Om sai ram

the answers are

for LS1A :

Exerise 1:

How do you run a program in gdb?
ANS : once you give the command gcc -g <executable> <filename>.c , we just give the command run at the prompt to make the program run in gdb as:

```
gdb <executable> or gdb <filename>
```

How do you pass command line arguments to a program when using gdb?
ANS : run [arglist] at the gdb prompt.

How do you set a breakpoint in a program?
ANS : b [file:]function -> sets breakpoint at function [in file]

```
break [file:]line   set breakpoint at line number [in  file]
 b [file:]line
```

How do you set a breakpoint which only occurs when a set of conditions is true?
Ans :break . . . if expr

How do you execute the next line of C code in the program after a break?
ANS : next command

If the next line is a function call, you'll execute the call in one step. How do you execute the C code, line by line, inside the function call?
ANS :s command

How do you continue running the program after breaking?
ANS : continue command

How can you see the value of a variable (or even an expression) in gdb?
ANS : print variable (or even an expression) command will give the value.

How do you configure gdb so it prints the value of a variable after every step?
ANS : display all

How do you exit out of gdb?
ANS : quit.

Exercise2:

using gdb to find the bugs in appendtest.c , we get some junk values to be appende to our result string this is solved through adding a line
str1[strlrn(s1)+strlrn(s2)] = '\0' which makes the rest of the characters after the resulted appending to be null.

Exercise 3:

In this the first use of gdb for the segmentation fault gives us the missing & at the scanf statement in the while loop.

For the correct inputs and the wrong answer case the bug was that the variable 'sum' is local and initialized to “0” in the main but another local variable sum in he average function is updated . So make the sum variable to be global variable and then we get the correct answers after the compilation!!!.

for LS1B :

All the three parts have no bugs .

for LS1 C :

The debug-test.c : the bugs were : NULL is not getting treated as string and the stmt printf("%c", (*message)+i); prints junk characters after the value in the gdb so remove +i ;

The valgrind-test.c : the different types of errors were noticed by the usage of the valgrind!!!.