14. 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) search f <pattern><filename> - search first occurrence of pattern in

filename

ii) search c <pattern><filename> - count no. of occurrences of pattern in

filename

iii) search a <pattern><filename> - search all occurrences of pattern in

filename

#include<stdio.h>

#include<unistd.h>

#include<fcntl.h>

#include<string.h>

#include<stdlib.h>

void search(char c, char \*s, char \*fn)

{

int handle,i=1,cnt=0,j=0;

char ch,buff[80],\*p;

if((handle=open(fn,O\_RDONLY))==-1)

{

printf("File %s not found\n",fn);

return;

}

switch(c)

{

case 'f':

while(read(handle,&ch,1)!=0)

{

if(ch=='\n')

{

buff[j]='\0'; /\* \0 is udes to terminate the string which represents a null character \*/

j=0;

if(strstr(buff,s)!=NULL)

{

printf("%d : %s\n",i,buff);

break;

}

i++;

}

else

buff[j++]=ch;

}

break;

case 'c':

while(read(handle,&ch,1)!=0)

{

if(ch=='\n')

{

buff[j]='\0';

j=0;

if(strstr(buff,s)!=NULL)

{

cnt++;

}

}

else

buff[j++]=ch;

}

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

break;

case 'a':

while(read(handle,&ch,1)!=0)

{

if(ch=='\n')

{

buff[j]='\0';

j=0;

if(strstr(buff,s)!=NULL)

printf("%d : %s\n",i,buff);

i++;

}

else

buff[j++]=ch;

}

}

close(handle);

}

int main()

{

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

int n;

system("clear");

while(1)

{

printf("NewShell$");

fflush(stdin);

fgets(command,80,stdin);

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

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

search(t2[0],t3,t4);

else

exit(0);

}

}