

Programa 1.

Pseudocódigo:

Inicio

Incluir funciones math

POSIBLES_OPCIONES_NUM=['numerica','Numerica','n','N','num','NUMERICA','NUM']

POSIBLES_OPCIONES_CIS=['cis','Cis','CIS','c','C']

LST=["Introducir parte real: ","Introducir parte imaginaria: ","Introducir modulo: ","Introducir angulo: "]

clase Numero:

método inicializador (self, parte_real, parte_imaginaria):

self.parte_real=parte_real

self.parte_imaginaria=parte_imaginaria

método string (self):

regresar str(self.parte_real)+" "+str(self.parte_imaginaria)+"i"

método mod(self):

self.m=math.sqrt(math.pow(self.parte_real,2)+math.pow(self.parte_imaginaria,2))

regresar self.m

método ang(self):

self.angle=math.degrees(math.atan2(self.parte_imaginaria,self.parte_real))

regresar self.angle

método num_a_cis(self):

regresar str(self.m)+"cis"+str(self.angle)

método suma(self, a, b):

self.parte_real+=a

self.parte_imaginaria+=b

método resta (self, a, b):

self.parte_real-=a

self.parte_imaginaria-=b

método multiplicacion(self, a, b):

c=self.parte_real

self.parte_real=self.parte_real*a - self.parte_imaginaria*b

self.parte_imaginaria=self.parte_imaginaria*a + c*b

clase Cis:

método inicializador (self, mod, theta):

self.mod=mod

self.theta=theta

while self.theta<0:

self.theta+=360

while self.theta>360:

self.theta-=360

método string (self):

```
return str(self.mod)+"cis"+str(self.theta)
```

método parte_r(self):

```
self.r=self.mod*math.cos(math.radians(self.theta))
```

```
regresar self.r
```

método parte_i(self):

```
self.i=self.mod*math.sin(math.radians(self.theta))
```

```
regresar self.i
```

método cis_a_num(self):

```
self.r=self.mod*math.cos(math.radians(self.theta))
```

```
self.i=self.mod*math.sin(math.radians(self.theta))
```

```
regresas str(self.r)+" "+str(self.i)+"i"
```

método division(self, modulo, angulo):

```
self.mod=self.mod/modulo
```

```
self.theta-=angulo
```

```
mientras self.theta<0:
```

```
    self.theta+=360
```

método potencia(self,exponente):

```
self.mod=math.pow(self.mod,exponente)
```

```
self.theta*=exponente
```

```
mientras self.theta>360:
```

```
    self.theta-=360
```

Funcion numero_nuevo(forma):

a=0

num=[]

Si forma en POSIBLES_OPCIONES_NUM:

para i en rango(2):

a=Dato_de_entrada(LST[i])

a=float(a)

num.agregar(a)

regresar Numero(num[0],num[1])

Sino Si forma en POSIBLES_OPCIONES_CIS:

Para i en rango(2,4):

a=Dato_de_entrada(LST[i])

a=float(a)

num.agregar(a)

regresar Cis(num[0],num[1])

num=Ninguno

b=0

lst=[]

Mientras:

forma=Dato_de_entrada("Introducir forma: numerica o cis (para salir (s)): ")

Si forma igual a "s":

salir mientras

num=numero_nuevo(forma)

imprimir num

Si num es igual a Ninguno:

imprimir "Opcion no valida"

continuar con la siguiente iteracion

Mientras:

operación=Dato_de_entrada ("Introducir lo que se desea hacer: suma(+), resta(-), división(/), multiplicación(*), potenciación(^), modulo(r), conversión(c), salir(s): ")

Si operación es igual a "s":

Fin mientras

si forma en POSIBLES OPCIONES_NUM:

si operación diferente de "^" y operación diferente de "r" y operación diferente de "c":

Para i en rango(2):

b=dato_de_entrada(LST[i])

b=int(b)

lst.agregar(b)

Si operacion igual a "+":

num.suma(lst[0],lst[1])

Sino si operacion igual a "-":

num.resta(lst[0],lst[1])

sino si operacion igual a "/":

num=Cis(num.mod(), num.ang())

num2=Numero(lst[0],lst[1])

num.division(num2.mod(),num2.ang())

num=Numero(num.parte_r(), num.parte_i())

sino si operacion igual a "*":

num.multiplicacion(lst[0],lst[1])

sino:

imprimir "Elegir una operacion valida"

continuar a la siguiente interacion

sino:

si operacion igual a "^":

num=Cis(num.mod(),num.ang())

b=raw_input("Introducir exponente:")

b=int(b)

num.potencia(b)

num=Numero(num.parte_r(), num.parte_i())

elif operacion igual a "r":

print num.mod()

elif operacion igual a "c":

num=Cis(num.mod(), num.ang())

forma= "cis"

Sino si forma in POSIBLES_OPCIONES_CIS:

Si operacion igual "^" y operacion igual "r" y operacion igual "c":

para i en rango(2,4):

b=(LST[i])

b=int(b)

lst.agregar(b)

if operacion == "+" or operacion == "-":

num=Numero(num.parte_r(),num.parte_i())

num2=Cis(lst[0],lst[1])

lst=[num2.parte_r(),num2.parte_i()]

```

    if operacion == "+":
        num.suma(lst[0],lst[1])
    else:
        num.resta(lst[0],lst[1])
        num=Cis(num.mod(),num.ang())
elif operacion == "/":
    num.division(lst[0],lst[1])
elif operacion == "*":
    num=Numero(num.parte_r(),num.parte_i())
    num2=Cis(lst[0],lst[1])
    num.multiplicacion(num2.parte_r(),num2.parte_i())
    num=Cis(num.mod(),num.ang())
Si no:
    Si operacion == "^":
        b=Dato_de_entrada("Introducir exponente:")
        b=int(b)
        num.potencia(b)
    sino si operacion == "r":
        num=Numero(num.parte_r(),num.parte_i())
        imprimir num.mod()
        num=Cis(num.mod(),num.ang())
    si no si operacion == "c":
        num=Numero(num.parte_r(), num.parte_i())
        forma= "numerica"

imprimir num
lst=[]

```

Fin