Assignment 01 – Warm-Up (Due: Friday February 16, 2024)

Now that you all have indicated that you have successfully been able to use the VPN to log in to and interact with the department's Linux servers, this simple assignment calls upon you to modify some simple and limited C source programs. I plan to keep track of your submissions but do not plan to undertake any deep review of them, but I may change my mind. The important thing is that you put in the time and effort to successfully accomplish the work of this assignment. We will have an in-class assessment (i.e. quiz) closely related to this assignment, and your performance on this will be factoring directly into grading.

The three C source programs, Fibonacci.c, Fibonacci2.c and Fibonacci3.c, each prompt the user to enter an integer, to be used as an upper bound, and then print the "Fibonacci Sequence" up to that bound. Your task for this assignment is to modify and rename each of these programs so that they each instead print the "Tribonacci Sequence" up to the inputted upper bound.

```
Fibonacci Program...
Enter an integer:100
1
1
2
3
5
8
13
21
34
55
89
Done!
```

```
Tribonacci Program...
Enter an integer:100
1
1
2
4
7
13
24
44
81
Done!
```

Note that the Tribonacci Sequence differs from the Fibonacci Sequence in that each subsequent term is the sum of the preceding three terms, rather than the sum of the preceding two terms.

Consider these sample executions.

The three versions of the given program each produce the same results. These given programs differ is how they are structured and how they go about performing the simple computation.

Since these programs are just simple C programs you may be tempted to accomplish this assignment without making use of the department's servers (`davinci` and `archimedes`). To do so would fall short of the full intent of this assignment being a "warm up". So, to force you to make use of the servers there is a second aspect of this assignment.

After you have successfully developed the three Tribonacci programs you are to additionally recompile the **`Tribonacci2.c`** program using the -S option so as to produce the equivalent 32-bit Architecture assembly language program in a file named **`Tribonacci2.s`**.

```
gcc -m32 Tribonacci2.c -S
```

You are to then perform a side-by-side comparison of the C source code with its corresponding assembly source code, for the purpose of identifying correlations between the items in the C version with their corresponding items in the assembly version. This includes as many of the identifiers and control flow constructs as you can correlate.

Consider the table presented at the end of this document as a format to be used to present the correlations identified.

As your work on this assignment you are to submit each of the following to the AS01 DropBox on Brightspace:

Tribonacci.c	Modified version of **Fibonacci.c, with identifying comments added.**
Tribonacci2.c	Modified version of **Fibonacci2.c, with identifying comments added.**
Tribonacci3.c	Modified version of **Fibonacci3.c, with identifying comments added.**
***LastFM*.docx**	Concise report of the results of your work on this assignment, in which you explain what you have done and provide screen shots of sample executions of each of the three programs. Among the "pet peeves" that I have is this. I dislike screen shots that employ black backgrounds. I find them to be too stark and difficult to read. So, consider it a requirement that the screen shots you present employ white backgrounds with generally black foreground text. This document is also to contain a table, similar to the one mentioned above and provided below, through which you are present the correlations you have identified. The file is required to be a MS Word readable document file, to be named in the following form, ***LastFM*.docx** (where ***Last*** is your Last Name, and ***F*** and ***M*** are your First Name and Middle Name initials; such as **JackowitzPM.docx**, and **SmithJ.docx**).

Good luck,
P.M.J.

Fibonacci.c

```
#include <stdio.h>
int main()  {
  int limit;
  int term1;
  int term2;
  int next;
  printf("Fibonacci Program...\n");
  printf("Enter an integer:");
  scanf("%d", &limit);
  term1 = 1;
  term2 = 1;
  if(term1 <= limit) {
    printf("%d\n",term1);
    while(term2 <= limit) {
      printf("%d\n",term2);
      next = term1 + term2;
      term1 = term2;
      term2 = next;
    }
  }
  printf("Done!\n");
}
```

Fibonacci.s

```
   .file "Fibonacci.c"
   .text
   .section .rodata
.LC0:
   .string  "Fibonacci Program..."
.LC1:
   .string  "Enter an integer:"
.LC2:
   .string  "%d"
.LC3:
   .string  "%d\n"
.LC4:
   .string  "Done!"
   .text
   .globl   main
   .type main, @function
main:
.LFB0:
   .cfi_startproc
   leal  4(%esp), %ecx
   .cfi_def_cfa 1, 0
   andl  $-16, %esp
   pushl -4(%ecx)
   pushl %ebp
   movl  %esp, %ebp
   .cfi_escape 0x10,0x5,0x2,0x75,0
   pushl %ecx
   .cfi_escape 0xf,0x3,0x75,0x7c,0x6
   subl  $20, %esp
```

```
   subl  $12, %esp
   pushl $.LC0
   call  puts
   addl  $16, %esp
   subl  $12, %esp
   pushl $.LC1
   call  printf
   addl  $16, %esp
   subl  $8, %esp
   leal  -24(%ebp), %eax
   pushl %eax
   pushl $.LC2
   call  __isoc99_scanf
   addl  $16, %esp
   movl  $1, -12(%ebp)
   movl  $1, -16(%ebp)
   movl  -24(%ebp), %eax
   cmpl  %eax, -12(%ebp)
   jg .L2
   subl  $8, %esp
   pushl -12(%ebp)
   pushl $.LC3
   call  printf
   addl  $16, %esp
   jmp   .L3
.L4:
   subl  $8, %esp
   pushl -16(%ebp)
   pushl $.LC3
   call  printf
   addl  $16, %esp
   movl  -12(%ebp), %edx
   movl  -16(%ebp), %eax
   addl  %edx, %eax
   movl  %eax, -20(%ebp)
   movl  -16(%ebp), %eax
   movl  %eax, -12(%ebp)
   movl  -20(%ebp), %eax
   movl  %eax, -16(%ebp)
.L3:
   movl  -24(%ebp), %eax
   cmpl  %eax, -16(%ebp)
   jle   .L4
.L2:
   subl  $12, %esp
   pushl $.LC4
   call  puts
   addl  $16, %esp
   movl  $0, %eax
   movl  -4(%ebp), %ecx
   .cfi_def_cfa 1, 0
   leave
   .cfi_restore 5
   leal  -4(%ecx), %esp
   .cfi_def_cfa 4, 4
   ret
   .cfi_endproc
.LFE0:
   .size main, .-main
   .ident   "GCC: (GNU) 8.5.0 20210514 (Red Hat 8.5.0-10.1.0.1)"
   .section .note.GNU-stack,"",@progbits
```