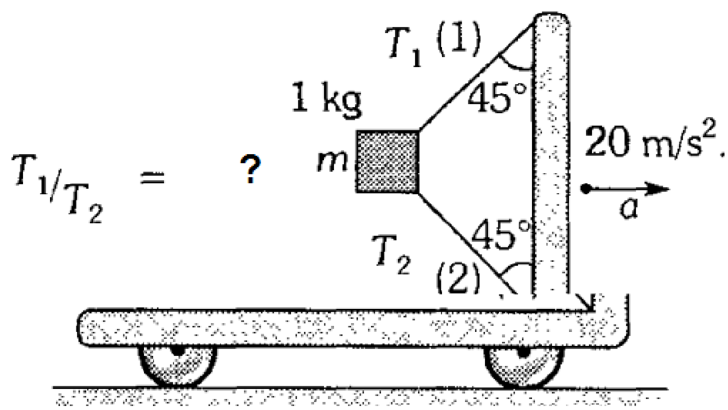
```
from sympy import *
from polyclass import *
from libaldo_math import *
from libaldo_show import *
from physic_lib import *
from IPython.display import display, Math
init_printing()
```

Basic Dynamic problems for understand how work the library

El coche se mueve hacia la derecha con aceleración de 20 m/s^2. El bloque de 1 kg no se mueve respecto de él y mantiene tensas las cuerdas. La relación de tensiones en la cuerda (1) y (2) es: (g=10 m/s^2).

car move right whit acc=20, m=1 kg and dont move whit car ref find T1/T2




```
T1,T2=symbols('T1 T2',positive=True)
P=mparticle(m=1,g=10)
P.add_forza(T1,pi/4)
P.add_forza(T2,-pi/4)
P.add_forza(10,-pi/2)
```

```
T=csolve(P.y_res(),T1,'T1')
```

$$T1 = T_2 + 10\sqrt{2}$$

```
P.store_val(T1,T) # remplaze the value of T1 for T in P object
```

```
T=csolve(P.simple_ac()-20,T2,'T2',unifique=True,kpositive=True)
```

$$T2 = 5\sqrt{2}$$

```
P.store_val(T2,T)
```

```
P.value(T1) # value get answer after find some results
```

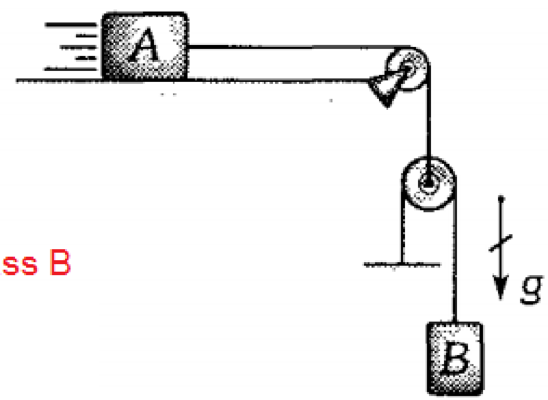
$$15\sqrt{2}$$

same problem using polyclass and Mpolyclass

```
T1,T2=symbols('T1 T2',positive=True)
P=mparticle(m=1,g=10)
P.add_forza(T1,pi/4)
P.add_forza(T2,-pi/4)
P.add_forza(10,-pi/2)
e1=polyclass(P.y_res(),0)
e2=polyclass(P.simple_ac(),20)
A=Mpolyclass([e1,e2])
A.csolves([T1,T2])
```

$$\left[\left(15\sqrt{2},\ 5\sqrt{2}\right)\right]$$

. En la figura mostrada, despreciando la masa de las poleas y la fricción, calcule el módulo de la aceleración de A, sabiendo que su masa es igual a la del bloque B. ($g = 10\ \mathrm{m/s^2}$)



the system is frictionless, pulley mass is 0, mass A = mass B
find acceleration

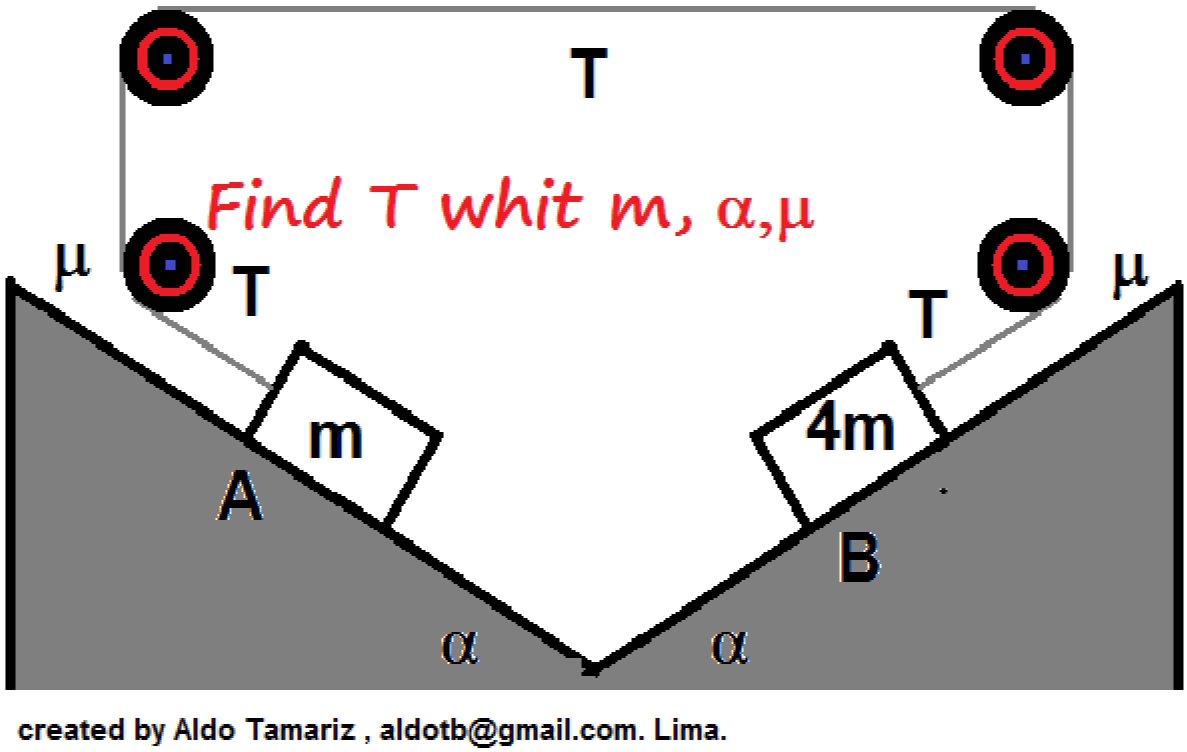
```
T,A_c,g,m=symbols('T A_c g m',positive=True)
```

```
T=m*A_c;T
```

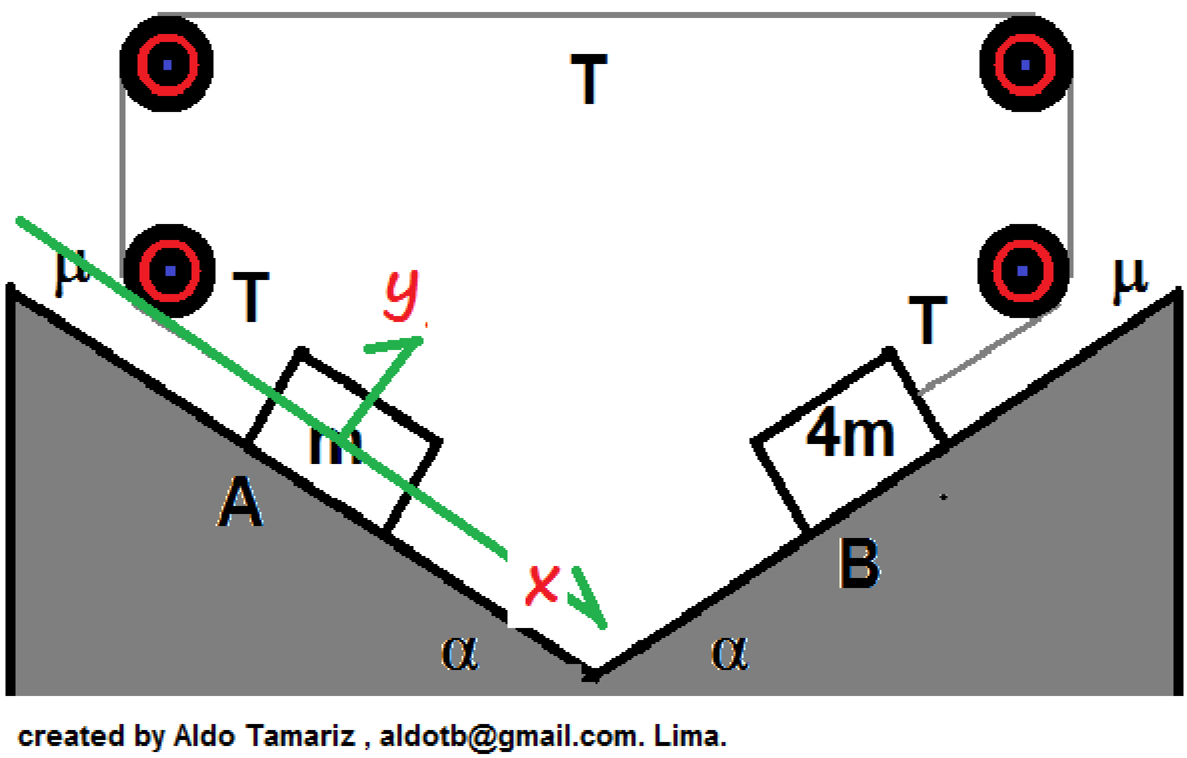
$$A_c m$$

```
A_c=csolve((m*10-T/2)-m*A_c/2,A_c,'Ac')
```

$$Ac = 10$$




```
T,g,m,alpha,mu,N1,N2,f_1,f_2=symbols('T g m alpha mu N1 N2 f_1 f_2',positive=True)
```




```
# wiltake axxis reference whit  mass move direcction refernce like x
A=mparticle(m=m,g=g)
A.add_forza(m*g,-pi/2+alpha)
A.add_forza(T,pi)
A.add_forza(N1,pi/2)
A.add_forza(f_1,0)
```

```
A.y_res()
```

$$N_1 - gm\cos(\alpha)$$

```
A.store_val(N1,g*m*cos(alpha))
A.store_val(f_1,N1*mu)
```

```
A.simple_ac()
```

$$\frac{-T + gm\mu\cos(\alpha) + gm\sin(\alpha)}{m}$$

```
# wiltake axxis reference whit  mass move direcction refernce like x
B=mparticle(m=4*m,g=g)
B.add_forza(4*m*g,-pi/2-alpha)
B.add_forza(T,0)
B.add_forza(N2,pi/2)
B.add_forza(f_2,pi)
```

```
B.y_res('x')
```

$$N_2 - 4gm\cos(\alpha)$$

```
B.store_val(N2,4*g*m*cos(alpha))
B.store_val(f_2,N2*mu)
```

```
B.simple_ac()
```

$$\frac{T - 4gm\mu\cos(\alpha) - 4gm\sin(\alpha)}{4m}$$

```
T=csolve(A.simple_ac()-B.simple_ac(),T,'T',unifique=True)
```

$$T = \frac{8gm(\mu\cos(\alpha) + \sin(\alpha))}{5}$$

```
T,g,m,alpha,mu,N1,N2,f_1,f_2=symbols('T g m alpha mu N1 N2 f_1 f_2',positive=True)
```

```
# Creating object and adding forcces...
A=mparticle(m=m,g=g)
A.add_forza(m*g,-pi/2+alpha)
A.add_forza(T,pi)
A.add_forza(N1,pi/2)
A.add_forza(f_1,0)
B=mparticle(m=4*m,g=g)
B.add_forza(4*m*g,-pi/2-alpha)
B.add_forza(T,0)
B.add_forza(N2,pi/2)
```

```
B.add_forza(f_2,pi)

# Creating know Eq and after add to Mpolyclass s...
e1=polyclass(N1,g*m*cos(alpha))
e2=polyclass(f_1,N1*mu)

e3=polyclass(N2,4*g*m*cos(alpha))
e4=polyclass(f_2,N2*mu)

e5=polyclass(B.simple_ac(),A.simple_ac())



# creatibg super class
vec=[e1,e2,e3,e4,e5]
P=Mpolyclass(vec)
P.s()
```

$$\begin{aligned} N_1 &= gm\cos(\alpha) \\ f_1 &= N_1\mu \\ N_2 &= 4gm\cos(\alpha) \\ f_2 &= N_2\mu \\ \frac{T - f_2 - 4gm\sin(\alpha)}{4m} &= \frac{-T + f_1 + gm\sin(\alpha)}{m} \end{aligned}$$

```
P.csolves([N1,N2,f_1,f_2,T]) # solve all
```

$$\left\{N_1 : gm\cos(\alpha),\ N_2 : 4gm\cos(\alpha),\ T : \frac{8gm\mu\cos(\alpha)}{5} + \frac{8gm\sin(\alpha)}{5},\ f_1 : gm\mu\cos(\alpha),\ f_2 : 4gm\mu\cos(\alpha$$

✓ 0 s completado a las 21:28