

Agenda / Learning Objectives:

1. Extract lab15.tar in your venus account after running the following command (note the dot at the end):
`cp ~ctse/cs111/lab15.tar .`
2. Unix:
Apply what we learn from ryanstutorials.net to:
 - a. List out files ending with cpp
 - b. List out files with fa18 prefix
 - c. Create a directory called **tmp** and move all the files with cpp file extension to tmp folder
 - d. Move all the files in tmp folder back to its parent directory
 - e. Save the console output of **cal 2018** to a file called 2018Calendar.txt
 - f. Send the output of **ls** as input of **wc** ([Piping](#))
3. Vim text editor:
Learn how to:
 - a. Join two lines
 - b. Search and replace a text for specified lines
4. Practice using **rand** and **sqrt** functions

Lab work – rand and sqrt functions

1) Write a complete C++ program which carries out the following tasks:

- Ask the user for two integers, a and b .
- If a is greater than b , exit the program.
- Print a random number (using the *rand* function) between a and b , inclusive. The formula is as follows:

$$\text{rand()} \% \underbrace{(b - a + 1)} + a$$

Amount of numbers from a to b .

Note: Using the same inputs, run it a few times with *srand()*. Comment out the *srand* statement, compile the code and run the program again for a few times. Notice the difference.

2) More practice on using **rand()**: Print the following numbers to the screen:

- A random single digit between 0 and 9
- 10 random numbers (between 1 and 10) separated with a space on one line
- A random number in the range 1000 and 9999 (inclusive)
- 5 random numbers between -1 and -9 on one output line
- 5 random 3-digit integers (one number per line)
- a random number r with $7 \leq r \leq 27$

(harder questions from MT1)

- a random 3-digit even number
- a random 3-digit number that is divisible by 3

3) (prac1.pdf) Write a complete C++ program that does the following.

- It repeatedly, asks the user to enter an integer.
- If the entered number is negative, the word “Negative” is printed and the program terminates.
- Otherwise the square root of the number is calculated and the **nearest integer to this square root** is printed.

Here is an example of how the program should work:

Enter a positive integer n (or a neg number to exit):

100

Nearest integer to its square root = 10

Enter a positive integer n (or a neg number to exit): **97**

Nearest integer to its square root = 10

Enter a positive integer n (or a neg number to exit): **101**

Nearest integer to its square root = 10

Enter a positive integer n (or a neg number to exit): **-100**

Negative

Answers for Unix command questions in part 2 on page 1:

a) `ls *.cpp`

b) `ls fa18*`

c) `mkdir tmp; mv *.cpp tmp/`

d) `cd tmp; mv * .. OR mv tmp/* .`

e) `cal 2018 > 2018Calendar.txt`

f) `ls | wc`

References for vi commands in part 3:

a) Join command – page 23 in [vim book - vimbook-OPL.pdf](#)

b) Search and replace - [Search and replace | Vim Tips Wiki | FANDOM powered by Wikia](#)