Problem 1

Yufei Yin

Question 1a, (2 points):

Write R code to download the course outline website for this year's offering of this course and extract all h3 headings remove all of the HTML formatting and any excess white space (leading and trailing white space and also repeated whitespace characters) from all h3 headings, and then print out those headings. Provide the R code.

```
# url
course_url ="https://www.sfu.ca/outlines.html?2021/spring/stat/240/d100"
# web page
course_page = readLines(course_url)
# <h3> tag index
grep('<h3', course_page)
```

```
## [1] 216 218
```

```
# <h3> contents
course_page[grep('<h3', course_page)]
```

```
## [1] "                    <h3 id=\"class-number\">Class Number: 3323</h3>"
## [2] "                    <h3 id=\"delivery-method\">Delivery Method: In Person</h3>"
```

```
# remove <...>
gsub("<.*?>","",course_page[grep('<h3', course_page)])
```

```
## [1] "                    Class Number: 3323"
## [2] "                    Delivery Method: In Person"
```

```
# remove leading and trailing space
trimws(gsub("<.*?>","",course_page[grep('<h3', course_page)]))
```

```
## [1] "Class Number: 3323"         "Delivery Method: In Person"
```

```
# remove repeated whitespace if there are any
gsub("/w+"," ",trimws(gsub("<.*?>","",course_page[grep('<h3', course_page)])))
```

```
## [1] "Class Number: 3323"         "Delivery Method: In Person"
```

Question 1b, (2 points):

Extract the course code from the text of the website https://www.sfu.ca/outlines.html?2021/spring/stat/240/d100, and provide the R code. Argue that the same code works on the pages for the outlines of other courses (or, modify that your code so that it does).

```
course_name = function(url) {
  course_page = readLines(url)
  # <h1> contents
  s = grep('<h1', course_page, value = T)
  # match
  m = regexec('- (.*?) <', s)
  s1 = regmatches(s, m)
  name = s1[[2]][2]
  return(name)
}
```

Test

```
STAT300W = 'http://www.sfu.ca/outlines.html?2021/summer/stat/300w/d100'
course_name(STAT300W)
```

```
## [1] "STAT 300W"
```

```
ARCH301 = 'http://www.sfu.ca/outlines.html?2021/summer/arch/301/c100'
course_name(ARCH301)
```

```
## [1] "ARCH 301"
```

```
STAT240 = 'https://www.sfu.ca/outlines.html?2021/spring/stat/240/d100'
course_name(STAT240)
```

```
## [1] "STAT 240"
```

```
CA104 = 'http://www.sfu.ca/outlines.html?2021/summer/ca/104/o101'
course_name(CA104)
```

```
## [1] "CA 104"
```

Question 1c, (6 points):

Write an R function called course. This function should take as an argument a string specifying the URL of a course outline, and return a list. The list should have two elements, one with name course and value given by the course code, and one with name instructor and value given by the name of the instructor of the course (with all extreneous whitespace removed). Demonstrate that this code works on a few URLs and provide the code and the demonstration.

```
library(stringr)
course = function(url){
  course_list = list(course = NA,
                     instructor = NA)

  # course name (function in quesiont 1b)
  course_list$course = course_name(url)

  # instructor name
  # web page
  course_page = readLines(url)

  # <h4> index
  head_index = str_which(course_page, '<h4')

  # <h4> contents
  s1 = course_page[str_detect(course_page, '<h4')]

  # match the index that contains instructor
  match = which(!is.na(str_match(s1, "Instructor")))

  # the case we did not match anything
  if (length(match) == 0){
    course_list$instructor = "Instructor name is not indicated in the web page"
  }
  else{
  # the contents before and after <h4> instructor </h4>
  s2 = course_page[(head_index[match]-1):(head_index[match]+1)]
  # remove <...> and leading and tailing space
  s3 = trimws(str_remove_all(s2,"<.*?>"))
  course_list$instructor =  s3[length(s3)]
  }
  return(course_list)
}
```

Test

```
course(STAT300W)
```

```
## $course
## [1] "STAT 300W"
##
## $instructor
## [1] "Michael Davis"
```

```
course(ARCH301)
```

```
## $course
## [1] "ARCH 301"
##
## $instructor
## [1] "Instructor name is not indicated in the web page"
```

```
course(STAT240)
```

```
## $course
## [1] "STAT 240"
##
## $instructor
## [1] "Lloyd Elliott"
```

```
course(CA104)
```

```
## $course
## [1] "CA 104"
##
## $instructor
## [1] "Sessional"
```