Programming Problems 3.14

Instruction & Output:

make

```
cskt@cskt-VirtualBox:~/Desktop/os_linux/3.14$ make
gcc -o main.o main.c
```

./main.o 4

./main.o 7

output sequence and invoke wait() that parent process wait until child process is complete

./main.o -3

when enter non positive integer, output “Invalid input : [input number], positive integer required”

```
cskt@cskt-VirtualBox:~/Desktop/os_linux/3.14$ ./main.o 4
4       2       1
Child process is done
Parent process is done
cskt@cskt-VirtualBox:~/Desktop/os_linux/3.14$ ./main.o 7
7       22      11      34      17      52      26      13      40      20      10      5       16      8       4       2       1
Child process is done
Parent process is done
cskt@cskt-VirtualBox:~/Desktop/os_linux/3.14$ ./main.o -3
Invalid input: -3, positive integer required
cskt@cskt-VirtualBox:~/Desktop/os_linux/3.14$
```

Source code:

Using the fork() system call that generates the Collatz conjecture sequence in the child process. Then have the parent invoke the wait() call to wait for the child process to complete before exiting the program.

```
31      pid = fork();
32
33      if(pid<0){ // error encountered
34          fprintf(stderr, "Fork Failed");
35          return 1;
36      }else if(pid == 0){ // child process
37          while(n!=1){
38              printf("%d\t", n);
39              n = collatz(n);
40          }
41          printf("1\n");
42          printf("Child process is done\n");
43      }else{ // parent process if(pid>0)
44          wait(NULL);
45          printf("Parent process is done\n");
46      }
47      return 0;
```

Fuction that return value of The Collatz conjecture.

```
 7  int collatz(int n){
 8      if(n%2==0)
 9          return n/2;
10      else
11          return 3*n+1;
12  }
```

Error checking that make sure the input value is positive integer on the command line.

```c
int main(int argc, char **argv){
    int n;
    pid_t pid;

    if (argc > 1)
    {
        //sscanf(argv[1], "%d", &n);
        n = atoi(argv[1]);
        if (n < 1){
            printf("Invalid input: %d, positive integer required\n", n);
            return 0;
        }
    }else{
        return 1;
    }
```