COL100 Lecture 26

Note Title 05-11-2018

Review: Recursion

- identify self-similarity and base-case
- ensure progress

.eg.

```
int power(int x, int n)
{
    if (n == 0) {
            return 1;
    } else if (n % 2 == 0) {
            return power(x * x, n/2);
    } else {
        return x * power(x, n-1);
    }
}
```

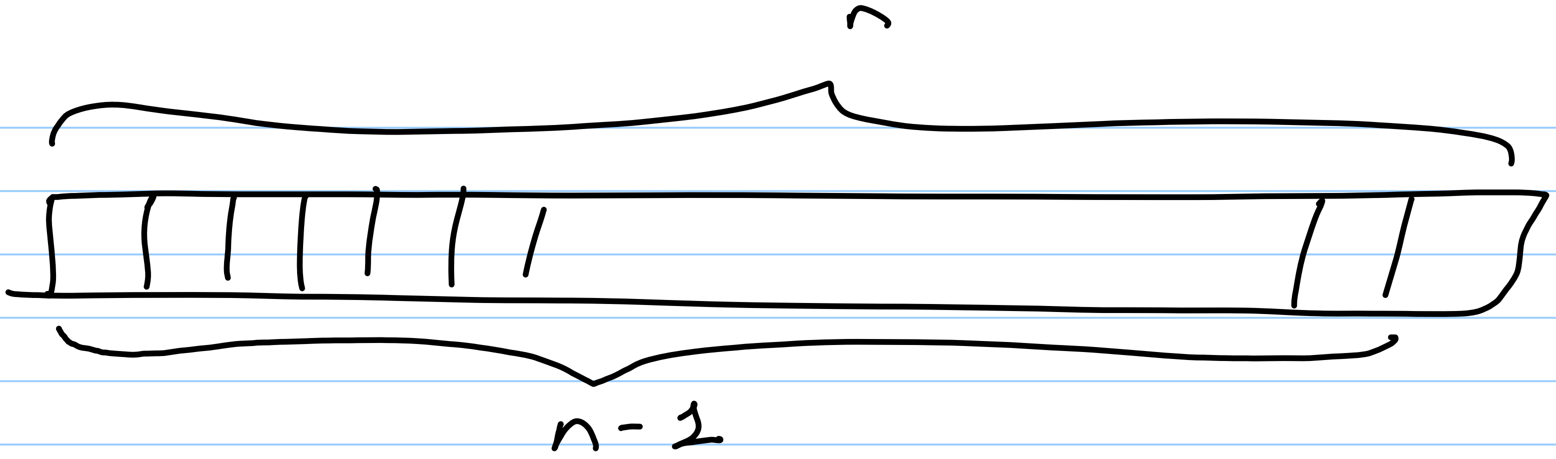
1	0	1	0	1	0
2^5	2^4	2^3	2^2	2^1	2^0

s= "101010" , output = 42

```
int     convertFromBinary ( string     s)
{
          int    total = 0;      int  place_value  = 1;
          for ( int   i =    s. length() - 1 ;   i >= 0  ;  i - -)
          {
                    char     ch =   s. at(i) ;
                    if     ( ch  ==   '0')    {
                              // do  nothing
                    }  else    {
                              total = total +  place_value ;
                    }
                    place_value  =  place_value   *  2;

          }
          return   total;
}
```

Recursive



$$\text{value}(s_{\underline{1..n}}) = \text{value}(s_{1..n-1}) * 2 + \text{value}(s_{n,n})$$

$$\text{value}(1010|1) = \underline{\text{value}(1010)} * 2 + \text{value}(1)$$

$$= (\text{value}(101) * 2 + \text{value}(0)) * 2 + 1$$

```
                // assume  binary.length() >= 1

int      convertFromBinary ( string    binary )
{           int    n =    binary.length() ;
             if     ( n == 1 )        {       // base case
                     if  (binary.at(0) == '0')  {
                                         return 0;
                     } else   return 1;
             }   else        {          // recursive    case
                  return    convertFromBinary (binary.substr(0, n-1))
                                                                * 2
                        +   convert From Binary (binary.substr(n-1, 1))
            }
}
```

Given a file, print its lines in reverse order

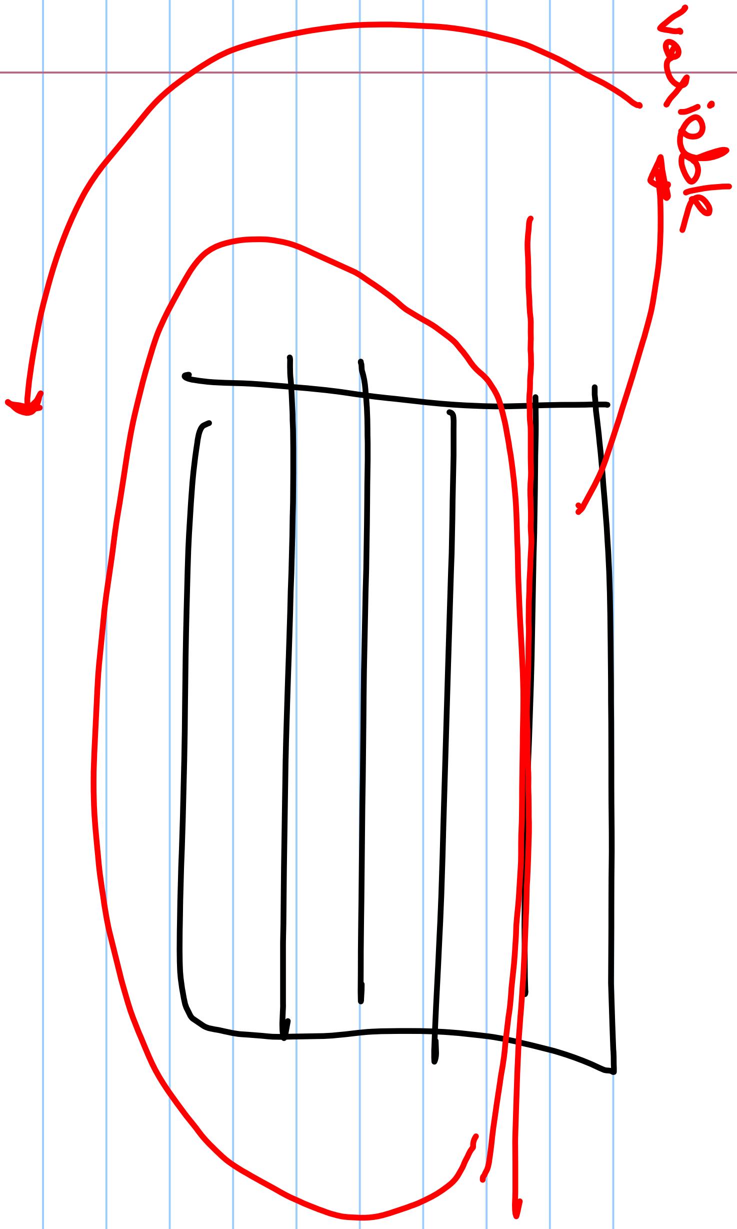
in.txt:

```
Hello world
Hello foo
bar bar
baz hello
```

---

expected output:

```
baz hello
bar bar
Hello foo
Hello world
```


variable

Recursive

base case : empty file

```
void reverseLines( ifstream & input)
{
        string line;
        if (getline (input, line))
        {
                reverseLines (input);
                cout << line << endl;
        } else {}
}
int main() {
        ifstream input ("in.txt");
        reverseLines (input);
        return 0;
}
```

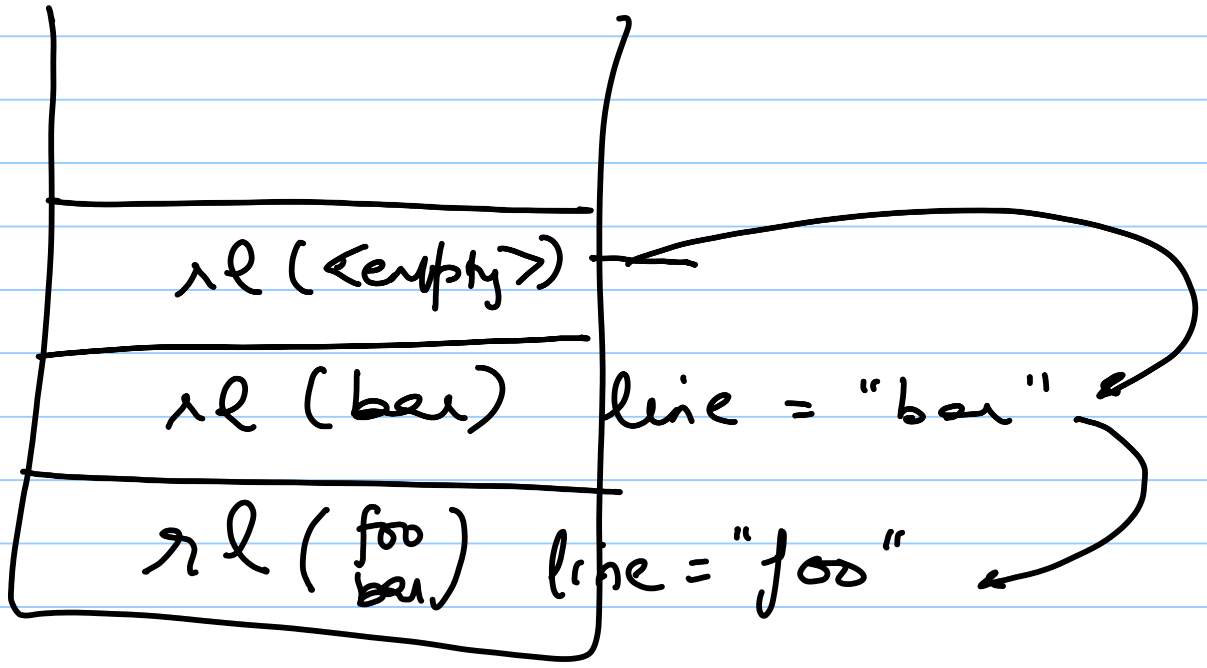
foo
bar

rl

= rl(bar)
cout << foo

rl (<empty>) → do nothing
cout << bar

cout << foo


rl(<empty>)
rl(bar)
line = "bar"
rl(foo bar)
line = "foo"

crawl (filename)

prints information about file

- if filename represents a normal file, just print it's name

- if filename represent a directory, print its name AND information about every file/directory inside it with indentation

```
courses
␣␣␣␣col100
    ␣␣␣␣lab2
        ␣␣␣␣hello_world.cpp
            order_of_evaluation.cpp
        lab3
            if-then-else.cpp
        minor1.pdf
    eel100
```

Helper functions "filelib.h"

bool isDirectory ("name")	returns true if "name" represent a directory stat()
listDirectory ("name")	returns a Vector<string> with the names of the files in the given directory "name"

```
isDirectory ("courses")        :    true
isDirectory (" courses/ col100") :   true
listDirectory (" courses"): [        "col100",
                                     "eel100"]
listDirectory ("courses/col100"):
        [                     "lab2",
                              "lab3" ,
                              " minor1.pdf "]
```

Self-similarity?

printing information about directory "name" at indentation level L
involves
printing information about
files inside "name" at L+4
or
directories

Base case: normal file
(which is not a
directory)

```
void crawl (string filename, string indent)
{
        cout << indent << getTail(filename)<< endl;
    if  (isDirectory(filename))
    {
        Vector <string> fileList;
        listDirectory(filename, fileList);
        for (string subfile : fileList)
        {
            crawl( filename+"/"+ subfile,
                   indent + "␣␣␣␣");
        }
    }
}
```

pass by value

crawl("courses", ""):

crawl("courses/col200", "␣␣␣␣")

crawl ("courses/col100/ lab2", "␣␣␣␣␣␣␣␣");

crawl("courses/col100/lab2/hello_world.cpp", " 12 spaces");

crawl ("courses/col100/lab2/order_of_end.cpp", " 12 speces");

↓

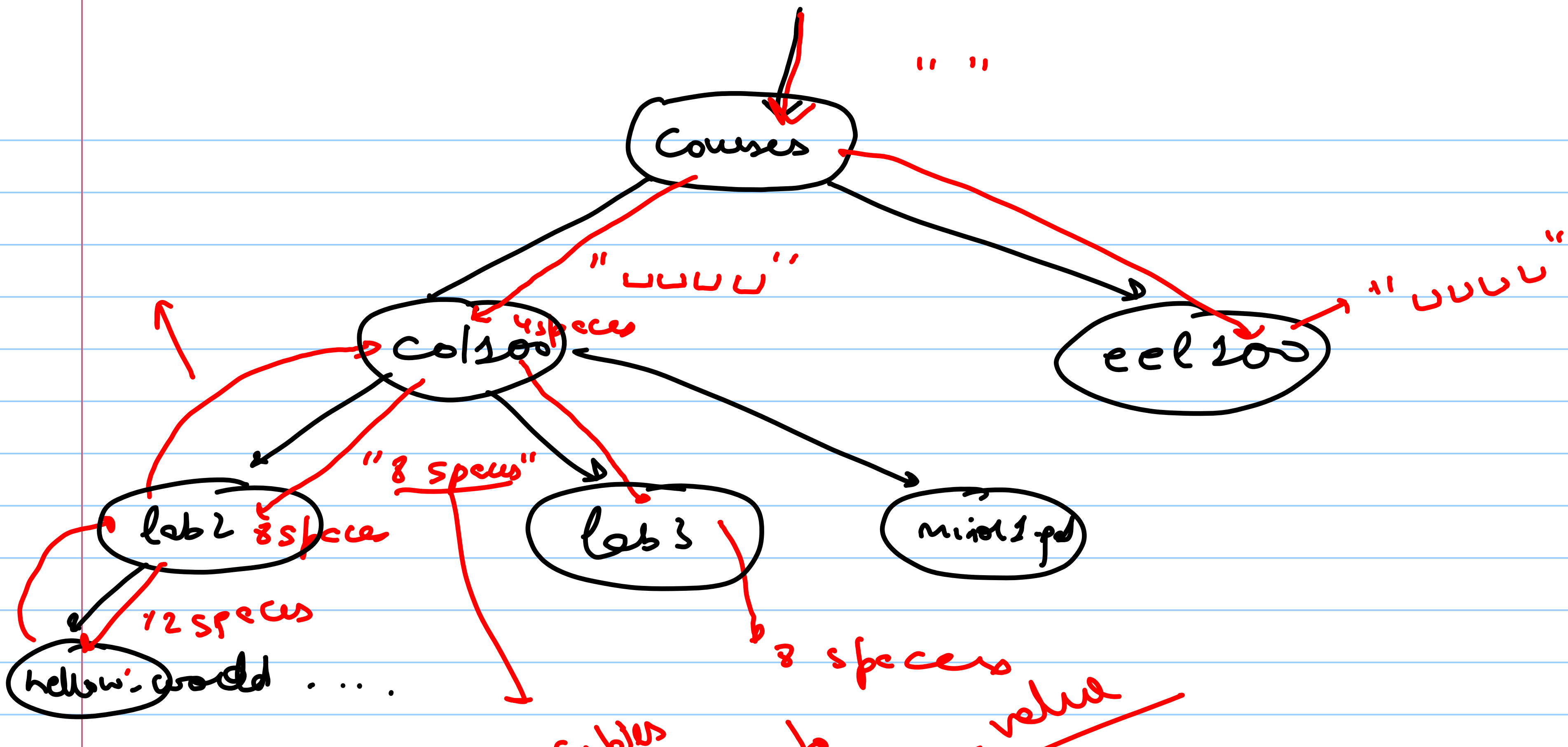
crawl ("courses/col100/ lab3", " 8 spaces");

```
getTail("courses/col100/minor1.pdf") : "minor1.pdf
```

```
int     main()
{
                                        empty string
        crawl(" courses" ,   "");

        return 0;

  }
```



copies due to pass-by-value