Practical No. 8 Date: / / 201

Aim: Program to perform list operations on Student Information.

Objectives:

- To study R List & its operations.
- Implement a program to perform operations on Student Information.

Theory:

R List

List is R object which can store different types of R objects. It may contain Vector, Numeric, Character, Matrix, Factor, Data Frame, and even another List. Unlike vectors or matrices, no coercion is performed on the elements of a list. Due to this, list doesn't have a predefined structure and it results in loss of some basic functionality offered by vector/matrices. A list is created using list() function.

Creating List

Syntax

```
list(list_elements)
```

- list_elements: vector/matrix/factor/data-frame/list etc.

Example

```
student_vec <- c(1, "ABC", "SYCOMP")
student_lst <- list(1, "ABC", "SYCOMP")
student_vec
[1] "1" "ABC" "SYCOMP"            # Coersion performed on numeric element
student_lst
[[1]]
[1] 1
[[2]]
[1] "ABC"
[[3]]
[1] "SYCOMP"                      # No Coersion performed
```

Naming List

Using names() Function

```
student_lst <- list(1, "ABC", "SYCOMP")
names(student_lst) <- c("RollNo", "Name", "Class")
student_lst
```

```
$RollNo
[1] 1
$Name
[1] "ABC"
$Class
[1] "SYCOMP"
```

Using list() Function

```
student_lst <- list(RollNo = 1, Name = "ABC", Class = "SYCOMP")
student_lst
$RollNo
[1] 1
$Name
[1] "ABC"
$Class
[1] "SYCOMP"
str(student_lst)              # Check Structure of List
List of 3
$ RollNo: num 1
$ Name  : chr "ABC"
$ Class : chr "SYCOMP"
```

Accessing List

Like any R object lists are also accessed by index or by name of element. Lists have different structure so as access methods are, here

- To access single list element (subset list) "[[" or "$" is used.
- To create sublist from list "[" is used.

Using [to Create Sublist

```
marks_lst <- list(Sub1 = 45, Sub2 = 54, Sub3 = 60)
student_lst <- list(RollNo = 1, Name = "ABC", Class = "SYCOMP",Marks = marks_lst)

student_lst[2]                # Sublist with single element
$Names
[1] "ABC"

student_lst[c(1,3)]           # Sublist with multiple elements
$RollNo
1
$Class
[1] "SYCOMP"
```

Using [[to Subset List

```
marks_lst <- list(Sub1 = 45, Sub2 = 54, Sub3 = 60)
student_lst <- list(RollNo = 1, Name = "ABC", Class = "SYCOMP", Marks = marks_lst)
student_lst[[2]]                    # Subsetting element from main list
[1] "ABC"

student_lst[[4]][[1]]
student_lst[[c(4,1)]]               # Subsetting element from inner list
[1] 45

student_lst[[1]][[3]]               # Subsetting multiple elements gives error
student_lst[[c(1,3)]]
Error in student[[1]][[3]] : subscript out of bounds
```

Using Names with [[

```
marks_lst <- list(Sub1 = 45, Sub2 = 54, Sub3 = 60)
student_lst <- list(RollNo = 1, Name = "ABC", Class = "SYCOMP", Marks = marks_lst)
student_lst[["RollNo"]]             # Subsetting element from main list
[1] 1

student_lst[["Marks"]][["Sub2"]]
student_lst[["Marks"]][[2]]
student_lst[[4]][["Sub2"]]
student_lst[[c("Marks","Sub2")]]    # Subsetting element from inner list
[1] 54
```

Using Names with $

```
marks_lst <- list(Sub1 = 45, Sub2 = 54, Sub3 = 60)
student_lst <- list(RollNo = 1, Name = "ABC", Class = "SYCOMP", Marks = marks_lst)
student_lst$Class                   # Subsetting element from main list
[1] "SYCOMP"

student_lst$Marks$Sub3
student_lst$Marks[[3]]              # Subsetting element from inner list
[1] 60
```

Using Names with [(Creating Sublist)

```
marks_lst <- list(Sub1 = 45, Sub2 = 54, Sub3 = 60)
student_lst <- list(RollNo = 1, Name = "ABC", Class = "SYCOMP", Marks = marks_lst)
```

```
student_lst["RollNo"]              # Sublist with single element
$RollNo
[1] 1

student_lst[c("Marks","Sub2")]     # Sublist with multiple element
$RollNo
[1] 1
$Name
[1] "ABC"
```

Using Logicals

```
marks_lst <- list(Sub1 = 45, Sub2 = 54, Sub3 = 60)
student_lst <- list(RollNo = 1, Name = "ABC", Class = "SYCOMP", Marks = marks_lst)
student_lst[c(T,F,F,F)]            # Sublist with single element
$RollNo
[1] 1

student_lst[c(T,T,F,F)]            # Sublist with multiple element
$RollNo
[1] 1
$Name
[1] "ABC"
student_lst[c(T,F)]                # Recycling c(T,F) to c(T,F,T,F)
$RollNo
[1] 1
$Class
[1] "SYCOMP"

# Using [[ gives error
student_lst[[c(T,T,F,F)]]
Error in student_lst[[c(T, T, F, F)]] :
      recursive indexing failed at level 2

student_lst[[T]][[T]][[F]][[F]]
Error in student_lst[[T]][[T]][[F]] :
  attempt to select less than one element in integerOneIndex
```

Manipulating List

In R, lists can be manipulated by index or by name of element. The list elements can be added, deleted or updated. The list elements can be added only at end of list, deletion or

updating can be done on any element. For manipulation [, [[, or $ can be used with index or name.

Adding List Element

```
marks_lst <- list(Sub1 = 45, Sub2 = 54, Sub3 = 60)
student_lst <- list(RollNo = 1, Name = "ABC", Class = "SYCOMP", Marks = marks_lst)
str(student_lst)                          # Checking list structure
List of 4
 $ RollNo: num 1
 $ Name  : chr "ABC"
 $ Class : chr "SYCOMP"
 $ Marks :List of 3
  ..$ Sub1: num 45
  ..$ Sub2: num 54
  ..$ Sub3: num 60
student_lst[5] <- "Shahada"  # Adding 5th list element
student_lst[[5]] <- "Shahada"
str(student_lst)
List of 5
 $ RollNo: num 1
 $ Name  : chr "ABC"
 $ Class : chr "SYCOMP"
 $ Marks :List of 3
  ..$ Sub1: num 45
  ..$ Sub2: num 54
  ..$ Sub3: num 60
 $       : chr "Shahada"
student_lst["City"] <- "Shahada"          # Adding "City" list element
student_lst[["City"]] <- "Shahada"
student_lst$City <- "Shahada"
str(student_lst)
List of 5
 $ RollNo: num 1
 $ Name  : chr "ABC"
 $ Class : chr "SYCOMP"
 $ Marks :List of 3
  ..$ Sub1: num 45
  ..$ Sub2: num 54
  ..$ Sub3: num 60
```

```
$ City  : chr "Shahada"
```

Deleting List Element

```
marks_lst <- list(Sub1 = 45, Sub2 = 54, Sub3 = 60)
student_lst <- list(RollNo = 1, Name = "ABC", Class = "SYCOMP", Marks = marks_lst)
str(student_lst)                          # Checking list structure
List of 4
 $ RollNo: num 1
 $ Name  : chr "ABC"
 $ Class : chr "SYCOMP"
 $ Marks :List of 3
  ..$ Sub1: num 45
  ..$ Sub2: num 54
  ..$ Sub3: num 60

student_lst["Class"] <- NULL              # Deleting "Class" list element
student_lst[["Class"]] <- NULL
student_lst$Class <- NULL
str(student_lst)
List of 3
 $ RollNo: num 1
 $ Name  : chr "ABC"
 $ Marks :List of 3
  ..$ Sub1: num 45
  ..$ Sub2: num 54
  ..$ Sub3: num 60
```

Updating List Element

```
marks_lst <- list(Sub1 = 45, Sub2 = 54, Sub3 = 60)
student_lst <- list(RollNo = 1, Name = "ABC", Class = "SYCOMP",
Marks = marks_lst)
str(student_lst)                          # Checking list structure
List of 4
 $ RollNo: num 1
 $ Name  : chr "ABC"
 $ Class : chr "SYCOMP"
 $ Marks :List of 3
  ..$ Sub1: num 45
  ..$ Sub2: num 54
  ..$ Sub3: num 60
```

```
student_lst[[4]][[1]] <- 75              # Updating sublist element
student_lst[["Marks"]][["Sub1"]] <- 75
student_lst$Marks$Sub1 <- 75
str(student_lst)
List of 4
 $ RollNo: num 1
 $ Name  : chr "ABC"
 $ Class : chr "SYCOMP"
 $ Marks :List of 3
  ..$ Sub1: num 75
  ..$ Sub2: num 54
  ..$ Sub3: num 60
```

Algorithm

1. Start.
2. Create a list “student”.
3. Name every list elements using any one method.
4. Read choice for list operations from menu as
    a. Access List
    b. Create Sublist
    c. Add Element
    d. Delete Element
    e. Update Element
    f. Display
5. As per choice perform list operations as
    a. If choice is “a”, access list element using [[or $.
    b. If choice is “b”, create a sublist using [.
    c. If choice is “c”, add new element using [[or $.
    d. If choice is “d”, delete existing element using [[or $.
    e. If choice is “e”, update existing element using [[or $.
    f. If choice is “f”, display contents of list.
6. Stop.