1.

in this question we need to use ./segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s 0 -c command to see the results of segmentation.

we have 128 address space size which means we need 7 bits to show this space.

we have 2 segments. this means the leftmost bit is used for choosing segment and other 6 bits are offset which is diffrent for segment 1 and 2.

for example we have address = 108 which is 1101100 in binary. leftmost bit is 1. so this address is pointing to segment 1. offset is 44. but this isn't correct.

we need to sub 44 of 64 to get the correct offset. because offset is equal o limit of space; this address is valid in segment 1.and mapped to address 512-20 = 492 which is the last location of this address.

but for example address = 97 is 1100001 in binary. like above this address is pointing to segment 1 too. offset of this address is 33 which main offset must be 33-64 = -31. this offset is bigger than limit of space. so segmentation violation occcurs.

3.

in this case we run ./segmentation.py -a 16 -p 128 -A 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 -c command.

is this question we need set base and bounds and limits of spaces of segments.

we have two segments and we want to see segment violation in index=3 to index=13.

so in segment 0 we must set base = 0 and limit = 2 to have valid address in index = 0,1

and we must set base = 15 and limit = 2 for segment 1 to have valid address for index = 14,15

2.

we have base = 0 and limit = 20 for segment 0. so this segment is mapped to address 0-20 in physical address

we have base = 0 and limit = 20 for segment 1 too. so this segment is mapped to address 512 to 512-20 = 492 in physical address

4.

if we have 16 byte in virtual address and we want to have 90% of valid addresses , we must have about 14 byte valid addresses and 2 byte segmentation violations.

so in segment 0 we must set base = 0 and limit = 7 and in segment 1 we must set base = 15 and limit = 7.

so we have about 10% segmentation violations

5.

yes we can.

if we set limit = 0, then we have no valid virtual addresses.