6. Write a C program that behaves like a shell (command interpreter). It has its

own prompt say “NewShell$”. Any normal shell command is executed from

your shell by starting a child process to execute the system program

corresponding to the command. It should additionally interpret the following

command.

i) list f<dirname> - print name of all files in directory

ii) list n <dirname> - print number of all entries

iii) list i<dirname> - print name and inode of all files

#include<stdio.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<unistd.h>

#include<dirent.h>

#include<fcntl.h>

#include<stdlib.h>

void list(char c, char \*dn)

{ DIR \*dir;

int cnt=0;

struct dirent \*entry;

struct stat buff;

if((dir=opendir(dn))==NULL)

{

printf("Directory %s not found\n",dn);

return;

}

switch(c)

{

case 'f':

while((entry=readdir(dir))!=NULL)

{

stat(entry->d\_name,&buff);

printf("%s\n",entry->d\_name);

}

break;

case 'n':

while((entry=readdir(dir))!=NULL)

cnt++;

printf("Total No.of Entries = %d\n",cnt);

break;

case 'i':

while((entry=readdir(dir))!=NULL)

{

stat(entry->d\_name,&buff);

printf("%s\t%d\n",entry->d\_name,buff.st\_ino);

}

break;

}

closedir(dir);

}

main()

{ char command[80],t1[20],t2[20],t3[20],t4[20];

int n;

system("clear");

while(1)

{

printf("myShell$");

fflush(stdin);

fgets(command,80,stdin);

n = sscanf(command,"%s %s %s %s",t1,t2,t3,t4);

if(strcmp(t1,"list")==0)

list(t2[0],t3);

else

exit(0);

}

}