OS Lab 1

Name: Paawan Kohli

Reg No: 180905416

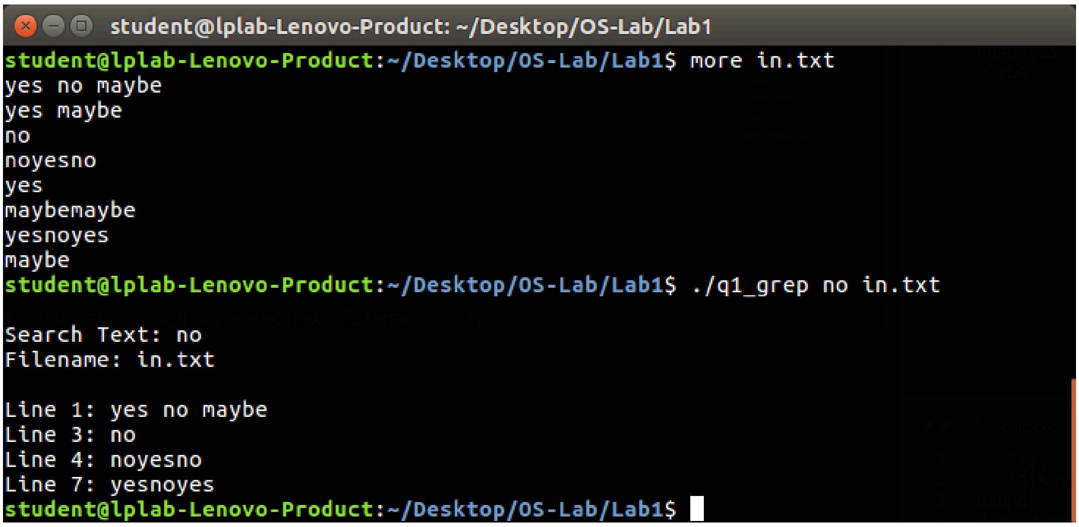
Roll No: 52

Q1. Write a program to print the lines of a file that contains a word given as the program argument (a simple version of grep UNIX utility).

```
how to run: . / q1_grep searchText filename
```

```
#include <unistd.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void main(int argv, char **arg) {

        // must have searchText and filename
        if (argv != 3) {
                printf("Invalid syntax.\nFormat: ./q1_grep searchText filename \n");
                exit(0);
        }

        char searchText[100];
        strcpy(searchText, arg[1]);

        char filename[100];
        strcpy(filename, arg[2]);

        printf("\n");
        printf("Search Text: %s \n", searchText);
        printf("Filename: %s \n", filename);
        printf("\n");

        // open file
        int fd = open(filename, O_RDONLY);

        // check if file exists
        if (fd == -1) {
                printf("%s doesn't exist.\n", filename);
                exit(0);
        }

        char buffer[1000] = "";
        char c;

        int currentLine = 1;

        // read char by char
        while (read(fd, &c, 1) == 1) {

                if (c != '\n') {
                        // if the char is NOT \n, append it to the string
                        strncat(buffer, &c, 1);
                }
                else {
                        // if char is append, reading one line is done. Do substring match

                        if (strstr(buffer, searchText) != NULL) {
                                // print the line if a match is found
                                printf("Line %d: %s\n", currentLine, buffer);
                        }
```

```
                                // increment current line count and reset buffer
                                currentLine++;
                                buffer[0] = '\0';
                        }
                }

                // last line is left in buffer
                if (strstr(buffer, searchText) != NULL) {
                        // print the line if a match is found
                        printf("Line %d: %s\n", currentLine, buffer);
                }

                close(fd);
}
```

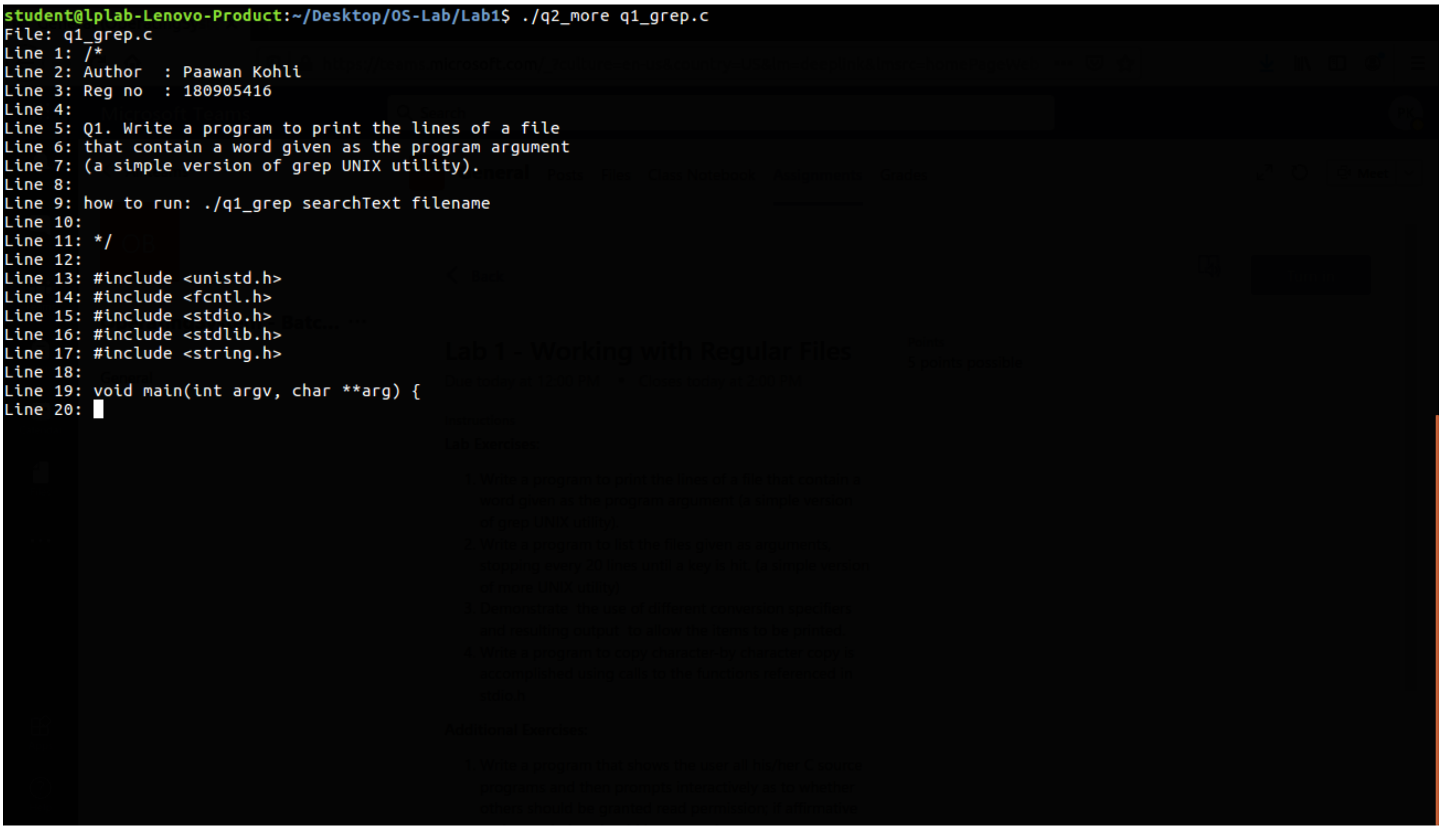


```
student@lplab-Lenovo-Product:~/Desktop/OS-Lab/Lab1$ more in.txt
yes no maybe
yes maybe
no
noyesno
yes
maybemaybe
yesnoyes
maybe
student@lplab-Lenovo-Product:~/Desktop/OS-Lab/Lab1$ ./q1_grep no in.txt

Search Text: no
Filename: in.txt

Line 1: yes no maybe
Line 3: no
Line 4: noyesno
Line 7: yesnoyes
student@lplab-Lenovo-Product:~/Desktop/OS-Lab/Lab1$
```

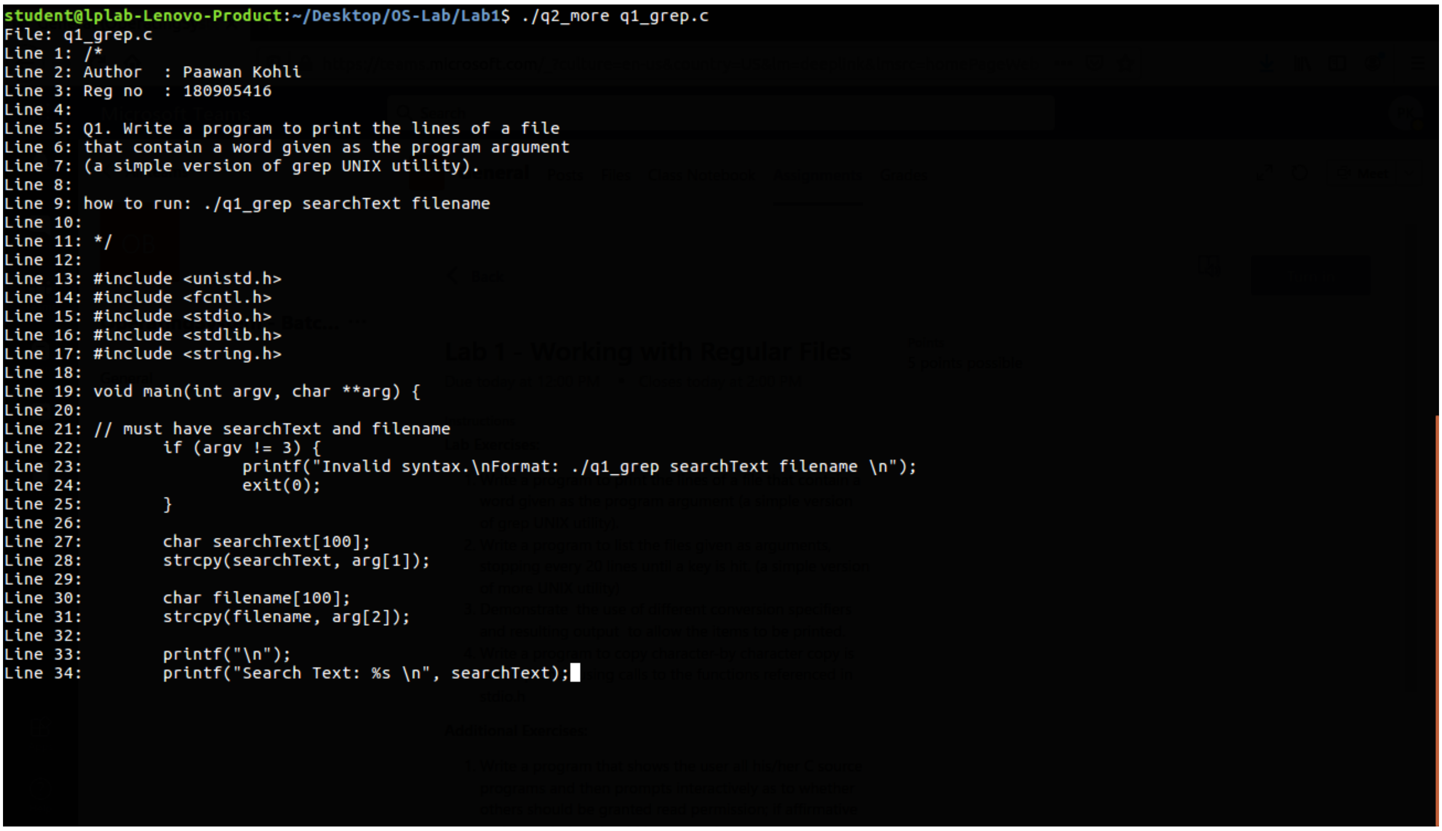
Q2. Write a program to list the files given as arguments, stopping every 20 buffers until a key is hit (a simple version of more UNIX utility)

```
how to run: ./q2_more filename
```

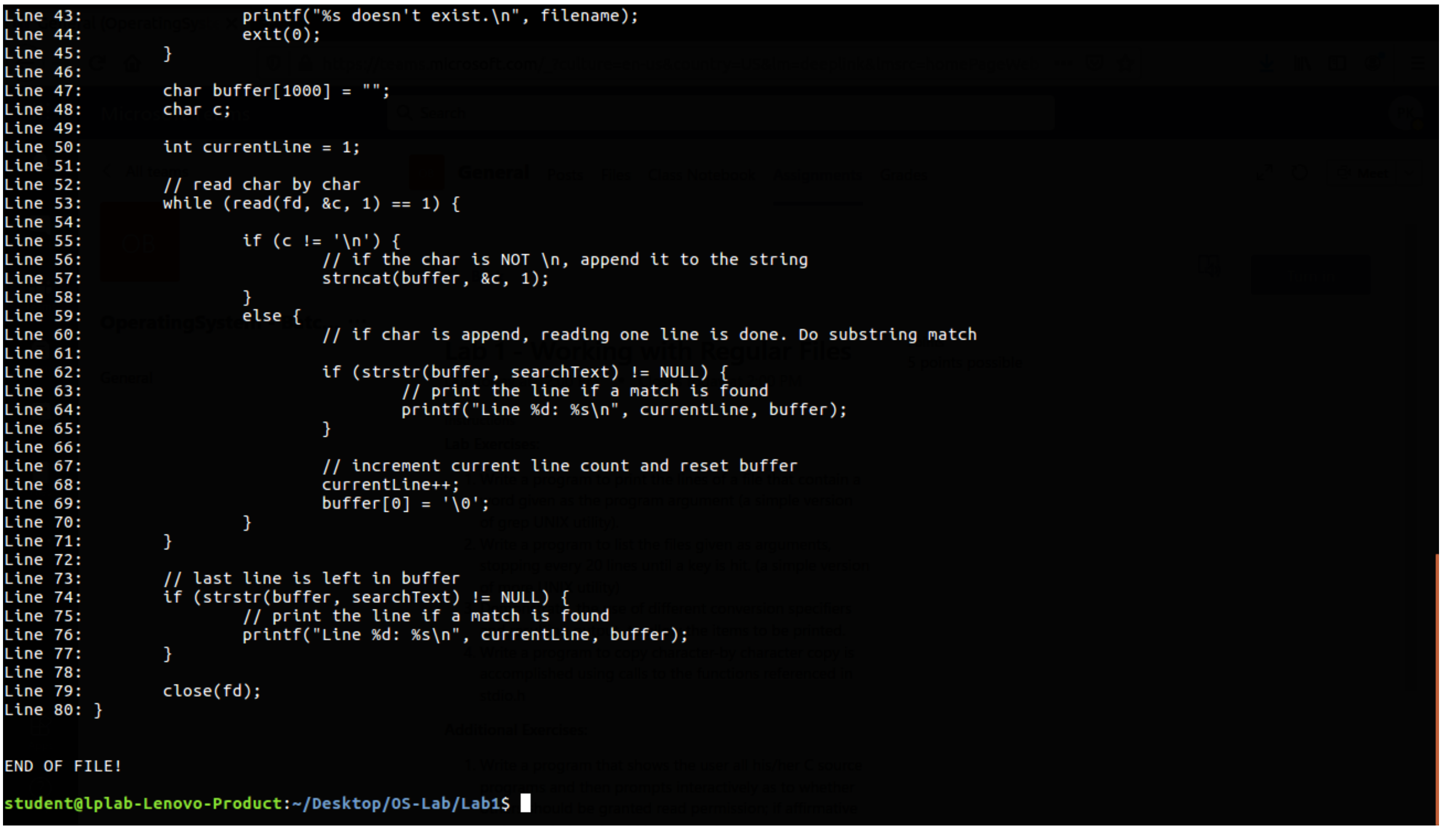
```
#include <unistd.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void main (int argv, char **arg) {

        // must have filename
        if (argv != 2) {
                printf("Invalid syntax.\nFormat: ./q2_more filename \n");
                exit(0);
        }

        char filename[100];
        strcpy(filename, arg[1]);

        printf("File: %s\n", filename);

        int fd = open(filename, O_RDONLY);

        // check if file exists
        if (fd == -1) {
                printf("%s doesn't exist.\n", filename);
                exit(0);
        }

        char buffer[1000] = "";
        char c;
        int currentLine = 1;
        while (read(fd, &c, 1) == 1 && currentLine <= 20) {
                if (c != '\n') {
                        strncat(buffer, &c, 1);
                } else {
                        printf("Line %d: %s", currentLine, buffer);

                        if (currentLine != 20) {
                                printf("\n");
                        }

                        buffer[0] = '\0';
                        currentLine++;
                }
        }

        while (read(fd, &c, 1) == 1) {
                if (c != '\n') {
                        strncat(buffer, &c, 1);
                } else {
                        // wait for key press before printing
                        char keyPress = getchar();
                        // read(0, &keyPress, 1);

                        printf("Line %d: %s", currentLine, buffer);
                        buffer[0] = '\0';
                        currentLine++;
                }
        }

        // last line is left in buffer
        char keyPress = getchar();
        // read(0, &keyPress, 1);
        printf("Line %d: %s\n", currentLine, buffer);

        close(fd);

        printf("\n\nEND OF FILE!\n\n");
}
```



```
student@lplab-Lenovo-Product:~/Desktop/OS-Lab/Lab1$ ./q2_more q1_grep.c
File: q1_grep.c
Line 1: /*
Line 2: Author   : Paawan Kohli
Line 3: Reg no   : 180905416
Line 4:
Line 5: Q1. Write a program to print the lines of a file
Line 6: that contain a word given as the program argument
Line 7: (a simple version of grep UNIX utility).
Line 8:
Line 9: how to run: ./q1_grep searchText filename
Line 10:
Line 11: */
Line 12:
Line 13: #include <unistd.h>
Line 14: #include <fcntl.h>
Line 15: #include <stdio.h>
Line 16: #include <stdlib.h>
Line 17: #include <string.h>
Line 18:
Line 19: void main(int argv, char **arg) {
Line 20:
```



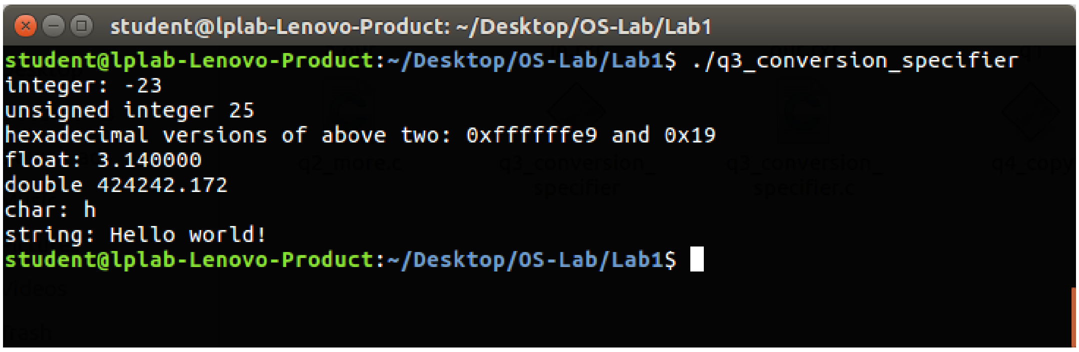
```
student@lplab-Lenovo-Product:~/Desktop/OS-Lab/Lab1$ ./q2_more q1_grep.c
File: q1_grep.c
Line 1: /*
Line 2: Author   : Paawan Kohli
Line 3: Reg no   : 180905416
Line 4:
Line 5: Q1. Write a program to print the lines of a file
Line 6: that contain a word given as the program argument
Line 7: (a simple version of grep UNIX utility).
Line 8:
Line 9: how to run: ./q1_grep searchText filename
Line 10:
Line 11: */
Line 12:
Line 13: #include <unistd.h>
Line 14: #include <fcntl.h>
Line 15: #include <stdio.h>
Line 16: #include <stdlib.h>
Line 17: #include <string.h>
Line 18:
Line 19: void main(int argv, char **arg) {
Line 20:
Line 21: // must have searchText and filename
Line 22:        if (argv != 3) {
Line 23:                printf("Invalid syntax.\nFormat: ./q1_grep searchText filename \n");
Line 24:                exit(0);
Line 25:        }
Line 26:
Line 27:        char searchText[100];
Line 28:        strcpy(searchText, arg[1]);
Line 29:
Line 30:        char filename[100];
Line 31:        strcpy(filename, arg[2]);
Line 32:
Line 33:        printf("\n");
Line 34:        printf("Search Text: %s \n", searchText);
```



```
Line 43:                printf("%s doesn't exist.\n", filename);
Line 44:                exit(0);
Line 45:        }
Line 46:
Line 47:        char buffer[1000] = "";
Line 48:        char c;
Line 49:
Line 50:        int currentLine = 1;
Line 51:
Line 52:        // read char by char
Line 53:        while (read(fd, &c, 1) == 1) {
Line 54:
Line 55:                if (c != '\n') {
Line 56:                        // if the char is NOT \n, append it to the string
Line 57:                        strncat(buffer, &c, 1);
Line 58:                }
Line 59:                else {
Line 60:                        // if char is append, reading one line is done. Do substring match
Line 61:
Line 62:                        if (strstr(buffer, searchText) != NULL) {
Line 63:                                // print the line if a match is found
Line 64:                                printf("Line %d: %s\n", currentLine, buffer);
Line 65:                        }
Line 66:
Line 67:                        // increment current line count and reset buffer
Line 68:                        currentLine++;
Line 69:                        buffer[0] = '\0';
Line 70:                }
Line 71:        }
Line 72:
Line 73:        // last line is left in buffer
Line 74:        if (strstr(buffer, searchText) != NULL) {
Line 75:                // print the line if a match is found
Line 76:                printf("Line %d: %s\n", currentLine, buffer);
Line 77:        }
Line 78:
Line 79:        close(fd);
Line 80: }

END OF FILE!
student@lplab-Lenovo-Product:~/Desktop/OS-Lab/Lab1$ 
```

Q3. Demonstrate the use of different conversion specifiers and resulting output to allow the items to be printed.

```
#include <stdio.h>

void main() {
        int x = -23;
        printf("integer: %d\n", x);

        unsigned int y = 25;
        printf("unsigned integer %u\n", y);

        printf("hexadecimal versions of above two: %#x and %#x\n", x, y);

        float z = 3.14;
        printf("float: %f\n", z);

        double d = 424242.171717;
        printf("double %3.3lf\n", d);


        char c = 'h';
        printf("char: %c\n", c);

        char str[] = "Hello world!";
        printf("string: %s\n", str);
}
```

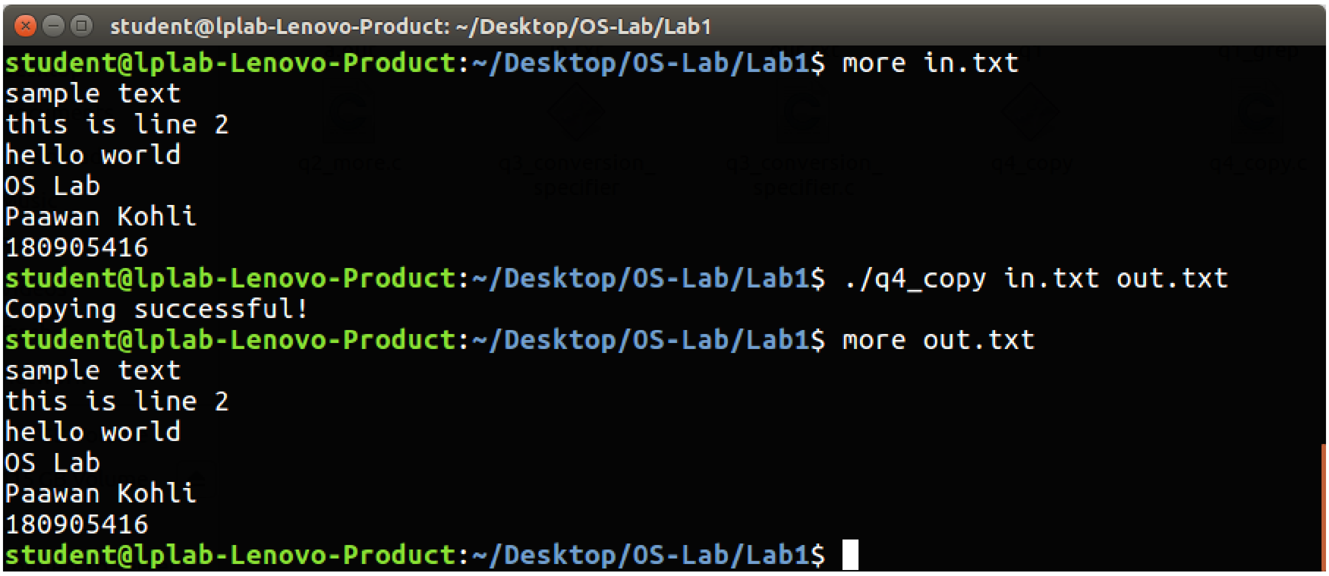


```
student@lplab-Lenovo-Product:~/Desktop/OS-Lab/Lab1$ ./q3_conversion_specifier
integer: -23
unsigned integer 25
hexadecimal versions of above two: 0xffffffe9 and 0x19
float: 3.140000
double 424242.172
char: h
string: Hello world!
student@lplab-Lenovo-Product:~/Desktop/OS-Lab/Lab1$ 
```

Q4. Write a program to copy character by character copy is accomplished using calls to the functions referenced in stdio.h

```
how to run: ./q4_copy sourceFilename destinationFilename
```

```
#include <unistd.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void main(int argv, char ** arg) {

        // must have searchText and filename
        if (argv != 3) {
                printf("Invalid syntax.\nFormat: ./q4_copy sourceFilename destinationFilename \n");
                exit(0);
        }

        char srcFile[100], dstFile[100];
        strcpy(srcFile, arg[1]);
        strcpy(dstFile, arg[2]);

        FILE* in = fopen(srcFile, "r");
        FILE* out = fopen(dstFile, "w");

        char c;

        while ( (c = fgetc(in)) != EOF) {
                fputc(c, out);
        }

        fclose(in);
        fclose(out);

        printf("Copying successful!\n");
}
```



```
student@lplab-Lenovo-Product:~/Desktop/OS-Lab/Lab1$ more in.txt
sample text
this is line 2
hello world
OS Lab
Paawan Kohli
180905416
student@lplab-Lenovo-Product:~/Desktop/OS-Lab/Lab1$ ./q4_copy in.txt out.txt
Copying successful!
student@lplab-Lenovo-Product:~/Desktop/OS-Lab/Lab1$ more out.txt
sample text
this is line 2
hello world
OS Lab
Paawan Kohli
180905416
student@lplab-Lenovo-Product:~/Desktop/OS-Lab/Lab1$ 
```