C Programming lab: Vim, Linux and Control flow Quiz

Name: Shreyas Iyer

Email: Shreyas.Srinivasan@iiitb.org

Roll no: IMT2015018

Only one answer per question is correct. If more than one answer may apply, choose the best answer.

Concept: *cursor movement - up, down, left, right* **Concept:** *cursor movement - numeric arguments*

1. What command moves the cursor down one line?
   (a) `^^` or the down-arrow key
   (b) `d` or the down-arrow key
   (c) `h` or the down-arrow key
   (d) `j` or the down-arrow key
2. What does the command `j` do?
   (a) jumps to the bottom of the file
   (b) moves the cursor down a line
   (c) deletes a line
   (d) joins two lines
3. What command moves the cursor left one character?
   (a) `l` or the left-arrow key
   (b) `<` or the left-arrow key
   (c) `h` or the left-arrow key
   (d) `L` or the left-arrow key
4. What does the command `l` do?
   (a) jumps to the bottom of the file
   (b) copies a line
   (c) deletes one line
   (d) moves the cursor right one character
5. The command to move upwards/backwards a page is `<Ctl>-b`. What is the fastest way to move up 7 pages?
   (a) press `<Ctl>-b` then 7
   (b) press 7 and then `<Ctl>-b`
   (c) press `<Ctl>-b` 7 times in a row
   (d) press the up-arrow a sufficent number of times
6. The command to move right one word is `w`. What is the fastest way to move right 8 words?
   (a) press the `w` key 8 times
   (b) press 8 and then `w`

(c) press the right-arrow a sufficent number of times
(d) press `w` then 8

Concept: *cursor movement - larger movements*

7. What command moves the cursor to the end of a line?
   (a) `E`
   (b) `f`
   (c) `$`
   (d) `^`
8. What does the command `0 (zero)` do?
   (a) moves the cursor to the beginning of the line
   (b) moves the cursor to the beginning of the file
   (c) does nothing
   (d) opens up a line above the current line
9. What command moves the cursor to the last line in a file?
   (a) `B`
   (b) `b`
   (c) `G`
   (d) `f`
10. What does the command `G` do?
    (a) moves the cursor to the last character in the line
    (b) replaces the current letter with a *G*
    (c) goes to the first occurrence of the letter *G*
    (d) moves the cursor to the last line in the file
11. What command moves the cursor upwards/backwards one half page?
    (a) `<Ctl>-b`
    (b) `<Ctl>-t`
    (c) `<Ctl>-u`
    (d) `<Ctl>-p`
12. What does the command `<Ctl>-u` do?
    (a) goes to the first line in the file
    (b) moves the cursor up/backwards one half page
    (c) finds the first occurrence of the letter *u*
    (d) backs up the current file

Concept: *line numbers*

13. What command moves the cursor to line 10?
    (a) `L11`
    (b) `:10`
    (c) `:11`

(d) `L10`

14. What does the command `:33` do?

(a) moves the cursor to line 33
(b) deletes line 33
(c) moves the cursor to column 33
(d) moves the cursor to the 33^{rd} line from the bottom

Concept: *inserting before and after the cursor*

15. What command begins insertion immediately before the cursor?

(a) `i`
(b) `I`
(c) `A`
(d) `a`

16. What does the command `i` do?

(a) starts insert mode before the cursor
(b) starts insert mode at the beginning of the line
(c) moves the cursor to the first occurrence of the letter i
(d) moves the cursor to the beginning of the line

17. What command opens up a new line for insertion after the current line?

(a) `a`
(b) `o`
(c) `A`
(d) `A`

18. What does the command capital `O` do?

(a) opens up a new line before the current line (enters insert mode)
(b) starts insert mode at the beginning of the line
(c) starts insert mode at the beginning of the current word
(d) starts insert mode at the beginning of the file

19. What does the command `A` do?

(a) starts insert mode before the first occurance of the letter A
(b) opens up a new line after the current line (enters insert mode)
(c) starts insert mode at the end of the current word
(d) starts insert mode at the end of the line

20. What command begins insertion at the end of the current line?

(a) `^`
(b) `A`
(c) `0`
(d) `a`

21. What command ends insertion mode?

(a) `d`

(b) the `Esc` key

(c) `z`

(d) `q`

Concept: *pasting text before and after the cursor*

22. What command pastes deleted or copied text before the cursor?

    (a) `*`

    (b) `P`

    (c) `b`

    (d) `B`

23. What does the `p` command do?

    (a) pastes deleted or copied text at the end of the file

    (b) pastes deleted or copied text after the cursor

    (c) pastes deleted or copied text before the cursor

    (d) pastes deleted or copied text at the beginning of the line

Concept: *searching*

24. What command starts a forward search?

    (a) `s`

    (b) `S`

    (c) `f`

    (d) `/`

25. What command searches again in the same direction?

    (a) `a`

    (b) `f`

    (c) `B`

    (d) `n`

26. What command searches for *cat*?

    (a) `Gcat`

    (b) `/cat`

    (c) `fcat`

    (d) `gcat`

27. What command searches for words beginning with *the*?

    (a) `/\<the`

    (b) `fthe`

    (c) `/the`

    (d) `wthe`

28. What command searches for words ending with *the*?

    (a) `/the`

(b) `/the\>`

(c) `Wthe`

(d) `ethe`

Concept: *deletions*

29. What command deletes a character?

    (a) `r`

    (b) `d`

    (c) `D`

    (d) `x`

30. What command deletes a word?

    (a) `rw`

    (b) `dw`

    (c) `\#w`

    (d) `xw`

31. What command deletes a line?

    (a) `xx`

    (b) `xl`

    (c) `dl`

    (d) `dd`

32. What command deletes the current character plus the next two?

    (a) `3rc`

    (b) `3x`

    (c) `3D`

    (d) `3d`

Concept: *copying*

33. What command copies a word?

    (a) `dw`

    (b) `Cw`

    (c) `yw`

    (d) `cw`

34. What does the command `yw` do?

    (a) copies the line up to the first occurrence of a letter *w*

    (b) copies the current word

    (c) changes the current letter to a *w*

    (d) copies the current line

35. What command copies a line?

    (a) `dl`

    (b) `cl`

(c) `yl`

(d) `yy`

36. What does the command `yy` do?

    (a) copies the current word

    (b) copies the current line

    (c) copies the line up to the first occurrence of a letter y

    (d) changes the current letter to a y

37. What command copies the current word and the four following words?

    (a) `5cw`

    (b) `5dw`

    (c) `5yw`

    (d) `5Cw`

38. What does the command `4yw` do?

    (a) moves the cursor to line 4 and copies the new current word

    (b) copies the current word along with the next three words

    (c) changes the current word along with the three following words

    (d) moves the cursor down 4 lines and copies the new current word

39. What command copies the current line and the four following lines?

    (a) `5yl`

    (b) `5cl`

    (c) `5dl`

    (d) `5yy`

40. What does the command `4yy` do?

    (a) moves the cursor to column 4 and copies the current line

    (b) moves the cursor down 4 lines and copies the new current line

    (c) copies the current line and the next three lines

    (d) moves the cursor to line 4 and copies the new current line

Concept: *using marks*

41. What command marks a line with the letter z?

    (a) `'z`

    (b) `tz`

    (c) `nz`

    (d) `mz`

42. What does the command `mz` do?

    (a) marks the current line with the name z

    (b) replaces the current letter with z

    (c) swaps the current letter with the following letter

    (d) moves the cursor to the first occurrence of z

43. What command deletes all the lines between the current line and a line marked with the letter z?

(a) `x'z`
(b) `d,z`
(c) `x,z`
(d) `d'z`

Concept: *modifying characters, words, lines, and files* **Concept:** *repeating commands*

44. What is the fastest command to change one character into another?

(a) `r`
(b) `x`
(c) `c`
(d) `s`

45. What command changes a line?

(a) `ss`
(b) `cc`
(c) `cM`
(d) `c$`

46. What does the command `cc` do?

(a) changes the current line
(b) changes the current word
(c) changes the current character to the letter *w*

47. What command changes the rest of the line?

(a) `S`
(b) `ss`
(c) `cc`
(d) `C`

48. What does the command `C` do?

(a) changes the rest of the line
(b) copies the previous line
(c) copies the current line
(d) converts the current character to upper-case

49. What command repeats the last edit command?

(a) `!`
(b) `r`
(c) `R`
(d) `.`

50. The . command can take a numeric argument.

(a) true
(b) false

Concept: *absolute and relative paths*

51. If a path starts with `"/"`, it is:

    (a) an absolute path

    (b) a relative path

52. Which command does *not* reference an absolute path?

    (a) `cd ~/clab/project1`

    (b) `cd /home/ubuntu/clab`

    (c) `cd clab/project1`

53. Which command does *not* reference relative path?

    (a) `cd clab/project1`

    (b) `cd ~/clab/project1`

    (c) `cd ../home/ubuntu/clab`

54. If a path is an absolute path, it starts with:

    (a) `"/"` or `".."`

    (b) `".."` or `"~"`

    (c) `".."` or `"."`

    (d) `"/"` or `"~"`

55. If a path is a relative path, it does *not* start with:

    (a) `"/"` or `"~"`

    (b) `".."` or `"~"`

    (c) `"/"` or `".."`

    (d) `".."` or `"."`

56. Which command references a absolute path?

    (a) `mkdir ~/Downloads/temp`

    (b) `mkdir ./project1`

57. Which command references a absolute path?

    (a) `mkdir /tmp/project1/`

    (b) `mkdir clab/project1`

58. Generally speaking, any command that references a file name works with an absolute path to the file name.

    (a) False

    (b) True

Concept: *directory names - home, parent, root, current*

59. The directory `~` refers to:

    (a) your home directory

    (b) the root directory

    (c) the parent directory

    (d) the current directory

60. The directory `.` refers to:

    (a) the root directory

(b) the parent directory
(c) your home directory
(d) the current directory

61. The directory `..` refers to:

(a) the parent directory
(b) the current directory
(c) the root directory
(d) your home directory

62. The directory `/` refers to:

(a) the parent directory
(b) the root directory
(c) your home directory
(d) the current directory

Concept: *listing files and wildcards*

63. What does the command `ls ..` do?

(a) list files and directories in the parent directory
(b) list files and directories in the current directory
(c) list files and directories in the root directory
(d) list files and directories in your home directory

64. What does the command `ls *.s` do?

(a) list files and directories ending in `.s`
(b) list files and directories named `*.s`
(c) list files and directories starting with `*.s`
(d) list files and directories ending in `*.s`

65. What command lists the files ending with `.s`?

(a) `dir *.s`
(b) `ls .s`
(c) `dir .s`
(d) `ls *.s`

66. What command lists the files beginning with `abc`?

(a) `ls abc`
(b) `ls abc*`
(c) `dir abc`
(d) `dir abc*`

Concept: *directory and file creation*

67. What does the command `mkdir /oak` do?

(a) makes a directory named *oak* in the parent directory
(b) makes a directory named *oak* in the current directory

(c) makes a directory named *oak* in your home directory

(d) makes a directory named *oak* in the root directory

68. What command will make a directory named *maple* that hangs off the current directory?

(a) `mkdir ../maple`

(b) `mkdir /maple`

(c) `mkdir maple`

(d) `mkdir ~/maple`

69. What does the command `mkdir -p black/white/grey` do?

(a) makes a directory chain with *grey* as the highest directory

(b) makes a directory chain with *black* as the highest directory

70. What does the command `mkdir -p grey/black/white` do?

(a) makes a directory chain with *black* as the highest directory

(b) makes a directory chain with *grey* as the highest directory

71. What does the command `touch` do?

(a) displays the last time a file was read

(b) displays the last time a file was updated

(c) deletes a file

(d) creates a file if it doesn't already exist

72. What command can be used to create a file?

(a) `touch`

(b) `cat`

(c) `ls`

(d) `file`

73. What command is *not* generally used create a file?

(a) `touch`

(b) `vim`

(c) `ls`

Concept: *removing files and directories*

74. What command removes a file?

(a) `rf`

(b) `rm`

(c) `rem`

(d) `del`

75. What does the `rm` command do?

(a) makes room for a new file

(b) removes a file

(c) releases previously sent emails

(d) renames/moves a file

76. What does the `rm blue/*` command do?

(a) removes all files in the blue directory (but not the directory itself)
(b) removes the blue directory (but not the files in the directory)
(c) removes the blue directory and all files in the current directory
(d) removes the blue directory (and all the files in it)

77. What command can be used to remove an empty directory?

(a) `rmdir`
(b) `rdir`
(c) `xdir`
(d) `deldir`

78. What does the `rmdir temp` command do?

(a) removes a directory named *temp* (if it is empty)
(b) removes a file named *dir* and a file named *temp*
(c) removes a directory named *temp* (even if it is not empty)
(d) removes all the files in the directory *temp* (but not the directory itself)

79. What does the `rm dir temp` command do?

(a) removes a directory named *temp* (even if it is not empty)
(b) removes a file named *dir* and a file named *temp*
(c) removes all the files in the directory *temp* (but not the directory itself)
(d) removes a directory named *temp* (if it is empty)

80. What command can be used to remove a non-empty directory?

(a) `rmdir`
(b) `delall`
(c) `xdir -all`
(d) `rmdirfull`
(e) `rm -r`

81. What does the `rm -r` command do?

(a) removes a non-empty directory, including all the files within
(b) removes all the files in a directory (but not the directory itself)
(c) removes a directory, but only if it is empty
(d) removes the directory (but not the files within the directory)

82. What command can be used to remove all files ending in *.c*?

(a) `rm *.c`
(b) `rmdir ?.c`
(c) `rem *.c`
(d) `rmdir *.c`

83. What does the `rm abc*.c` command do?

(a) removes all files that both begin with *abc* and end with *.c*
(b) removes the file *abc* and all files ending with *.c*
(c) removes all files beginning with *abc* and a file named *.c*

(d) removes all files that begin with *abc* and all files that end with *.c*

84. What does the `rm abc *.c` command do?

    (a) removes all files beginning with *abc* and a file named *.c*

    (b) removes all files that both begin with *abc* and end with *.c*

    (c) removes the file *abc* and all files ending with *.c*

85. What does the `rm abc* .c` command do?

    (a) removes the file *abc* and all files ending with *.c*

    (b) removes all files that both begin with *abc* and end with *.c*

    (c) removes all files beginning with *abc* and a file named *.c*

Concept: *moving about the directory structure*

86. What command does *not* reliably change the current directory to your home directory?

    (a) `cd ~`

    (b) `cd ..`

    (c) `cd`

    (d) `cd ~/`

87. What command changes the current directory to the parent directory?

    (a) `cd ..`

    (b) `cd`

    (c) `cd /`

88. What does the command `cd /` do?

    (a) changes the current directory to the root directory

    (b) changes the current directory to the parent directory

    (c) changes the current directory to your home directory

    (d) changes the current directory to the current directory (a useless command)

89. What command changes the current directory to the *pink* directory that hangs off the parent directory?

    (a) `cd /pink`

    (b) `cd \pink`

    (c) `cd ~/pink`

    (d) `cd ../pink`

90. The command `cd ../pink` changes the current directory to the *pink* directory that hangs off the current directory.

    (a) True

    (b) False

91. The command `cd /pink` changes the current directory to the *pink* directory that hangs off the current directory.

    (a) True

    (b) False

92. The command `cd ~/pink` changes the current directory to the *pink* directory that hangs off the root directory.

    (a) False

(b) True

93. The command `cd pink` changes the current directory to the *pink* directory that hangs off the home directory.

(a) True

(b) False

Concept: *copying, moving and renaming files*

94. What command copies a file named *brown.c* to *tan.c*?

(a) `move brown.c tan.c`

(b) `cp brown.c tan.c`

(c) `ren brown.c tan.c`

(d) `mv brown.c tan.c`

95. The command `cp ../colors/red.c ~/green/` is equivalent to what series of commands?

(a) `cd ..;cd colors; cp red.c ../green/`

(b) `cd ~/colors/red.c; cp red.c ../green/`

(c) `cd ;cd colors; cp red.c ~/green/`

(d) `cd ..;cd colors; cp red.c ~/green/`

96. The command `mv ~/colors/red.c ~/green/` is equivalent to what series of commands?

(a) `cd /; cd colors; mv red.c ~/green/`

(b) `cd ~/colors; cp red.c ~/green/`

(c) `cd ~;cd colors; mv red.c ~/green/`

(d) `cd ;cd colors; cp red.c ~/green/`

97. What command moves a file named *brown.c* to the parent directory?

(a) `move brown.c ..`

(b) `mv brown.c ..`

(c) `ren brown.c ..`

98. What command copies a file named *brown.c* to the root directory and renames it *tan.c*?

(a) `coc brown.c ../tan.c`

(b) `cp brown.c /tan.c`

(c) `coc /brown.c tan.c`

(d) `cp brown.c ../tan.c`

(e) `cp /brown.c tan.c`

(f) `coc brown.c /tan.c`

99. What command moves a file named *brown.c* in the root directory to the current directory?

(a) `mv / brown.c .`

(b) `mv / ~/brown.c ..`

(c) `mv ~/ ./brown.c ~`

(d) `mv ../ brown.c .`

Concept: *verifying code*

100. Consider the problem statement: *sum the numbers from* a *to* b (*inclusive*). Does this logic compute the correct result?

```
int sum(int a, int b)
    {
    int total = 0;
    for (;a<b;a++)
        total = total + a;
    return total;
    }
```

(a) Yes

(b) No

101. Consider the problem statement: *sum the numbers from* a *to* b (*inclusive*). Does this logic compute the correct result?

```
int sum(int a, int b)
    {
    int total = 0;
    for (;a<=b;a++)
        total = total + a;
    return total;
    }
```

(a) No

(b) Yes

102. Consider the problem statement: *sum the numbers from* a *to* b (*inclusive*). Does this logic compute the correct result?

```
int sum(int a, int b)
    {
    int total = 0,i;
    for (i=a;i<=b;i++)
        total = total + i;
    return total;
    }
```

(a) No

(b) Yes

103. Consider the problem statement: *sum the numbers from* a *to* b (*inclusive*). Does this logic compute the correct result?

```
int sum(int a, int b)
    {
    int total = 0,i;
    for (i=a;i<b+1;i++)
        total = total + i;
    return total;
    }
```

(a) No

(b) Yes

Concept: *implementing loops*

104. Implement this loop: Product of the numbers from 1 to x, inclusive.

105. Implement this loop: Product of the numbers from a to b, inclusive.

106. Implement this loop: sum of the numbers from a to b, inclusive.

107. Implement this loop: sum of the numbers from 1 to x, inclusive.

108. Implement this loop: determine a^b by updating an accumulator via multiplication by a, b times.

109. Implement this loop: count the number of prime numbers from a to b inclusive. Here is a naive implementation of an *isPrime* logic, if you want to test your loop. We will assume that math.h header file is included. The option of math-library (-lm) is also assumed for compilation.

```
int isPrime(int n)
    {
    for (i=2;i<=(int)pow(n,0.5))
        if (n % i == 0)
            return 0;
    return 1;
    }
```

110. Implement this loop: count the number of numbers that are divisible by 2 or 3 from a to b inclusive.

Command line arguments: Verifying code

111. Consider the problem statement: *sum the numbers from* a *to* b (*inclusive*) using command line arguments. Does this logic compute the correct result?

```
int main(int argc, char *argv[])
    {
    int a, total = 0;
    for (a=atoi(argv[1])+1;a<atoi(argv[2]);a++)
        total = total + a;
    return 1;
    }
```

(a) Yes

(b) No

112. Consider the problem statement: *sum the numbers from* a *to* b (*inclusive*) using command line arguments. Does this logic compute the correct result?

```
int main(int argc, char *argv[])
    {
    int a, total = 0;
    for (a=atoi(argv[1]);a<=atoi(argv[2]);a++)
        total = total + a;
    return 1;
    }
```

(a) Yes

(b) No

113. Consider the problem statement: *sum the numbers from* a *to* b (*inclusive*) using command line arguments. Does this logic compute the correct result?

```
int main(int argc, char *argv[])
    {
    int a, total = 0;
    for (a=atoi(argv[1])-1;a<atoi(argv[2]);a++)
        total = total + a;
    return 1;
    }
```

(a) No

(b) Yes

114. Implement this loop: total up the product of the numbers from 1 to x where 'x' is provided as command line arguments (argv[1]), inclusive. (Show complete program)

115. Implement this loop: total up the product of the numbers from a to b, inclusive where 'a' and 'b' are provided as command line arguments (argv[1], argv[2]). (Show complete program)

116. Implement this loop: total up the sum of the numbers from a to b, inclusive. where 'a' and 'b' are provided as command line arguments. (Show complete program)

117. Implement this loop: total up the sum of the numbers from 1 to x, inclusive. where 'x' is provided in the command line argument. (Show complete program)

Activity problems

118. Implement a calculator program that takes three command line arguments (excluding the program file name) and performs a simple calculation as shown below:

```
./calc 2 + 3

./calc 5 / 2

./calc 5 - 6

./calc 4 \* 6
```

(Show complete program)

119. Implement a program which will take three arguments (intial final step) as shown below and prints the numbers over step-size. Use loop in this task. For example:

```
./loop 3 10 2
should show the following output.
3 5 7 9
```

(show complete program)

120. Implement a program which does eight modes of arithmatic logic unit (ALU) operations: multiplexing, demultiplexing, encoding, decoding, direct-memory-access, random-memory-access, updating-cache, and recent-memory-search. All eight modes are controlled by eight bits. Design an ALU such that when user specifies an option in command line argument, the program should indicate the specific operation. The users can specify multiple operations by passing the bits in the command line argument. In the multiple operations scheme, we will assume that sequential operations are performed in the ALU. Use bitwise operations to display the operations selected by the user. Passing a Zero (0) option, should display entire menu of ALU as shown below:

```
 ./alu 0
Select menu
0x01: multiplexer
0x02: demultiplexer
```

```
0x04: encoding
0x08: decoding
0x10: dma
0x20: rma
0x40: updating-cache
0x80: recent-memory-search
```

Passing 1 option, should display "multiplexer" as shown below:

```
./alu 1
Multiplexer selected
```

Passing 2 option, should display "Demultiplexer" as shown below:

```
./alu 2
Demultiplexer selected
```

Passing 3 option, should display two operations as shown below:

```
./alu 3
Demultiplexer selected
Multiplexer selected
```

Passing 255 option, should display all the operations.