Your answers will be checked for plagiarism and AI generated texts so be careful while answering the questions below:

- **What is “list” in R?**

  In R, a list is a data structure that stores heterogeneous types of data such as numeric, and logical values as well as other lists or objects.List is created using list() function.It takes any type of arguments separated by commas.
  For example,

  ex_list<-list(20 , “hello” , True)

  To access the data from the list, we use [[i]] where i is the index of the list.

  ie: ex_list[[1]]

- How to create a list containing strings, numbers, vectors and logical values in R?

  To create a list containing strings, numbers, vectors, and logical values in R, we can use the list() function and pass in the desired elements separated by commas.

  For example:

  ```
  ex_list <- list("hello", 123, c(1, 2, 3), TRUE)
  ```

  In this example, the ex_list object is a list containing four elements:

  - The first element is a string: "hello".
  - The second element is a number: 123.
  - The third element is a vector of three numbers: 1, 2, and 3.
  - The fourth element is a logical value: TRUE.

- How to name the list elements in R?

  We can name the elements of a list in R using the ‘ name = value ’ syntax when creating the list with the list () function.

  For_example:

  ```
  ex_list <- list (first = "hello", second = 123, third = c(1, 2, 3), fourth = TRUE)
  ```

In this example, each element of the list has been assigned a name: "first", "second", "third", and "fourth".

Also, we can assign names to list elements after the list has been created using the names() function.

Example:

ex_list <- list (first = "hello", second = 123, third = c(1, 2, 3), fourth = TRUE)

names(ex_list) <- c("first", "second", "third", "fourth")

- # How to assess list elements in R?

we can access and assess list elements in R using the **[[ ]]** or **$** operator followed by the index or name of the element.

To access an element of a list using its index, you can use double square brackets **[[ ]].** For example, to access the second element of a list called **ex_list**, we can use:

ex_list[[2]]

Alternatively , To access an element of a list using its name, we can use the $ operator.

For example, if you have a list called **ex_list** with elements named **first, second, third , and fourth**, we can access the second element using:

ex_list$second

This will return the value of the **second** element in the list.

- # How to manipulate list elements in R?

We can manipulate list elements in R using various functions and operations. We can add, delete and update list elements as shown below. We can add and delete elements only at the end of a list. But we can update any element.

1 Create a list containing a vector, a matrix and a list.

```
list_data <- list(c("Jan","Feb","Mar"), matrix(c(3,9,5,1,-2,8), nrow = 2),

 list("green",12.3))
```

2 Give names to the elements in the list.

```
names(list_data) <- c("1st Quarter", "A_Matrix", "A Inner list")
```

3 Add an element at the end of the list.

```
list_data[4] <- "New element"

print(list_data[4])
```

4 Remove the last element.

```
list_data[4] <- NULL
```

5 Print the 4th Element.

```
print(list_data[4])
```

6 Update the 3rd Element.

```
list_data[3] <- "updated element"

print(list_data[3])
```

- # How to convert lists to vectors in R?

We can convert a list to a vector in R using the unlist() function. The unlist() function takes a list as input and returns a vector by concatenating all the elements in the list.

```
# Create a list with different types of elements
```

```
ex_list <- list("a", 1, TRUE, c(2, 4, 6))
# Convert the list to a vector
ex_vector <- unlist(ex_list)
# Print the type to check whether it is vector or not
print(typeof(ex_vector))
```