

```

---
title: "Homework_1"
author: "LekhShetty"
format: pdf
output: pdf
editor: visual
---

## Question 2

### Part 1

```{r}
my_vec <- c(
 "+0.07",
 "-0.07",
 "+0.25",
 "-0.84",
 "+0.32",
 "-0.24",
 "-0.97",
 "-0.36",
 "+1.76",
 "-0.36"
)
typeof(my_vec)
"The data types of the elements in the vector are characters"
```

### Part 2

```{r}
my_vec_double <- as.double(my_vec)
my_vec_double

my_vec_integer <- as.integer(my_vec)
my_vec_integer
```

### Part 3

```{r}
my_vec_bool <- sum((ifelse(my_vec_double <= 0, TRUE, FALSE))==FALSE)
my_vec_bool
```

### Part 4

```{r}
my_vec_double <- sort(my_vec_double)
my_vec_double
```

## Question 3

```

```
### Part 1
```

```
```{r}
matrix_1 = matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9), nrow=3, ncol=3, byrow=TRUE)
matrix_1
```

```
row1 <- matrix(c(1:100), nrow=1, ncol=100)
row2 <- matrix(c((1:100)**2), nrow=1, ncol=100)
matrix_2 <- rbind(row1, row2)
matrix_2
```

```
```
```

```
### Part 2
```

```
```{r}
row_wise_scan <- function(x){
 n <- nrow(x)
 m <- ncol(x)

 # Insert your code here
 count <- 0
 for(i in 1:n){
 for(j in 1:m){
 if(x[i,j] >= 0){
 count <- count + 1
 }
 }
 }

 return(count)
}

matrix_1 = matrix(c(1, -1, 3, 4, 5, 6, 7, 8, 9), nrow=3, ncol=3, byrow=TRUE)
row_wise_scan(matrix_1)
```

```
#my_vec_bool <- sum((ifelse(my_vec_double <= 0, TRUE, FALSE))==FALSE)
```

```
```
```

```
### Part 3
```

```
```{r}
col_wise_scan <- function(x){
 count <- 0

 # Insert your code here
 n <- nrow(x)
 m <- ncol(x)
 for(j in 1:m){
 for(i in 1:n){
 if(x[i,j] >= 0){
 count <- count + 1
 }
 }
 }
}
```

```

 }
}

 return(count)
}

matrix_1 = matrix(c(1, -1, 3, 4, 5, 6, 7, 8, 9), nrow=3, ncol=3, byrow=TRUE)
col_wise_scan(matrix_1)
```

### Part 4

```{r}
"col_wise_scan and row_wise_scan will have similar runtime. Both codes
traverse through each element in m and in n in a nested for loop."
```

### Part 5

```{r}
time_scan <- function(f, M){
 initial_time <- Sys.time() # Write your code here
 f(M)
 final_time <- Sys.time() # Write your code here

 total_time_taken <- final_time - initial_time
 return(total_time_taken)
}

M = matrix(c((1:100)**2), nrow=1, ncol=100)

list(
 row_wise_time = time_scan(row_wise_scan, M),
 col_wise_time = time_scan(col_wise_scan, M)
)
```

### Part 6

```{r}
M<-matrix(rnorm(25) , nrow = 100, ncol=100)

list(
 row_wise_time = time_scan(row_wise_scan, M),
 col_wise_time = time_scan(col_wise_scan, M)
)
```

```{r}
M<-matrix(rnorm(25) , nrow = 1000, ncol=1000)

list(

```

```
 row_wise_time = time_scan(row_wise_scan, M),
 col_wise_time = time_scan(col_wise_scan, M)
)
```

```
```\n
```

```
```\n{r}
```

```
M<-matrix(rnorm(25) , nrow = 5000, ncol=5000)
```

```
list(
 row_wise_time = time_scan(row_wise_scan, M),
 col_wise_time = time_scan(col_wise_scan, M)
)
```

```
"seems like as the number of rows and columns increase, the runtime of both
algorithms takes longer as it has to traverse through more elements"
```\n
```