Dennis Johnson
180905025
CSE - B
Batch 3

OS – Lab

LAB-1

Q1. **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/types.h>
#include<fcntl.h>
#include<string.h>
#include<stdlib.h>

#define LEN_BUFFER 256
/*
 * Search for a word in file given as cmd line ard
 * Similar to grep for a single file
 * */

int main(int argc, char* argv[]){
  int in;
  char ch, buffer[LEN_BUFFER];

  if(argc != 3){
    fprintf(stderr, "Usage: arg1 - word, arg2 - file src to search in");
  }

  if((in = open(argv[2], O_RDONLY)) < 0){
    fprintf(stderr, "Error opening file %s", argv[2]);
    perror(" ");
    exit(EXIT_FAILURE);
  }

  size_t len_read, char_count = 0;
  while((len_read = read(in, &ch, sizeof(char))) > 0){
    if (ch == '\n'){
      buffer[char_count] = '\0';

      if(strstr(buffer, argv[1]) != NULL)
        printf("Found '%s': line --> %s\n", argv[1], buffer);

      char_count = 0;
      memset(buffer, '\0', sizeof(buffer));
    }

    else {
      buffer[char_count++] = ch;
    }
  }
```

```
 close(in);
        return 0;
}
```

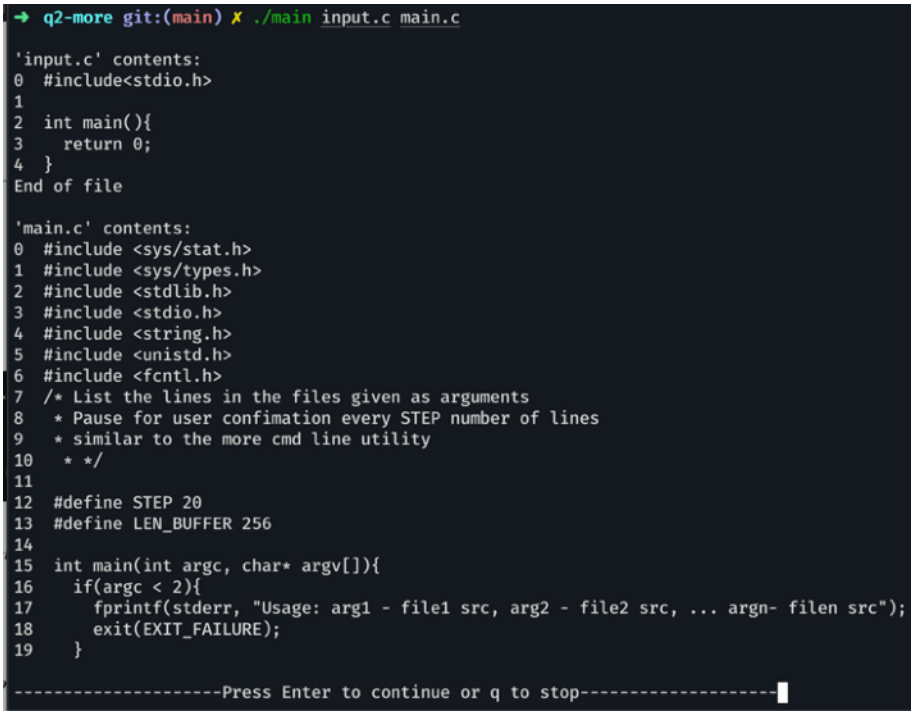
```
➜  q1-grep git:(main) ✗ ./main buffer main.c
Found 'buffer': line -->   char ch, buffer[LEN_BUFFER];
Found 'buffer': line -->       buffer[char_count] = '\0';
Found 'buffer': line -->       if(strstr(buffer, argv[1]) != NULL)
Found 'buffer': line -->         printf("Found '%s': line --> %s\n", argv[1], buffer);
Found 'buffer': line -->       memset(buffer, '\0', sizeof(buffer));
Found 'buffer': line -->       buffer[char_count++] = ch;
```

Q2. 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 <sys/stat.h>

#include <sys/types.h>

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <unistd.h>

#include <fcntl.h>

/* List the lines in the files given as arguments

 * Pause for user confimation every STEP number of lines

 * similar to the more cmd line utility

 * */


#define STEP 20

#define LEN_BUFFER 256


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

  if(argc < 2){

    fprintf(stderr, "Usage: arg1 - file1 src, arg2 - file2 src, ... argn- filen src");

    exit(EXIT_FAILURE);

  }


  int in;

  char buffer[LEN_BUFFER];
```

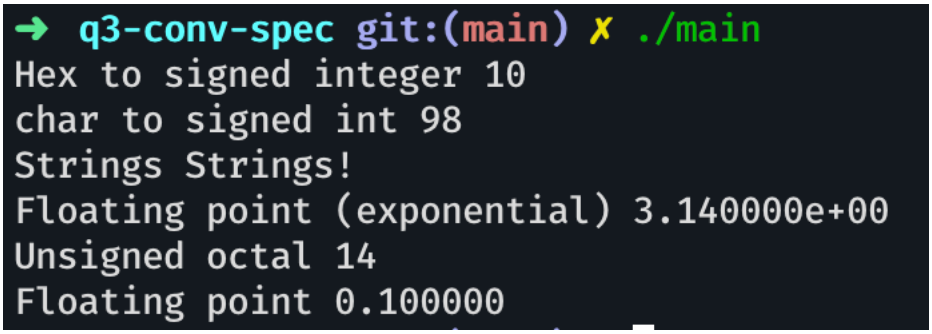
```
for(int i = 1; i < argc; i++){

  if((in = open(argv[i], O_RDONLY)) < 0){
    fprintf(stderr, "Error opening file %s", argv[i]);
    perror(" ");
    continue;
  }
  else printf("\n'%s' contents:\n", argv[i]);

  char ch;
  int line_count = 0;
  size_t len_read, char_count = 0;



  while((len_read = read(in, &ch, sizeof(char))) > 0){
    if(ch == '\n'){
      buffer[char_count] = '\0';
      printf("%d  %s\n", line_count, buffer);

      char_count = 0;
      line_count++;

      //Pause for user prompt to continue
      if(line_count % STEP == 0){
        printf("\n----------------------Press Enter to continue or q to stop----------------------");
        char ch = getchar();

        if(ch == '\r')
          continue;
        else if (ch == 'q')
          break;
```

```
      }
      memset(buffer, '\0', sizeof(buffer));
    }
    else {
      buffer[char_count++] = ch;
    }
  }

  line_count = 0;
  char_count = 0;
  close(in);
  printf("End of file\n");
 }

}
```



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

```
#include<stdio.h>

#include<unistd.h>

#include<string.h>

#include<stdlib.h>

#include<fcntl.h>

#include<sys/types.h>


int main()

{

        printf("Hex to signed integer %d\n", 0xa);

        printf("char to signed int %d\n", 'b');

        printf("Strings %s\n", "Strings!");

        printf("Floating point (exponential) %e\n", 3.14);

        printf("Unsigned octal %o\n", 12);

        printf("Floating point %f\n", 0.1 );

  return 0;

}
```



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

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/uio.h>
#include <sys/stat.h>
#include <unistd.h>
/*
 * Program to copy char by char from src to dst file
 * */

int main(int argc, char* argv[]){
  if(argc != 3){
    fprintf(stderr, "Usage: arg1 -- file1 src, arg2 -- file2 src");
    exit(EXIT_FAILURE);
  }

  int ch, in, out;

  if ((in = open(argv[1], O_RDONLY)) < 0){
    fprintf(stderr, "Error opening src file %s", argv[1]);
    perror(" ");
    exit(EXIT_FAILURE);
  }

  if ((out = open(argv[2], O_WRONLY | O_CREAT, S_IWUSR | S_IXOTH)) < 0){
    fprintf(stderr, "Error opening dst file %s", argv[2]);
    perror(" ");
```

```
    exit(EXIT_FAILURE);
  }

  size_t len_read, len_write;
  while((len_read = read(in, &ch, 1)) > 0){
    if ((len_write = write(out, &ch, 1)) < 0){
      fprintf(stderr, "Error writing to dst file");
      perror(" ");
    }
  }

  printf("Done copying!\n");
  return 0;
}
```