

Programming

Midterm Exam

10:10-12:00, November 29, 2019

Student ID & Name:

1. (15%) **True or False.** Please specify if each of following statement is true. If it is false, please modify the statement or the code.

(a) Type the following codes.

```
x <- c(7,12,6,20,9)
sort(x)
```

The result is

```
## [1] 20 12 9 7 6
```

(b) Type the following codes.

```
x <- c("Y", "N", "Y", "N", "N")
y <- factor(x)
```

Then, the number of levels is 5.

(c) Letting **x.df** is a data frame, then **names(x.df)** and **colnames(x.df)** are the same.

(d) Type the following codes.

```
x.names <- c("John", "Jack")
x.sex <- c("M", "M")
x.age <- c(20, 30)
X <- data.frame(x.names, x.sex, x.age)
X[2, 1]
```

The result is

```
## [1] "Jack"
```

(e) Letting **x** be a numeric vector, the following two codes are the same.

code 1

```
y <- c()
for(j in 1:length(x)){
  if(x[j] > 120){
    y <- c(y, x[j]*3)
  }else{
    y <- c(y, x[j])
  }
}
```

code 2

```
y <- ifelse(x > 120, x*3, x)
```

Stage

Isla	Silas	Olivia	Theodore	Posie	Jack
Finn	Aurora	Aarav	Cora	Felix	Ada
Maeve	Henry	Amara	Wyatt	Charlotte	Aryan
Oscar	Amelia	John	Ophelia	Leo	Ava
Rose	Julian	Genevieve	Levi	Luna	Ethan

Figure 1: Seat Matrix

2. (15%) Following the figure, please specify the following questions.

- (a) Construct a matrix to show the names of seats in a classroom, which is shown in the figure. The name of the matrix is *class*.
- (b) What is the name if you type `class[8]`.
- (c) How to use two dimensional way for a matrix to get the same result as (b)?
- (d) What is the reason that (b) is equal to (c)?
- (e) John would like to change the seat with Jack. Use code to show the change of the matrix.
- (f) There are six new students, but you do not know their names. You make labels for the six students as (S1, S2, S3, S4, S5, S6) and assign them to be seated at the last row. Write a code to show this result.
- (g) After the class, you knows their names: - S3: Kelly - S4: Ben - S5: Susan - S6: Jackson - S1: Helen - S2: Owen. Modify the seat matrix in (f).

3. (15%) **Comparison.** Please describe the output of the codes and the difference between the two codes. Use examples for explanation.

- (a) Let x and y be arbitrary numbers and $x < y$.

```
seq(x, y, by = 2)
seq(x, y, length.out = 2)
```

- (b) Letting x be a set of data, evaluate the average of the data.

```
mean(x)
mean(x, trim = 0.2)
```

- (c) Letting x is a vector, what is the difference of the following codes?

```
is.character(x)
as.character(x)
```

4. (20%) A pair of fair dice are to be continually rolled until all the possible outcomes 2, 3, . . . , 12 have occurred at least once. Develop a function to collect the total number of rolls. Conduct the code for 10000 times and give the average of the 10000 rolls. Is it possible that the total number of rolls is 11? How many times do you get the result that the number of rolls is 11? (You should use the following two functions.)

```
sample(...)
function(...)
```

5. (25%) Read the file from **reliability_training.csv**. There are two columns including unit ID, the measurement time (in cycle). The last measurement time is the lifetime of the unit. If the unit ID is “**F**”**x**, it means the unit fails. If the unit ID is “**C**”**x**, it means the unit is still working. Please specify the following questions.

- (a) How many failures and how many units which is not fail are there?
- (b) Write a code to collect the lifetime for all units.
- (c) What is the average lifetime over all failures?
- (d) Write a function to collect the above information so that you just set the input file and the results in (a) and (b) are shown automatically, which is printed by the following format.

```
Number of failures:
Number of working device:
The average of the failure:
```

- (e) Try to use the function in (d) to show the information from **reliability_testing.csv**.
6. (10%) Construct functions for (a) and (b) to provide the information of a vector, including **min**, **max**, **average**, **order** simultaneously.

- (a) The output is like

```
## min: 3
## max:10
## average:6.4
## order: 25413
```

- (b) The output is like

```
## $min
## [1] 2
##
## $max
## [1] 10
##
## $mean
## [1] 6.4
##
## $order
## [1] 2 3 1 5 4
```