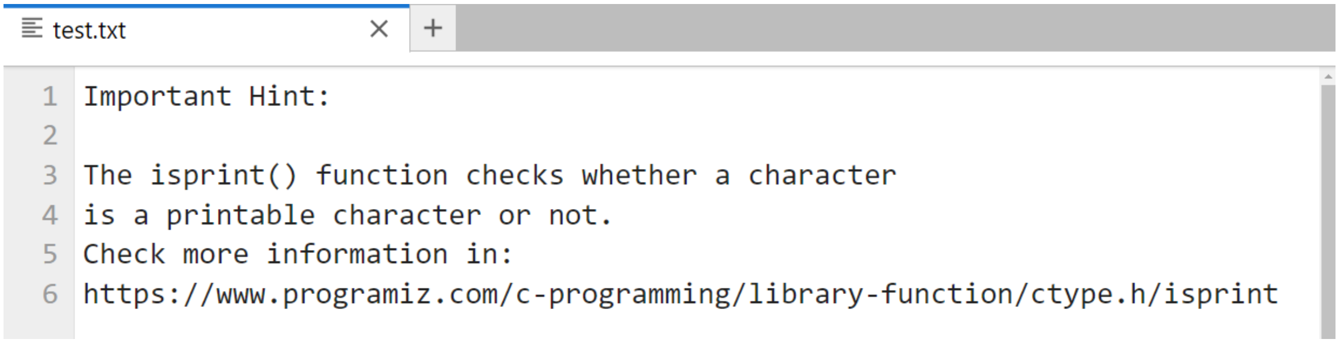
CSCI3240/6050 Project 1: Hex Dump

Introduction:

xxd command in Linux creates a hex dump of a given file or standard input. It's simple. For example, suppose you have a text file **'test.txt'** with contents as shown below:

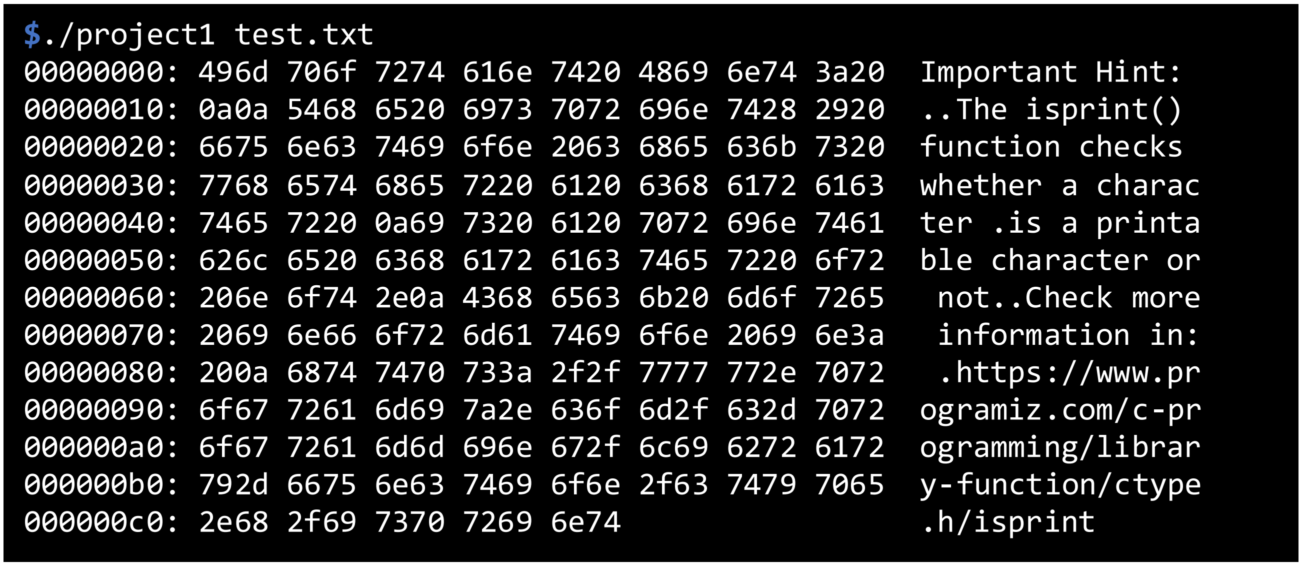


```
Important Hint:

The isprint() function checks whether a character
is a printable character or not.
Check more information in:
https://www.programiz.com/c-programming/library-function/ctype.h/isprint
```

Here's how you can use xxd to convert the file contents into hex:

```
$ xxd test.txt
```

And here's the output:



```
$./project1 test.txt
00000000: 496d 706f 7274 616e 7420 4869 6e74 3a20  Important Hint: 
00000010: 0a0a 5468 6520 6973 7072 696e 7428 2920  ..The isprint() 
00000020: 6675 6e63 7469 6f6e 2063 6865 636b 7320  function checks 
00000030: 7768 6574 6865 7220 6120 6368 6172 6163  whether a charac
00000040: 7465 7220 0a69 7320 6120 7072 696e 7461  ter .is a printa
00000050: 626c 6520 6368 6172 6163 7465 7220 6f72  ble character or
00000060: 206e 6f74 2e0a 4368 6563 6b20 6d6f 7265   not..Check more
00000070: 2069 6e66 6f72 6d61 7469 6f6e 2069 6e3a   information in:
00000080: 200a 6874 7470 733a 2f2f 7777 772e 7072   .https://www.pr
00000090: 6f67 7261 6d69 7a2e 636f 6d2f 632d 7072  ogramiz.com/c-pr
000000a0: 6f67 7261 6d6d 696e 672f 6c69 6272 6172  ogramming/librar
000000b0: 792d 6675 6e63 7469 6f6e 2f63 7479 7065  y-function/ctype
000000c0: 2e68 2f69 7370 7269 6e74                 .h/isprint
```

Hint: Notice that some unprintable characters such as newline characters, are converted into dots.

Problem:

Write a C program that takes a single command-line argument and produces the same output as the xxd program with the same argument. You must include meaningful comments on your code. You must name your program file as **project1.c.**

A sample run would look like this:

```
$ ./project1 test.txt
```

```
$ xxd test.txt
```

Note that the C function isprint(3) could prove quite useful here.

##The grader may test on multiple different files.

How to Test Correctness (Graders will use the same method):

Make sure you test your program using "**diff**" command. For example, to compare xxd's output with your program's output on project1.c file, you can do the following:

```
$ gcc project1.c -o project1

$ xxd project1.c > expetedOutput.txt

$ ./project1 project1.c > actualOutput.txt

$ diff expectedOutput.txt actualOutput.txt
```

The diff command is helpful in determining the difference between the two files. In case of **no difference**, the diff command will **not print anything**. Read more on: https://phoenixnap.com/kb/linux-diff

Steps to Create the Log File:

1. **Open your terminal and start the scripting process by typing:**

```
$ script Project1_Log.txt
```

2. **List all the files in the current directory:**

```
$ ls
```

3. **Compile your project 1:**

```
$ gcc project1.c -o project1
```

4. **Run your project 1:**

```
$ ./project1 test.txt
```

```
$ ./project1 project1.c
```

Note: You can test with multiple other input files here.

5. **Verify the correctness of your solution:**

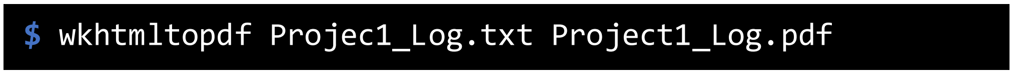
```
$ xxd project1.c > expetedOutput.txt

$ ./project1 project1.c > actualOutput.txt

$ diff expectedOutput.txt actualOutput.txt
```

Note: You can verify with multiple other input files here.

6. **Exit the scripting process to finish and save the log file:**

```
$ exit
```

7. **Convert the log file from txt to pdf using the txt created from using script**



```
$ wkhtmltopdf Projec1_Log.txt Project1_Log.pdf
```

The ‘`Project1_Log.pdf` ’ file will be generated in your current directory. Make sure this file is included in your submission.

Submission Instruction:

1. Submit the following file int Project1 Dropbox:
   - project1.c
   - Project1_Log.pdf
   - AI_Disclaimer.pdf
2. **Submission Due:** Check Lab 2 Dropbox

Grading Rubrics:

No.	Description	Deduction
1	Did not use command-line arguments	-70
2	Insightful comments missing	-5
3	Your program’s output does not match xxd’s output	-5 to -20
	a. Incorrect numbers of digits in counter	-5
	b. Incorrect space alignment (usually happens in the last line when there are less than 16 characters)	-10
	c. Unprintable characters not handled	-10
	d. Printing extra spaces or characters	-20
5	project1.c missing	-100
6	Output only contains plain characters with 16 characters in each line	-50
7	Output only contains hex codes for 16 characters in each line	-50
8	Filenames not named as requested	-30
9	AI Disclaimer missing	-100
10	Log file missing	-30
11	Segmentation fault	TBD

Useful Links:

1. Command-line Arguments:
   https://www.tutorialspoint.com/cprogramming/c_command_line_arguments.htm
2. How to save command output to a file in linux?:
   https://itsfoss.com/save-command-output-to-file-linux/
3. Isprint() function to find unprintable characters:
   https://www.programiz.com/c-programming/library-function/ctype.h/isprint
   4. Diff command to comparing two files in linux:
      https://www.tutorialspoint.com/how-to-use-diff-command-in-linux