1.yes we can

in this case we run ./malloc.py -n 10 -H 0 -p BEST -s 0 -c to see the result.

as we know malloc retrun a pointer to first of space that specify for command.

so alloc(3) return 1000 that is the start value of space

and free list updated and address of free list is incresed by 3 and starts from 1003 and size of free space is updated by 100-3 = 97.

as we see, in next line we have free command that is remove pointer from pointing space

so free space is updated but this time takes to two 3 and 97 size segments.

if we run another alloc next time, because we have best policy, we allocate a space that remind lager free space for other programs.

so we have allocate 5 bite from 97 bit space.

2.

for this case we must run ./malloc.py -n 10 -H 0 -p WORST -s 0 -c command

from starting command to ptr[3] we see same results, but from freelist in next line we see

size 4 to worst policy which is bigger than best policy.

because in best policy when we write alloc(8) in this situation, free space with size 8 chooses.

but in worst policy alloc(8) sub space from size 84.

3.

we run ./malloc.py -n 10 -H 0 -p FIRST -s 0 -c command to see result

diffrene between first and worst or best policy is that the first space that fit the address is allocate to it.

in this situation the best and first policy work same.

but with worst case in alloc(2) in final differs

4.

in we have list orders, we can handle more than one command at the same time,
as we choose sizesort for example, the more important command will choose

in this case because we have one commadn at the same time notihng will change

5.

if we use coalesing ,the free space can converge and make bigger free space.
so in this example in we have 3,5,8,84 in free list and allocate(8) with usuing coales
we allocate 3+5 to this address

6.

if we increase this percant we can break more space in address space to allocate to programs
so our outcome will be more efficient.