# include <stdio.h>

int main()

{

int i, num, j=1;

printf ("Enter the number: ");

scanf ("%d", &num );

for (i=1; i<=num; i++)

j=j\*i;

printf("The factorial of %d is %d\n",num,j);

}

**Run 1: badprog.c**

**% gcc -g badprog.c** **#-- compile program with -g flag**

**% gdb a.out** **#-- invoke gdb with the executable**

GNU gdb 6.4.90-debian

Copyright (C) 2006 Free Software Foundation, Inc.

GDB is free software, covered by the GNU General Public License, and you are

welcome to change it and/or distribute copies of it under certain conditions.

Type "show copying" to see the conditions.

There is absolutely no warranty for GDB. Type "show warranty" for details.

This GDB was configured as "i486-linux-gnu"...Using host libthread\_db library "/lib/tls/libthread\_db.so.1".

**(gdb) break main** **#-- set a breakpoint at the begining of the program's execution**

Breakpoint 1 at 0x8048436: file badprog.c, line 36.

**(gdb) run** **#-- run the program**

Starting program: /home/newhall/public/gdb\_examples/a.out

**Breakpoint 1, main () at badprog.c:36** **#-- gdb stops at breakpoint**

36 int arr[5] = { 17, 21, 44, 2, 60 };

**(gdb) list** **#-- list the source code near the break point**

31 return 0;

32 }

33

34 int main(int argc, char \*argv[]) {

35

36 int arr[5] = { 17, 21, 44, 2, 60 };

37

38 int max = arr[0];

39

40 if ( findAndReturnMax(arr, 5, max) != 0 ) {

**(gdb) list 11** **#-- list source code around line 11**

11 // this function should find the largest element in the array and

12 // "return" it through max

13 // array: array of integer values

14 // len: size of the array

15 // max: set to the largest value in the array

16 // reuturns: 0 on success and non-zero on an error

17 //

18 int findAndReturnMax(int \*array1, int len, int max) {

19

20 int i;

**(gdb) list** **#-- list the next few lines of code**

21

22 if(!array1 || (len <=0) ) {

23 return -1;

24 }

25 max = array1[0];

26 for(i=1; i <= len; i++) {

27 if(max < array1[i]) {

28 max = array1[i];

29 }

30 }

**(gdb) next** **#-- execute the next instruction**

38 int max = arr[0];

**(gdb)**  **#-- hitting Enter executes the previous command (next in this case)**

40 if ( findAndReturnMax(arr, 5, max) != 0 ) {

**#-- also you can use the up and down arrows to scroll through previous commands**

**(gdb) print max** **#-- print out the value of max**

$1 = 17

**(gdb) p arr** **#-- p is short for the print command**

$2 = {17, 21, 44, 2, 60}

**(gdb) step #-- step into the function call**

**#-- if we had entered 'next' the entire function call would have been executed**

findAndReturnMax (array1=0xbfc5cb3c, len=5, max=17) at badprog.c:22

22 if(!array1 || (len <=0) ) { **#-- 'step' takes us to the entry point of findAndReturnMax**

**(gdb) print array1[0]** **#-- lets see what the param values are**

$3 = 17

**(gdb) p max**

$4 = 17

**(gdb) list**

17 //

18 int findAndReturnMax(int \*array1, int len, int max) {

19

20 int i;

21

22 if(!array1 || (len <=0) ) {

23 return -1;

24 }

25 max = array1[0];

26 for(i=1; i <= len; i++) {

**(gdb) break 26** **#-- set a breakpoint at line 26 (inside findAndReturnMax)**

Breakpoint 2 at 0x80483e7: file badprog.c, line 26.

**(gdb) cont** **#-- continue the execution**

Continuing.

Breakpoint 2, findAndReturnMax (array1=0xbfc5cb3c, len=5, max=17) **#-- gdb hits the next breakpoint**

at badprog.c:26

26 for(i=1; i <= len; i++) {

**(gdb) p i**

$5 = 0

**(gdb) n** **#-- n is short for next**

27 if(max < array1[i]) {

**(gdb) display max** **#-- display will print out the value everytime we hit a breakpoint**

1: max = 17

**(gdb) display array1[i]**

2: array1[i] = 21

**(gdb) break 27** **#-- set a breakpoint inside the loop**

Breakpoint 3 at 0x80483f0: file badprog.c, line 27.

**(gdb) cont** **#-- continue execution**

Continuing.

Breakpoint 3, findAndReturnMax (array1=0xbfc5cb3c, len=5, max=21)

at badprog.c:27

27 if(max < array1[i]) {

**#-- display prints these out:**

2: array1[i] = 44

1: max = 21

**(gdb) cont**

Continuing.

Breakpoint 3, findAndReturnMax (array1=0xbfc5cb3c, len=5, max=44)

at badprog.c:27

27 if(max < array1[i]) {

2: array1[i] = 2

1: max = 44

**(gdb) cont**

Continuing.

Breakpoint 3, findAndReturnMax (array1=0xbfc5cb3c, len=5, max=44)

at badprog.c:27

27 if(max < array1[i]) {

2: array1[i] = 60

1: max = 44

**(gdb) cont**

Continuing.

Breakpoint 3, findAndReturnMax (array1=0xbfc5cb3c, len=5, max=60)

at badprog.c:27

27 if(max < array1[i]) {

2: array1[i] = 17

1: max = 60 **#-- so max is 60 here**

**(gdb) where** **#-- show the stack frames**

**#-- findAndReturnMax is the active function at line 27, it was called by main at line 40:**

#0 findAndReturnMax (array1=0xbfd043ec, len=5, max=60) at badprog.c:27

#1 0x08048479 in main () at badprog.c:40

**frame 1** **#-- move into main's calling context (stack frame 1) to examine main's state**

#1 0x08048479 in main () at badprog.c:40

40 if ( findAndReturnMax(arr, 5, max) != 0 ) {

**(gdb) print max** **#-- in main's stack frame max is 17**

$1 = 17

**(gdb) cont** **#-- continue execution**

Continuing.

max value in the array is 17 **#-- main prints out value of max after function call**

**#-- This looks like a bug:"**

**#-- findAndReturnMax set max to 60, but 60 isn't getting "passed back" to main after the call**

**#-- to fix this we need either have findAndReturnMax return the value of max or pass max by reference**

**(gdb) quit** **#-- quit gdb**

The program is running. Exit anyway? (y or n) y