

```

elif j[1] < 0:
    #para no dejar la carga util en menos de cero si hubieramos
    #llegado al límite, volvemos atrás de este modo.
    #así además cumplimos el requisito de llenar cohetes hasta
    #que tengan la carga util menor a cualquier petición pendiente
    #y nunca se asignará carga a dicho cohete
    j[1] = j[1] + i[2]

else:
    #comprobamos que la peticion no haya sido asignada previamente
    for g in peticionAsignada:

        if g[0] == i[0]:

            asignar = False

        else:

            asignar = True

    while asignar:
        print("La petición ", i[0], "(", i[2], " Kg de ", i[1], ")")
        print("ha sido asignada al cohete ", k[0], "...Duración lanzamiento: ", k[1], "días.")

        print("El cohete ", k[0], " aún dispone de ", j[1], "kg de carga útil.\n")

        peticionAsignada.append([i[0],k[1],j[0], 0])

        cambioEstado = peticiones.index(i)
        peticiones[cambioEstado][4]=1 #Actualizamos el estado de la peticion a "Asignada"

        cohetesLlenos.append(j[0])

        break

print("MISIONES IMPOSIBLES DE CUMPLIR:") #cualquier peticiones que siga en la lista de peticiones significa que no
print(chr(734)*35)
for z in peticiones:

    if z[4]!=1:

        print("La petición", z[0], " es imposible de cumplir")
        print(chr(734)*15)

print("PETICIONES YA ASIGNADAS PERO NO ENVIADAS")
print(chr(734)*38)

for r in peticionAsignada:

    if r[3]==0:
        print("Mision: ", r[0], " Duración: ", r[1], " Cohete asignado: ", r[2])
        print(chr(734)*15)

```