CMSC 257 Lab 4
Debugging with GDB

Summary

- Create a new directory: “lab4”. All the files for this lab should be saved in this directory only.
- Copy all the files from Lab-3 into this directory.
- We will use the same set-up as Lab-3. Here we want to include two new functions:
  (i) `sumall(int a[], int size)` and
  (ii) `power(int base, int exponent)`.
- **MAKE SURE YOU TYPE THE FOLLOWING CODE CAREFULLY. THERE ARE BUGS IN THIS CODE YOU SHOULD IDENTIFY WITH GDB**

Checkpoint 1

Add these function prototypes in lab3header.h and lab3.support.c. In the switch statement in the `main()` function, add these options when the user types “S” or “B” respectively.

Implement the two functions as follows:

```
int sumall(int a[], int size)
{
     int i, sum=0;
     for (i=0; 1<size; i++)
          sum+=a[i];
     return sum;
}


int power (int base, int exponent)
{
  if (exponent == 1)
    return base;
  else
    return base * power(base, exponent);
}
```

Define an array “a” in the main() function:
`int a[] = {4,8,10};`

Call the sumall function in the relevant position in the switch statement as follows:
`sumall(a,3);`

Ask for user input to get the values of base and exponent.
Call the power function in the relevant position in the switch statement as follows:
`result = power(base, exponent);`
where “result” is an integer.

CMSC 257 Lab 4
Debugging with GDB

Checkpoint 2

Here, you will use gdb to test out these two functions.
Compile your code using `gcc –g lab3support.c loop-control-chk1.c –o test`

Start gdb using
`gdb ./test`

then type,
r

Give the option of **S** to test the `sumall` function; you should see that the program encounters a segfault error and terminates.

Print out the values in the array. Then print out the value of "i".

Question 1. *What do you see? What value should "i" be? What value is "i"?*
$1 = (int *) 0x7fffffffe470
$2 = 740
I is supposed to be 3.
I is the counter variable used in the for loop

Checkpoint 3

Now quite out of gdb typing quit. Enter gdb again using `gdb ./test`.
Set up a breakpoint on the line

```
sum+=a[i];
```

Now run the program using "r". For each iteration of the for loop print out the value of "i"; do this 5 times. Do you spot the error?

I = 0,1,2,3,4,5
The loop is not adding correctly, it is just incrementing infinitely.

If so, quit out of gdb again and fix the error in the code. Restart gdb test and check out if your `sumall()` function is fixed.

Checkpoint 4

We will next test out the power function.
Start gdb as before and give the option "B" to call the `power()` function. Give inputs `base=3` and `exponent=4`.

*Type commands **p base** and **p exponent** to display the values of the base and the exponent. What responses do you get for each of the commands?*

$1 = 3

$2 = 4

CMSC 257 Lab 4
Debugging with GDB

Continue executing the program step-by-step by repeating the command **s**. Use command **where** to display the sequence of function calls.

Question 1. *What is the sequence of function calls after you repeat command* ***s*** *5 times?*

```
Breakpoint 1, power (base=3, exponent=4) at lab3support.c:79

Breakpoint 1, power (base=3, exponent=4) at lab3support.c:79

Breakpoint 1, power (base=3, exponent=4) at lab3support.c:79

Breakpoint 1, power (base=3, exponent=4) at lab3support.c:79

Breakpoint 1, power (base=3, exponent=4) at lab3support.c:79
```

The values of base and exponent are not changing

Continue testing the program until you figure out the error. If you want the program to run all the way to the end without stopping, type **c**. If you want to restart the program after it has finished running, type **run**. If you want to add more breakpoints, use command **b**. Use other GDB commands if needed.

Question 2. *What is the error in the program? What happens when the program runs with this error? Justify your answer by information that you got when running the debugger.*

The value of base and exponent are not changing between calls. The exponent value should be changing because recursion requires approaching an exit condition.
The program encounters this error caused by infinite recursion:

```
Program received signal SIGSEGV, Segmentation fault.
0x00000000004008ac in power (base=3, exponent=4) at lab3support.c:79
79                      return base * power(base,exponent);
```

Correct the error. Recompile the program (see above). You don't need to restart the debugger.

Question 3. *How many function calls does the program make before the recursive function starts returning? (Count main as the first call).*
It does infinite calls because the function does not approach the exit condition.