [4217673]
x [3572679]
y [3371282]
z [2734357]

```

Código 4: La siguiente salida en consola es de nuestro ultimo script llamado alfabetoVerificacionCriba.py el cual nos ayudara a verificar si se encontro correctamente como se menciono anteriormente aunque este script se utilizo mas para descifrar las letras que no son minusculas y los signos de admiracion

```

python3 alfabetoVerificacionCriba.py
RSA Cifrado/ Descifrado

ELEGIMOS VALORES DE NUMEROS PRIMOS PARA (p) y (q)
CALCULAMOS EL VALOR DE (n)
(n)=(p)*(q)
(n)=(2111)*(2011)
(n): 4245221

CALCULAMOS EL VALOR DE (phi)
(phi)=(p-1)*(q-1)
(phi)=(2111-1)*(2011-1)
(phi): 4241100

ELEGIMOS UN VALOR DE (e) PARA LA LLAVE PUBLICA
(e)/ 1<e<phi and mcd(e,phi)==1
(e)=61

```

```

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA
(d)=e^-1 mod phi
(d)=(61)^-1 mod (4241100)
(d): 903841

Generando la llave publica y privada . . .
Llave publica: (61, 4245221)
Llave privada: (903841, 4245221)

Ingrese el mensaje a encriptar: o
Mensaje a encriptar: o

Mensaje encriptado: 2621019

para desencriptar mensaje con llave privada (903841,
4245221) . . .
Mensaje desencriptado: o

El cifrado y descifrado funciona correctamente!

```

3.5 D

verifique si la firma digital RSA f firma = 1107437 del mensaje $m = 1550905$ con parámetros (4245221, 7) es valida.

Respuesta: De acuerdo al valor de (n) que es 4245221 y nuestro valor de (e) que es 7 Debemos descifrar que la firma digital RSA 1107437 es valida con el mensaje 1550905. Primero debemos de obtener los valores correspondientes a (p, q) los cuales con ayuda de nuestro script `alfabeto.py` o `alfabetoCriba.py` podemos encontrar, estos valores son.

VALORES DE (p, q)
 $p = 2789$ $q = 2791$

Para corroborar esto CALCULAMOS EL VALOR DE (n)
 $(n) = (p) * (q)$
 $(n) = (2789) * (2791)$
 $(n) : 7784099$

CALCULAMOS EL VALOR DE (ϕ)
 $(\phi) = (p - 1) * (q - 1)$
 $(\phi) = (2789 - 1) * (2791 - 1)$
 $(\phi) : 7778520$

CALCULAMOS EL VALOR DE (d) PARA LA LLAVE PRIVADA

$$\begin{aligned}(d) &= e^{-1} \bmod \phi \\(d) &= (7)^{-1} \bmod (7778520) \\(d) &: 6667303\end{aligned}$$

GENERANDO LA LLAVE PUBLICA Y PRIVADA . . .

Llave pública: (7, 7784099)

Llave privada: (6667303, 7784099)

Posteriormente debemos de sustituir los valores encontrados en la siguiente formula para resolver el problema

Dada la función

$$y = Sig_k(x) = x^d \bmod(n) \quad (1)$$

$$Ver_k(x, y) \iff x \equiv y^e \bmod n \quad (2)$$

sustituyendo los valores tenemos algo como

$$Sig_k(x) = Sig_k(1550905) \quad (3)$$

$$= m^{6667303} \bmod n = 1107437 \quad (4)$$

$$x \equiv y^7 \bmod n \quad (5)$$

$$1550905 = 1107437 \bmod(4245221) \quad (6)$$

$$1550905 \equiv 1550905 \bmod 4245221 \quad (7)$$

\therefore La firma es valida

4 Problema 4

El siguiente mensaje fue cifrado con el algoritmo de ElGamal con llave pública $= (2011, 17, 19)$, mediante el algoritmo de cálculo de índices con la base $B = 2, 3, 5, 7, 11$ encuentre el índice de 19 base 17 módulo 2011.

4.1 A

De las ecuaciones ya solucionadas para cada índice

$$\begin{aligned} 17^{1165} \bmod 2010 &= 2 \longrightarrow 1165 = \log_{17}(2) \bmod 2010 \\ 17^{1703} \bmod 2011 &= 3 \longrightarrow 1703 = \log_{17}(3) \bmod 2010 \\ 17^{1202} \bmod 2011 &= 5 \longrightarrow 1202 = \log_{17}(5) \bmod 2010 \\ 17^{11} \bmod 2011 &= 7 \longrightarrow 11 = \log_{17}(7) \bmod 2010 \\ 17^{1231} \bmod 2011 &= 11 \longrightarrow 1231 = \log_{17}(11) \bmod 2010 \end{aligned}$$

4.2 B

De la iteración en la cual se obtiene el índice de 19 base 17 módulo 2011.

Elegimos a k aleatoriamente, entonces $k = 503$

$$\beta * \alpha^k \bmod 2011 = 19 * 17^{503} \bmod 2011 = 77$$

Y 77 se descompone en: $11 * 7$ y $(7, 11) \in S$

$$\begin{aligned} \implies \log_{17}(19) &= (\log_{17}(7) + \log_{17}(11) - 503) \bmod 2010 = 739 \\ \therefore y &= 739 = \log_{17}(19) \end{aligned}$$

4.3 C

Descifre el mensaje:

$(891, 260), (1070, 1838), (91, 934), (1547, 1835), (156, 761), (641, 1542), (842, 1820), (237, 1757), (7, 1215), (119, 1898)$

4.4 D

Verifique la siguiente firma digital Gammal $\text{sk}(33, 7) = (\gamma = 156, \delta = 477)$, con llave pública $= (2011, 17, 19)$ ¿Es valida la firma?