LS 1 A

EXERCISE 1:

1 A) FIRST COMPILE using >> gcc -g sample.c -o sample.
Then run using >> gdb sample

2 A) after normal execution of programme gdb will run with 'run' command at the time,
we can give command line arguments next to run.
Eg:- (gdb) run 3 //where 3 is command line argument.

3 A) two ways
1.break line-number // setting break point particular line.
2.break function-name //setting break point function.

4 A)>>break expression //expression cold be our condition.

5 A)by entering 'c' in gdb prompt.

6 A)by entering 's' in the gdb prompt.

7 A) by entering 'c' in gdbprompt.

8 A)by typing 'print variable name' or 'print expression' in gdb command prompt.

9 A)

10 A)command 'info locals'.

11 A)command 'quit' in gdb prompt.

EXERCISE 2:

observation::

the value of 's2' is correct but 's1' isnot getting freed, meaning buffer of 's1' is not getting cleared. This is the reason for wrong result.

Change:: just add the '/0' at the end of s1.

Reason:: c generally includes this delimiter at the endof any string.

Added line:: s1[s1len+s2len]=0; after the for loop in the function.

EXERCISE 3:

OBSERVATION:

the segmentation fault is because of absene of '&' in 'scanf' function.
But the output is '0' because

reason:: the veriable 'sum' is expired outside the function 'read__values'.
'sum' is loccal to the function only.

Change:: make the veriable 'sum' global so that it can be used all through the programme.

By setting the break point at while loop in function we will come to know the segmentation fault.

LS1 B:

observation:

PATR A:
THERE is no bug at all

actually memory leak (definitely lost)is presentbut its given in the begining of the programme snippet
the function returns non-null pointer to 'dest'.

PART B:

PATR C: NO bug.

LS1 C:

OBSERVATION:

debug-test.c

the line 'printf("%c", (*message)+i);' is creating the problem.

Atthe end of the whileloop (after it runs till last letter of message) the printf command is trying to print the values pointed by wrong address . That address may belong to other processes .