1.

#include <unistd.h>

#include <errno.h>

#include <stdio.h>

#include <sys/wait.h>

int main(){

pid\_t pid=fork();

if (pid<0)

return errno;

else

if(pid==0){

char \*argv[] = {"ls", NULL};

execve("/bin/ls", argv, NULL);

perror(NULL);}

else {

wait(NULL);

printf("Child %d Me %d\n", pid, getpid());}

return 0;

}

2.

#include <unistd.h>

#include <errno.h>

#include <stdio.h>

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

int n;

pid\_t pid=fork();

if(pid < 0)

return errno;

else

if (pid==0){

n=atoi(argv[1]);

printf("%d: ", n);

while(n>1){

printf("%d ", n);

if(n%2==0)

n=n/2;

else

n=3\*n+1;

}

printf("%d\n",n);

}

else

printf("Child %d finished\n", getpid());

return(0);

}

3.

#include <unistd.h>

#include <errno.h>

#include <stdio.h>

int i=1, n;

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

for(i=1;i<argc;i++){

pid\_t pid=fork();

if(pid < 0)

return errno;

else

if (pid==0){

n=atoi(argv[i]);

printf("%d: ", n);

while(n>1){

printf("%d ", n);

if(n%2==0)

n=n/2;

else

n=3\*n+1;

}

printf("%d\n",n);

printf("Done parent %d Me %d finished\n",getppid(), getpid());

exit(0);

}

}

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

wait(NULL);

printf("Done parent %d Me %d finished\n",getppid(), getpid());

return(0);

}