The built-in vector LETTERS contains the uppercase letters of the alphabet. Produce a vector of

(i) the first 12 letters;

(ii) the odd ‘numbered’ letters;

(iii) the (English) consonants.

```
first12 <- LETTERS[1:12]
> first12
 [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L"
```

```
odd_letters <- LETTERS[1:length(LETTERS) %% 2 == 1]
> odd_letters
 [1] "A" "C" "E" "G" "I" "K" "M" "O" "Q" "S" "U" "W" "Y"
```

```
consonants <- LETTERS[!LETTERS %in% c("A", "E", "I", "O", "U")]
> consonants
 [1] "B" "C" "D" "F" "G" "H" "J" "K" "L" "M" "N" "P" "Q" "R" "S" "T" "V" "W" "X" "Y" "Z"
```

2. The function rnorm() generates normal random variables. For instance, rnorm(10) gives a vector of 10 i.i.d. standard normals. Generate 20 standard normals, and store them as x. Then obtain subvectors of

(i) the entries in x which are less than 1;

(ii) the entries between – 0.5 and 1;

(iii) the entries whose absolute value is larger than 1.5.

```
x <- rnorm(20)
 [1] -1.28619639 -0.12752418 -0.76420717  1.52002349  0.03484504  1.06510315  0.40637424
 [8] -0.22479954 -1.07620521 -0.77248134 -1.00649813 -0.50197229  0.32312136 -0.64784885
[15]  1.47180071  0.85880009 -0.06466283 -0.16360194  0.29830258 -1.29678467
less_than_1 <- x[x < 1]
 [1] -1.2861964 -0.1275242 -0.7642072  0.0348450  0.4063742 -0.2247995 -0.7724813
```

```
 [8] -1.0064981 -0.5019723 -0.6478488 -0.0646628 -0.1636019  0.2983026
between_neg05_and_1 <- x[x > -0.5 & x < 1]
[1] -0.1275242  0.0348450  0.4063742 -0.2247995 -0.5019723 -0.6478488 -0.0646628 -0.1636019
[9]  0.2983026
abs_larger_than_15 <- x[abs(x) > 1.5]
[1] 1.5200235 1.0651031 1.4718007
```

3. Solve the following system of simultaneous equations using matrix methods.

a + 2b + 3c + 4d + 5e = −5

2a + 3b + 4c + 5d + e = 2

3a + 4b + 5c + d + 2e = 5

4a + 5b + c + 2d + 3e = 10

5a + b + 2c + 3d + 4e = 11

```
A <- matrix(c(1, 2, 3, 4, 5,
         2, 3, 4, 5, 1,
         3, 4, 5, 1, 2,
         4, 5, 1, 2, 3,
         5, 1, 2, 3, 4), nrow = 5, byrow = TRUE)
b <- c(-5, 2, 5, 10, 11)
x <- solve(A, b)
print(x)
[1]  1 -1  0  1 -1
```

4. Create a factor object for an apple color such as 'green', 'green', 'yellow', 'red', 'red', 'red','

green'. Print the factor and applying the nlevels function to know the number of distinct

values

```
colors <- c('green', 'green', 'yellow', 'red', 'red', 'red', 'green')
color_factor <- factor(colors)
```

```
print(color_factor)
[1] green  green  yellow red    red    red    green
Levels: green red yellow
print(nlevels(color_factor))
[1] 3
```

5. Create an S3 object of class fruit contains a list with following required components such as name, quantity, cost and also Define and create s4 objects.Define a reference class of fruit

```
fruit <- function(name, quantity, cost) {
  list(name = name,
       quantity = quantity,
       cost = cost,
       class = "fruit")
}
apple <- fruit(name = "Apple", quantity = 10, cost = 1.5)
print(apple)
library(R6)

fruit_s4 <- R6Class("fruit",
  public = list(
    name = NA,
    quantity = NA,
    cost = NA,
    initialize = function(name, quantity, cost) {
      self$name <- name
      self$quantity <- quantity
      self$cost <- cost
    }
  )
```

```
)

banana <- fruit_s4$new(name = "Banana", quantity = 5, cost = 0.5)
print(banana)
$name
[1] "Apple"

$quantity
[1] 10

$cost
[1] 1.5

$class
[1] "fruit"


<fruit>
Public:
  clone: function (deep = FALSE)
  finalize: function ()
  initialize: function(name, quantity, cost)
  name: <NA_character_>
  quantity: <NA_real_>
  cost: <NA_real_>
  private: <environment>


Reference class object of class "fruit"
Field "name":
[1] "Orange"
```

Field "quantity":

[1] 8

Field "cost":

[1] 2