```
### Python Programming for Software Engineers
### Assignment 7
### 'Lambda De Parser'


# [Firstname Lastname]

# Task 1
# -----------------------------------------------
# Given the following:
f = lambda x, y: x * y

# 1. Rewrite to its logical equivalence using ordinary funcion definition(s)
# [code]


# Task 2
# -----------------------------------------------
# Given the following:
f = lambda x: (lambda y: (lambda z: x + y + z))

# 1. How would you call it to get the result of `x + y + z`?
# [code]

# 2. Rewrite it using only one lambda expression and show how to call it
# [code]


# Task 3
# -----------------------------------------------
# Given the following:
(lambda b = (lambda *c: print(c)): b("a", "b"))()

# 1. What happens here? Rewrite it so that the code can be
# understood by a normal or your mate who has no idea what the lambda is!
# Provide comments, neat formatting and a bit more meaningful var names.
# [multiline code interlaced with comments]

# Task 4 (soft)
# -----------------------------------------------
# What are the main restrictions on the lambda?
# Provide "If yes, why? If not, why not?" for each of the following:
# 1. Does lambda restrict side effects?
# 2. Does lambda restrict number of allowed statements?
# 3. Does lambda restrict assignments?
# 4. Does lambda restrict number of return values?
# 5. Does lambda restrict the use of default arguments values?
# 6. Does lambda restrict possible function signatures?

# [your enumerated answers; if possible, code is welcomed]


# Task 5
# -----------------------------------------------
# Given the following:
(lambda f = (lambda a: (lambda b: print(list(map(lambda x: x+x, a+b))))):
f((1,2,3))((4,5,6)))()

# 1. What happens here? Do the same as in Task 3 and
# enumerate order of execution using (1,2,3...) in comments
# [multiline code interlaced with comments]

# 2. Why does map() requires list() call?
# [written answer]
```