**Q0 Sample Program**

#include <stdio.h>

#include <stdlib.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <stdlib.h>

int main()

{

char c;

int in, out;

char buffer[128];

int nread;

in=open("samp.txt",O\_RDWR);

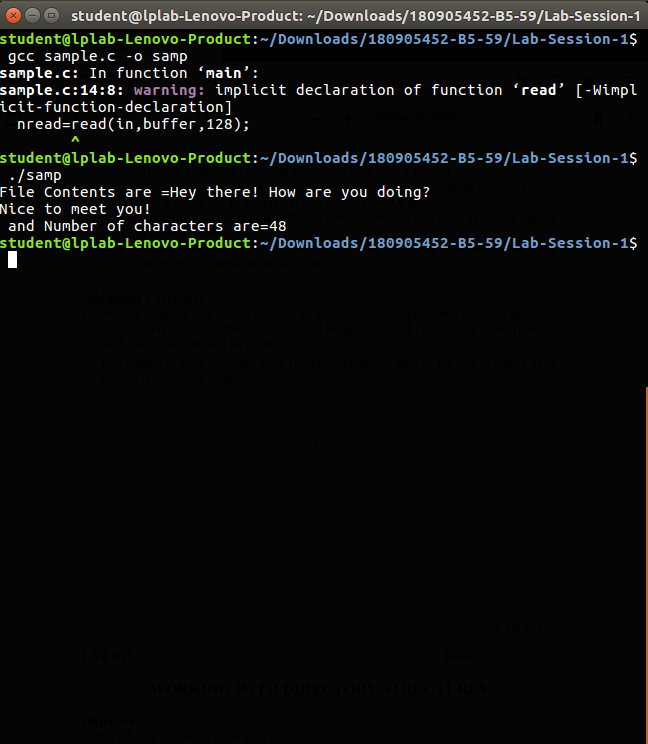
nread=read(in,buffer,128);

printf("File Contents are =%s and Number of characters are=%d\n",buffer, nread);

exit(0);

}

**Output**



**Lab 1-Working with Regular Files**

1.) Write a program to print the lines of a file that contain a word given as the program argument (a simple version of grep UNIX utility

#include <stdio.h>

#include <unistd.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <stdlib.h>

#include <string.h>

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

{

int sfd,i=0,k=0;

char ch[100],chr;

if(argc!=3)

{

printf("Insufficient Arguments\n");

exit(1);

}

if( (sfd=open(argv[2],O\_RDONLY))==-1)

{

printf("File not found\n");

exit(1);

}

// printf("File Contents=%s and Number of characters are=%d",buffer, nread);

while((read(sfd,&chr,1))>0)

{

if(chr!='\n')

{

ch[i]=chr;

i++;

}

else

{

k++;

ch[i]='\0';

i=0;

if(strstr(ch,argv[1])!=NULL){

printf("Line:%d \t %s \n", k,ch);

}

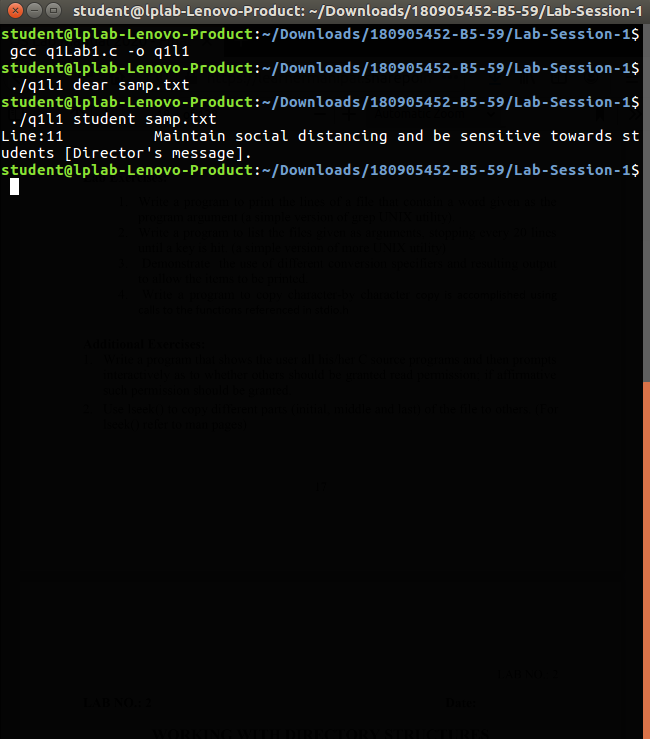
}

}

exit(0);

}

**Output**



2.) Write a program to list the files given as arguments, stopping every 20 lines until a key is hit. (a simple version of more UNIX utility)

#include <stdio.h>

#include <unistd.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <stdlib.h>

#include <string.h>

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

{

int sfd,sfd2,i=0,k=0,p=0;

char ch[100],ch2[100],chr;

if(argc!=3)

{

printf("Insufficient Arguments\n");

exit(1);

}

if( (sfd=open(argv[1],O\_RDONLY))==-1)

{

printf("File not found\n");

exit(1);

}

// printf("File Contents=%s and Number of characters are=%d",buffer, nread);

while((read(sfd,&chr,1))>0)

{

if(chr!='\n')

{

ch[i]=chr;

i++;

}

else

{

k++;

p++;

ch[i]='\0';

i=0;

printf("Line:%d \t %s \n", p,ch);

if(k==20)

{

fgetc(stdin);

k=0;

}

}

}

close(sfd);

if( (sfd2=open(argv[2],O\_RDONLY))==-1)

{

printf("File not found\n");

exit(1);

}

p=0;

while((read(sfd,&chr,1))>0)

{

if(chr!='\n')

{

ch2[i]=chr;

i++;

}

else

{

k++;

p++;

ch[i]='\0';

i=0;

printf("Line:%d \t %s \n", p,ch2);

if(k==20)

{

fgetc(stdin);

k=0;

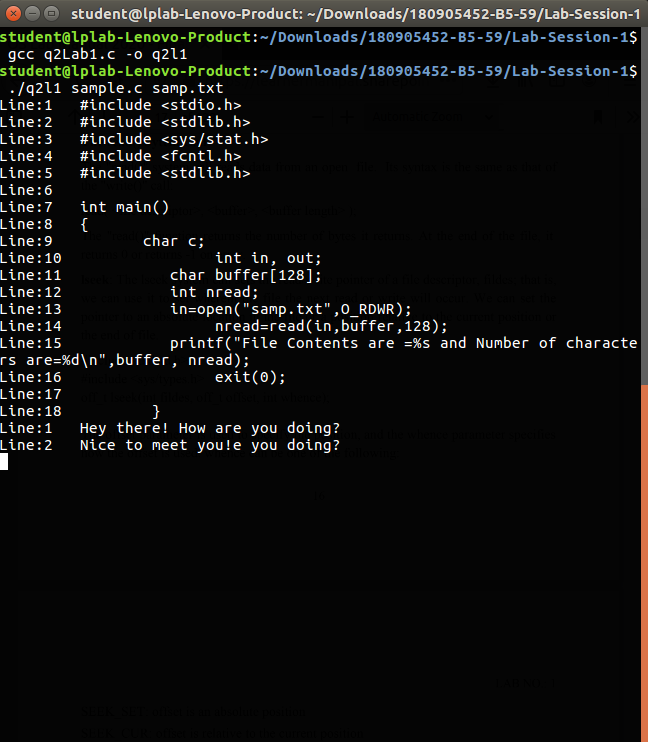
}

}

}exit(0);

}

**Output**



3.) Demonstrate the use of different conversion specifiers and resulting output to allow the items to be printed.

#include<stdio.h>

#include<stdlib.h>

#include<errno.h>

// extern int errno;

int main()

{

int a=81;

float b=9.1;

char c='p';

char str[]="OS Lab";

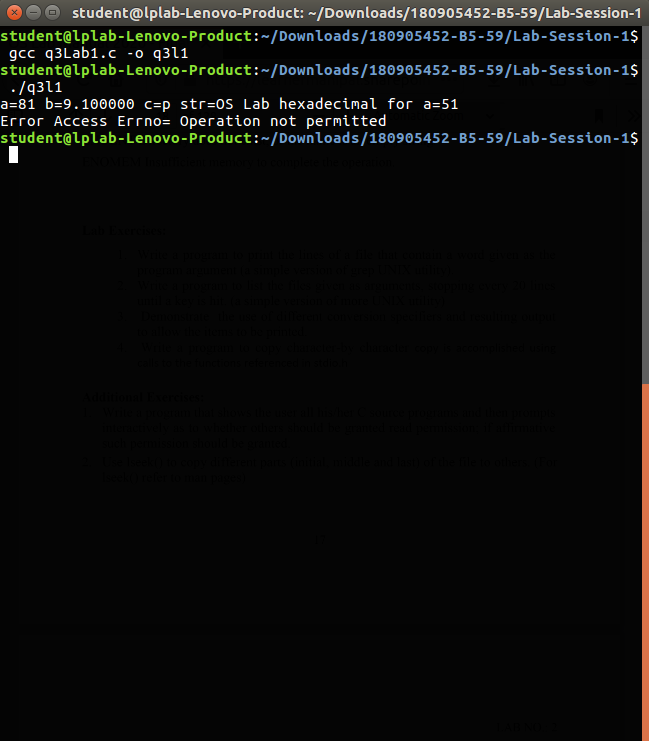
printf("a=%d b=%f c=%c str=%s hexadecimal for a=%x \n",a,b,c,str,a);

errno=EPERM;

printf("Error Access Errno= %m\n");

}

**Output**



4.) Write a program to copy character-by character copy is accomplished using calls to the functions referenced in stdio.h.

#include<stdio.h>

#include <unistd.h>

#include<sys/stat.h>

#include<fcntl.h>

#include<stdlib.h>

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

{

char c;

int in,out;

char buffer[128];

int nread;

if(argc!=3)

{

printf("Insufficient Arguments\n");

exit(1);

}

in=open(argv[1],O\_RDWR);

out=open(argv[2],O\_WRONLY|O\_CREAT, S\_IRUSR|S\_IWUSR);

if( in==-1 || out==-1)

{

printf("File not found\n");

exit(1);

}

while(read(in,&c,1) == 1)

{

write(out,&c,1);

}

printf("Contents of file copied\n");

exit(0);

}

**Output**

