UNIVERSITY OF MALAYA

WIX1002

FUNDAMENTALS OF PROGRAMMING

LAB REPORT 1

OCC 12

PREPARED BY: GROUP 1

NAME	MATRIC NUMBER
NICOL HENG SI YI	24004564
TAN YI JING	24004568
TAN SYN YEE	24004607
VALENCIEN SEOW YUN SUN	24004618
KAN PENNY	24004556

LECTURER: DR. ADELEH ASEMI ZAVAREH

Question 1

1. Problem:

   The requirement is to process a number and reduce it to its digital root, which means summing the digits of a number repeatedly until a single-digit number is obtained. This concept is helpful in error detection algorithms, where reducing a number to its single-digit form helps in quick verification.

   The program needs to prompt input from users and handle such calculations efficiently, using different loop control structures.

2. Solution:

   Use a do-while loop to prompt user until user inputs a valid positive integer.

   Use a while loop to check whether the number is more than one digit. The sum variable is initialized to zero every time while the loop is looping.

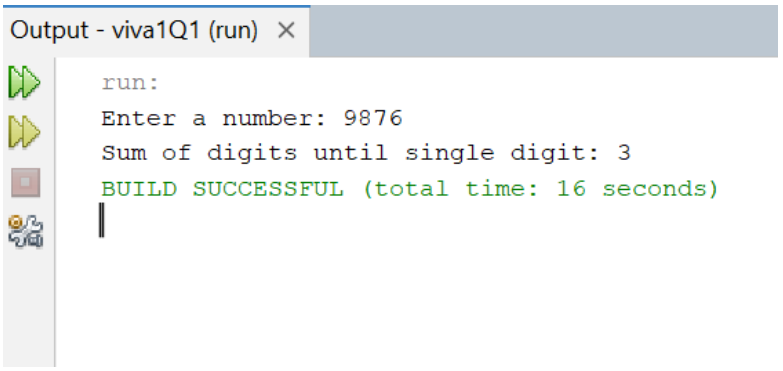
   Inside the while loop, there is a for loop that sum the digits of the number from right to left. After the for loop, the value of sum is assigned to the variable num. Sum the digits of the number repeatedly until a single-digit number is obtained. Once the number is reduced to a single digit, the loop exits, and the result is printed.

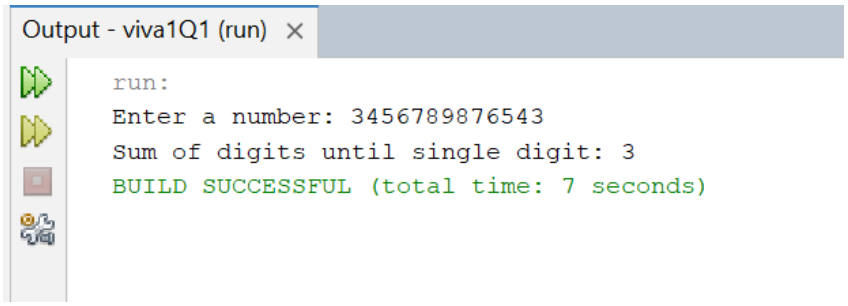
   Pseudocode

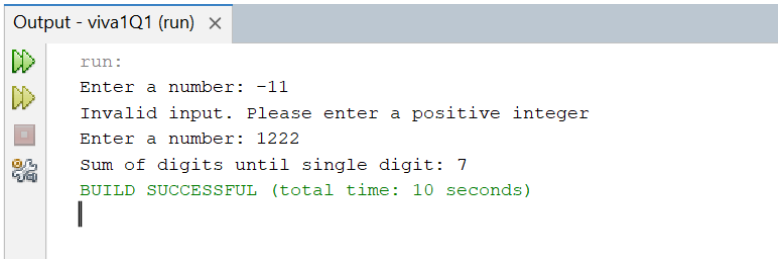
```
Start
        do
        Input num
        if(num<=0)
                display "Invalid input. Please enter a positive integer"
        end if
        while(num<=0)
        while(num>=10)
                Initialize sum to 0
                while(num>0)
                        sum+=num%10
                        num/=10
                end while
                num=sum
        end while
display sum
```

Stop

3. Sample input and output







4. Source code

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package vivalql;


/**
 *
 * @author kanpenny
 */
import java.util.Scanner;
public class VivalQql {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        long num;
        //prompt user until a positive integer is entered
        do{
            System.out.print("Enter a number: ");
            num = sc.nextLong();
            if(num<=0){
                System.out.println("Invalid input. Please enter a positive integer");
            }
        }while(num<=0);

        //repeat the process as long as num has more than one digit
        while(num>=10){
            long sum= 0;//initialize sum
            //loop to calculate the sum of all digits in num
            for(;num>0;num/=10){//eliminate the last digit(has been added to sum) from num
                sum+=num%10;}//extract the last digit(remainder after modulus 10) and add it to sum
            num=sum;//update the num to the sum for the next iteration
            }
        System.out.println("Sum of digits until single digit: " +num);//print final single digit sum
        sc.close();
        }
    }
```

Question 2

1. Problem:

   Accept three inputs: an integer n, and two integers a and b from the user. The task is to determine the fewest steps required to reduce n to 1, using either subtraction of n by a or division of n by b. If n is possible to reach 1, return the fewest step used to reach 1, else return -1.

2. Solution:

   Prompt the user to enter values for n, a, and b. There are two possible strategies to approach the solution: one where division is attempted first, and one where subtraction is attempted first. Both strategies should be implemented using a while loop, which will continue until n is less than or equal to 1, or until no further operations can be applied. The number of steps taken for each strategy will be compared, and the fewest number of steps will be printed, provided that n can be reduced to 1. If it's not possible, return -1.

   Pseudocode

```
initialise step1 to 0 and step2 to 0
get 3 number, n1, a and b from the user
WHILE the number is out of the constraints
    get 3 number, n1, a and b from the user
assign value of n1 to n2
WHILE n1 is greater than 1
     WHILE n1 is divisible by b
         n1 divide by b
         increase step1 by 1
      WHILE n1 is bigger than a
         n1 minus by a
         increase step1 by 1
      IF n1 is not divisible by b and n1 is less than or equal to a
         break the loop
      END IF
END WHILE
WHILE n2 is greater than 1
     WHILE n2 is bigger than a
```

n2 minus by a

increase step2 by 1

WHILE n2 is divisible by b

n2 divide by b

increase step2 by 1

IF n2 is not divisible by b and n2 is less than or equal to a

break the loop

END IF

END WHILE

END WHILE

IF n1 is equal to 1 and step1 is less than or equal to step2

display step1

ELSE IF n2 is equal to 1 and step2 is less than or equal to step1

display step2

ELSE

display -1

END IF

close scanner

3. Sample Input and Output:

```
Constraints:
             1 <= n <= 10^9
             1 <= a <= n
             2 <= b <= 10^5

Enter the number for n, a and b : 10 2 2
3
BUILD SUCCESSFUL (total time: 3 seconds)
```

```
Constraints:
             1 <= n <= 10^9
             1 <= a <= n
             2 <= b <= 10^5

Enter the number for n, a and b : 15 5 3
-1
BUILD SUCCESSFUL (total time: 2 seconds)
```

4. Source code:

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package q2;

import java.util.Scanner;

/**
 *
 * @author user
 */
public class Q2 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        Scanner sc = new Scanner(System.in);
        int n1, n2, a, b, step1 = 0, step2 = 0;
        System.out.println("""
                           Constraints:
                                   1 <= n <= 10^9
                                   1 <= a <= n
                                   2 <= b <= 10^5
                           """);
        System.out.print("Enter the number for n, a and b : ");
        n1 = sc.nextInt();
        a = sc.nextInt();
        b = sc.nextInt();
        while (n1 < 1 || n1 > 1000000000 || a < 1 || a > n1 || b < 2 || b > 100000) {
            System.out.println("Input is invalid. Try again.");
            System.out.print("Enter the number for n, a and b : ");
            n1 = sc.nextInt();
            a = sc.nextInt();
            b = sc.nextInt();
        }
        n2 = n1;        //divide then minus
        while (n1 > 1) {
            while (n1 % b == 0) {
                n1 /= b;

                step1++;
            }
            while (n1 > a) {
                n1 -= a;
                step1++;
            }
            if (n1 % b != 0 && n1 <= a)
                break;
        }
        //minus then divide
        while (n2 > 1) {
            while (n2 > a) {
                n2 -= a;
                step2++;
            }
            while (n2 % b == 0) {
                n2 /= b;
                step2++;
            }
            if (n2 % b != 0 && n2 <= a)
                break;
        }
        //compare which step is fewer
        if (n1 == 1 && step1 <= step2)
            System.out.println(step1);
        else if (n2 == 1 && step2 <= step1)
            System.out.println(step2);
        else
            System.out.println("-1");

        sc.close();
    }

}
```

Question 3

1. Problem:

   the program has to find the value listed below based on a given number.

   a. Whether it is prime.
   b. Its factors and how many there are.
   c. The sum and product of its factors.
   d. If it is a perfect number.
   e. All prime numbers less than 96 for comparison in research.

2. Solution:

   To check whether a number is prime, use a for loop with an if statement to determine if the number is divisible by any integer i. If the number is divisible by i, then it is not prime. Since i is a factor, increase the factors counter by 1. Display a message indicating that the number is not prime.

   If the number is not prime, the program should also find all of its factors and calculate the product and sum of these factors. To find the factors, use a for loop with an if statement to check if the number is divisible by each integer i. If it is, i is a factor, so display it. Add each factor to the factor sum. To calculate the product of factors without risking overflow, use an if statement to check if the current product multiplied by i would exceed the maximum value for a long integer (by comparing it to max value of long / i).

   Next, to determine if the number is a perfect number, use an if statement to check if the number is equal to the factor sum minus the number itself. If it is, print that the number is a perfect number; otherwise, print that it is not.

   Lastly, to display all prime numbers smaller than the given number, use a for loop to iterate through all numbers (i.e., integers i) from 2 up to the given number. For each i, use a while loop to check if i is divisible by any integer less than i. If i is not divisible by any of these numbers, print i as a prime number.

   Pseudocode

   INPUT n

   isPrime = true, factorCtr = 2, factorSum = 0, factorPdt =1

```
FOR i from 0 to (n-1)
        IF (n%i == 0)
                isPrime = false
                factorCtr++
        END IF
END FOR
isOverflow = false
IF !(isPrime)
        DISPLAY “not prime number”, factorCtr
        FOR i from 1 to n
                IF (n % i == 0)
                        DISPLAY i
                        factorSum += i
                        IF !(isOverflow)
                                IF (factorPdt > Long.MAX_VALUE / i)
                                        isOverflow = true
                                ELSE
                                        factorPdt *= i
                                END IF
                        END IF
                END IF
        END FOR
        DISPLAY factorSum
        If (isOverFlow)
                DISPLAY “product too large”
        ELSE
                DISPLAY factorPdt
        END IF
        IF (factorSum-n == n)
                DISPLAY “is perfect number”
        ELSE
                DISPLAY “is not perfect number”
        END IF
ELSE
```

```
        DISPLAY “is prime number”
    END IF
    int count = 0
    FOR i from 2 to (n-1)
        J = 2
        isPrime = true
        WHILE (j < i)
            If (i % j == 0)
                isPrime = false
            END IF
            J++
        END WHILE
        IF (isPrime)
            DISPLAY i
        END IF
    END FOR
```

3. Sample input and output

```
run:
Enter a positive integer larger than 1 : 28
Integer is not a prime number, it has 6 factors
The factors of this integer are:
1, 2, 4, 7, 14, 28
The sum of the factors is 56
The product of the factors is 21952
28 is a perfect number.
Prime numbers between 2 and 28: 2, 3, 5, 7, 11, 13, 17, 19, 23
BUILD SUCCESSFUL (total time: 11 seconds)
```

```
run:
Enter a positive integer larger than 1 : 96
Integer is not a prime number, it has 12 factors
The factors of this integer are:
1, 2, 3, 4, 6, 8, 12, 16, 24, 32, 48, 96
The sum of the factors is 252
The product of the factors is 782757789696
96 is not a perfect number.
Prime numbers between 2 and 96: 2, 3, 5, 7, 11, 13, 17, 19, 23,
29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89
BUILD SUCCESSFUL (total time: 2 seconds)
```

4. Source code

```java
import java.util.Scanner;

/**
 *
 * @author user
 */
public class Q3 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        Scanner sc = new Scanner(System.in);
        int n, factorCtr = 2, factorSum = 0, j;
        long factorPdt = 1;
        boolean isPrime = true;

        System.out.print("Enter a positive integer larger than 1 : ");
        n = sc.nextInt();
        //Ensure input is larger than 1
        while (n <= 1){
            System.out.print("Input integer is less than or equal to 1. Enter a positive integer that is larger than 1: ");
            n = sc.nextInt();
        }
        //determine whether ineteger is prime, if not count the number of factor
        for (int i = 2; i < n; i++) {
            if (n % i == 0) {
                isPrime = false;
                factorCtr++;
            }
        }
        //handle overflow case
        boolean isOverflow = false;
        //if integer is not prime, list out factor and display sum and product of factor(if not overflowed)
        if (!isPrime) {
            System.out.println("Integer is not a prime number, it has " + factorCtr + " factors");
            System.out.println("The factors of this integer are: ");
            for (int i = 1; i <= n; i++) {
                if (n % i == 0) {
                    System.out.print(i);
```

```java
                    factorSum += i;
                    if (!isOverflow) {
                        if (factorPdt > Long.MAX_VALUE / i)
                            isOverflow = true;
                        else {
                            factorPdt *= i;
                        }
                    }
                    if (i != n)
                        System.out.print(", ");
                    else
                        System.out.println();
                }
            }
            System.out.println("The sum of the factors is " + factorSum);
            if (isOverflow)
                System.out.println("The product of the factors is too large to display");
            else
                System.out.println("The product of the factors is " + factorPdt);

            //check perfect number
            if (factorSum-n == n)
                System.out.println(n + " is a perfect number.");
            else
                System.out.println(n + " is not a perfect number.");
        }
        else
            System.out.println("Integer is a prime number.");

        //print prime number
        int count = 0;
        int y = -1;
        System.out.printf("Prime numbers between 2 and %d: ", n);
        for (int i = 2; i < n; i++) {
            j = 2;
            isPrime = true;
            while (j < i) {
                if (i % j == 0)
                    isPrime = false;
                j++;
            }
```

```
            if (isPrime) {
                if (i != 2) {
                    System.out.print(", ");
                    if (count == 25 + (y * 16)) {
                        System.out.println();
                        y++;
                    }
                }
                System.out.print(i);
                count++;
            }
        }
        System.out.println();

        sc.close();
    }

}
```

Question 4

1. Problem:

   Prompt the customer to enter their options repeatedly. Once customers pick an item, they can navigate back to the same sub-menu to pick more items. Customers also can navigate back to the main menu when choosing the 6th option in any sub-menu. Thus, the main menu is printed out repeatedly. Calculate the bills of customers based on their order in Maroni's Pizza. Determine whether the customer can get a 20% discount.

2. Solution:

   Use the while(true) loop to keep the program running infinitely until a break statement ends it. Use break and continue statements to stop or continue the particular labeled loop. Prompt the customer to enter options. Switch statement is used to represent multiway if-else statements for options input by customer in 4 menus. Create several methods and call the methods, for instance, PizzaMenu, DrinksMenu, DessertMenu, addTocart and Checkout to help navigate menus and calculate total bills effectively. Lastly, use the if statement to determine whether the customer orders at least 1 item from each sub-menu. The customer will be given a 20% discount offer if he or she orders at least 1 item from each sub-menu.

Pseudocode

```
set total to 0
set orderedPizza to false, orderedDrinks to false, orderedDessert to false
while (true)
Print message "Welcome to Maroni's Pizza!
                        1. Pizza
                        2. Drinks
                        3. Dessert
                        4. CHECKOUT"
read option
if option ==1
        while (true)
                print message
                "PIZZA
                        1 Chicken Peperoni - RM15
```

```
        2 Chicken Supreme  - RM18
        3 Vegan Indulgence - RM12
        4 Beef Delight           - RM22
        5 Margherita   - RM9
        6 BACK TO MAIN MENU"
        read optionPizza
        if optionPizza==1
          total=total+15
        print "Added Chicken Peperoni"
        print total
          orderedPizza=true
  else if optionPizza==2
        total=total+18
        print "Added Chicken Supreme"
        print total
          orderedPizza=true
 else if optionPizza==3
          total=total+12
        print "Added Vegan Indulgence"
        print total
          orderedPizza=true
else if optionPizza==4
          total=total+22
        print "Added Beef Delight"
        print total
          orderedPizza=true
else if optionPizza==5
          total=total+9
        print "Added Margherita"
        print total
          orderedPizza=true
else if optionPizza==6
        break loop
else
```

```
                print “Invalid option. Please try again”
            end if
        end while


else if option==2
while (true)
            print message
            “DRINKS
                1 Strawberry Smoothie - RM8
                2 Banana Smoothie - RM8
                3 Mocktail - RM12
                4 Soft Drink - RM5
                5 Mineral Water - RM3
                6 BACK TO MAIN MENU”
                read optionDrinks
                if optionDrinks==1
                  total=total+8
                print “Added Strawberry Smoothie”
                print total
                 orderedDrinks=true
                else if optionDrinks==2
                  total=total+8
                print “Added Banana Smoothie”
                print total
                 orderedDrinks=true
            else if optionDrinks==3
                  total=total+12
                print “Added Mocktail”
                print total
                 orderedDrinks=true
            else if optionDrinks==4
                  total=total+5
                print “Added Soft Drink”
```

```
                print total
                  orderedDrinks=true
            else if optionDrinks==5
                   total=total+3
                print “Added Mineral Water”
                print total
                  orderedDrinks=true
            else if optionDrinks==6
                break loop
            else
                print “Invalid option. Please try again”
            end if
        end while
else if option==3
while (true)
            print message
            “DESSERT
                1 Tiramisu - RM7
                2 Strawberry Shortcake - RM10
                3 Green Jello - RM4
                4 Creme Brulee - RM15
                5 Raspberry Pie - RM20
                6 BACK TO MAIN MENU”
                read optionDessert
                if optionDessert==1
                  total=total+7
                print “Added Tiramisu”
                print total
                  orderedDessert=true
                else if optionDessert==2
                  total=total+10
                print “Added Strawberry Shortcake”
                print total
                  orderedDessert=true
```

```
                else if optionDessert==3
                        total=total+4
                    print “Added Green Jello”
                    print total
                     orderedDessert=true
                else if optionDessert==4
                        total=total+15
                    print “Added Creme Brulee”
                    print total
                     orderedDessert=true
                else if optionDessert==5
                        total=total+20
                    print “Added Raspberry Pie”
                    print total
                     orderedDessert=true
                else if optionDessert==6
                    break loop
                else
                    print “Invalid option. Please try again”
                end if
        end while
else if option==4
print total
if (orderedPizza==true && orderedDrinks==true && orderedDessert==true)
    newtotal=total*0.8
        print “You've availed the One-of-each offer. You get a 20% discount!”
        print newtotal
else
        print “Have a nice day!”
```

3. Sample Input and Output

```
run:

Welcome to Maroni's Pizza!
1. Pizza
2. Drinks
3. Dessert
4. CHECKOUT

Pick an option: 1

PIZZA
1 Chicken Peperoni - RM15
2 Chicken Supreme  - RM18
3 Vegan Indulgence - RM12
4 Beef Delight     - RM22
5 Margherita       - RM9
6 BACK TO MAIN MENU

Pick an option:2
Added Chicken Supreme
Current total: RM18.0

Pick an option:6

Welcome to Maroni's Pizza!
1. Pizza
2. Drinks
3. Dessert
4. CHECKOUT

Pick an option: 4

Your total is RM18.0!
Have a nice day!
BUILD SUCCESSFUL (total time: 9 seconds)
```

```
run:

Welcome to Maroni's Pizza!
1. Pizza
2. Drinks
3. Dessert
4. CHECKOUT

Pick an option: 1

PIZZA
1 Chicken Peperoni - RM15
2 Chicken Supreme  - RM18
3 Vegan Indulgence - RM12
4 Beef Delight     - RM22
5 Margherita       - RM9
6 BACK TO MAIN MENU

Pick an option:3
Added Vegan Indulgence
Current total: RM12.0

Pick an option:6

Welcome to Maroni's Pizza!
1. Pizza
2. Drinks
3. Dessert
4. CHECKOUT

Pick an option: 2
```

```
DRINKS
1 Strawberry Smoothie - RM8
2 Banana Smoothie - RM8
3 Mocktail - RM12
4 Soft Drink - RM5
5 Mineral Water - RM3
6 BACK TO MAIN MENU

Pick an option:2
Added Banana Smoothie
Current total: RM20.0

Pick an option:6

Welcome to Maroni's Pizza!
1. Pizza
2. Drinks
3. Dessert
4. CHECKOUT

Pick an option: 3

DESSERT
 1 Tiramisu - RM7
 2 Strawberry Shortcake - RM10
 3 Green Jello - RM4
 4 Creme Brulee - RM15
 5 Raspberry Pie - RM20
 6 BACK TO MAIN MENU

Pick an option: 1
Added Tiramisu
Current total: RM27.0


 Pick an option: 6

 Welcome to Maroni's Pizza!
 1. Pizza
 2. Drinks
 3. Dessert
 4. CHECKOUT

 Pick an option: 4

 Your total is RM27.0!
 You've availed the One-of-each offer. You get a 20% discount!
 Your new total is RM21.6!
 Have a nice day!
 BUILD SUCCESSFUL (total time: 25 seconds)
```

4. Source code

```
11  import java.util.Scanner;
12
13  public class viva5q4 {
14      static Scanner input=new Scanner(System.in);
15      static boolean orderedPizza=false;
16      static boolean orderedDrinks=false;
17      static boolean orderedDessert=false;
18      static double total=0;
19
20
21      public static void main(String[] args) {
22      mainMenu:
23      while(true) {
24           System.out.print("""
25                            \nWelcome to Maroni's Pizza!
26                            1. Pizza
27                            2. Drinks
28                            3. Dessert
29                            4. CHECKOUT
30
31                              """);
32          System.out.print("Pick an option: ");
33          int option=input.nextInt();
34
            switch (option) {
36              case 1 :
37                  PizzaMenu();
```

```
                continue mainMenu;
            case 2:
                DrinksMenu();
                continue mainMenu;
            case 3 :
                DessertMenu();
                continue mainMenu;
            case 4:
                Checkout(orderedPizza,orderedDrinks,orderedDessert);
                break mainMenu;
            default:
                System.out.println("Invalid option. Please try again.");
                continue mainMenu;
                }
            }
    }
    public static void PizzaMenu() {
        System.out.print("""
                         \nPIZZA
                         1 Chicken Peperoni - RM15
                         2 Chicken Supreme  - RM18
                         3 Vegan Indulgence - RM12
                         4 Beef Delight     - RM22
                         5 Margherita       - RM9
                         6 BACK TO MAIN MENU

                           """);
```

```
65              Pizzaloop:
66              while(true) {
67                  System.out.print("Pick an option:");
68                  int optionPizza=input.nextInt();
69
                    switch (optionPizza){
71                      case 1:
72                          addTocart("Chicken Peperoni",15);
73                          orderedPizza=true;
                            continue Pizzaloop;
75                      case 2:
76                          addTocart("Chicken Supreme",18);
77                          orderedPizza=true;
                            continue Pizzaloop;
79                      case 3:
80                          addTocart("Vegan Indulgence",12);
81                          orderedPizza=true;
                            continue Pizzaloop;
83                      case 4:
84                          addTocart("Beef Delight",22);
85                          orderedPizza=true;
                            continue Pizzaloop;
87                      case 5:
88                          addTocart("Margherita",9);
89                          orderedPizza=true;
                            continue Pizzaloop;
91                      case 6:
92                          break Pizzaloop;
93                      default:
94                          System.out.println("Invalid option. Please try again.");
95                  }
96              }
97          }
```

```
99              public static void DrinksMenu() {
100                         System.out.print("""
101                                 \nDRINKS
102                                 1 Strawberry Smoothie - RM8
103                                 2 Banana Smoothie - RM8
104                                 3 Mocktail - RM12
105                                 4 Soft Drink - RM5
106                                 5 Mineral Water - RM3
107                                 6 BACK TO MAIN MENU
108
109                                 """);
```

```
110                 Drinksloop:
111                 while(true) {
112                 System.out.print("Pick an option:");
113                 int optionDrinks=input.nextInt();
114
115                 switch (optionDrinks) {
116                     case 1:
117                         addTocart("Strawberry Smoothie",8);
118                         orderedDrinks=true;
119                         continue Drinksloop;
120                     case 2:
121                         addTocart("Banana Smoothie",8);
122                         orderedDrinks=true;
123                         continue Drinksloop;
124                     case 3:
125                         addTocart("Mocktail",12);
126                         orderedDrinks=true;
127                         continue Drinksloop;
128                     case 4:
129                         addTocart("Soft Drink",5);
130                         orderedDrinks=true;
131                         continue Drinksloop;
132                     case 5:
133                        addTocart("Mineral Water ",3);
134                        orderedDrinks=true;
135                        continue Drinksloop;
136                     case 6:
137                         break Drinksloop;
138                     default:
139                         System.out.println("Invalid option. Please try again.");
140                 }
141                     }
142                 }
143
144                 public static void DessertMenu() {
145                         System.out.print("""
146                                         \nDESSERT
147                                         1 Tiramisu - RM7
148                                         2 Strawberry Shortcake - RM10
149                                         3 Green Jello - RM4
150                                         4 Creme Brulee - RM15
151                                         5 Raspberry Pie - RM20
152                                         6 BACK TO MAIN MENU
153
154                                        """);
```

```
155                 Dessertloop:
156                 while(true) {
157                     System.out.print("Pick an option: ");
158                     int optionDessert=input.nextInt();
159 
160                     switch(optionDessert) {
161                         case 1 :
162                             addTocart("Tiramisu",7);
163                             orderedDessert=true;
164                             continue Dessertloop;
165                         case 2:
166                             addTocart("Strawberry Shortcake",10);
167                             orderedDessert=true;
168                             continue Dessertloop;
169                         case 3 :
170                             addTocart("Green Jello",4);
171                             orderedDessert=true;
172                             continue Dessertloop;
173                         case 4:
174                             addTocart("Creme Brulee",15);
175                             orderedDessert=true;
176                             continue Dessertloop;
177                         case 5:
178                             addTocart("Raspberry Pie",20);
179                             orderedDessert=true;
180                             continue Dessertloop;
181                         case 6:
182                             break Dessertloop;
183                         default:
184                             System.out.println("Invalid option. Please try again.");
185                     }
186                 }
187             }
```

```
188             public static void addTocart(String item, double price) {
189                 total+=price;
190                 System.out.println("Added " + item);
191                 System.out.printf("Current total: RM%.1f\n\n",total);
192             }
193             public static void Checkout(boolean orderedPizza ,boolean orderedDrinks, boolean orderedDessert) {
194                 System.out.printf("\nYour total is RM%.1f!\n",total);
195                 if (orderedPizza && orderedDrinks && orderedDessert) {
196                     double newtotal=total*0.8;
197                     System.out.println("You've availed the One-of-each offer. You get a 20% discount!");
198                     System.out.printf("Your new total is RM%.1f!\n",newtotal);
199                 }
200 
201                 System.out.println("Have a nice day!");
202             }
203         }
```

Question 5

1. Problem:

   Remove all “REMIX” from string input by user, and display the original text.

2. Solution:

   Firstly, use the method replaceAll to replace all “REMIX” with “ “. Afterwards, use replaceAll again, but to remove extra space between words by using a regex to tell the program to look for extra space and replace them with a single space. Then, use trim to remove extra space in the front and at the back.

   Pseudocode:

   1. Input text
   2. Initialise original
   3. original = Replace all “REMIX” in text with “ “
   4. original = trim space in the front and at the back of of words in original
   5. original = replace all extra space with single space
   6. Display original

3. Sample Input and Output:

```
run:
Enter text: REMIXREMIXIREMIXLOVEREMIXMUSICREMIX
original text: I LOVE MUSIC
BUILD SUCCESSFUL (total time: 11 seconds)
```

```
run:
Enter text: REMIXREMIXREMIXABCREMIX
original text: ABC
BUILD SUCCESSFUL (total time: 16 seconds)
```

4. Source Code

```java
import java.util.Scanner;
public class Q5 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner input = new Scanner (System.in);
        System.out.print("Enter text: ");
        String text = input.nextLine();

        //initialize var
        String original;

        //removing process
        //remove remix
        original = text.replaceAll("REMIX", " ");
        //remove extra space, trim remove the one in the front and at the end, replace all replace the one extra in the middle
        original = original.trim().replaceAll(" +"," ");
        System.out.println("original text: "+ original);

        input.close();
    }

}
```

Question 6

1. Problem:

   Enter scores of players into the system one by one. The data entry ends when the score 0 is entered, indicating end of the round. Identify the highest score among all the scores entered and count how many times it appears. Then, determine the second-highest score and its frequency if it exists and calculate the total sum of scores recorded in the round. Lastly, check and indicate any negative scores.

2. Solution:

   Prompt the user to enter numbers using while loop and break the loop if a number 0 is entered. Subsequently, use if and else-if condition to find highest score, frequency of highest score, second-highest score, frequency of second-highest score (if exists), check whether there are negative scores in the data, and sum of all scores entered. The frequency of highest score and second-highest score will reset to 1 if a new highest score and second-highest score are found. A message "Negative numbers were entered" will be printed if there exists a negative score.

Pseudocode:

```
countSecondLarge=0
countMax=0
sum=0
max=Integer.MIN_VALUE
secondLarge=Integer.MIN_VALUE
Print "Enter numbers: "
Repeat while the condition is true
    Input numbers
    if (number=0)
        Break the loop
    end if
    if (number>max)
        secondLarge=max
        max=number
        Update countSecondLarge to countMax if secondLarge not equal to
    Integer.MIN_VALUE
        countMax=1
```

```
        else if (number=max)
                countMax=countMax+1
        else if (number<max & number>secondLarge)
                secondLarge=number
                countSecondLarge=countSecondLarge+1
        end if

        if (number<0)
                negative=true
        end if
sum=sum+number
end while
Print “The largest number is: ”, max
Print “The occurrence count of the largest number is ”, countMax
if (countSecondLarge > 0)
        Print “The second-largest number is ” , secondLarge
        Print “The occurrence count of the second-largest number is ”, countSecondLarge
else
        Print “There was no valid second-largest number.”
end if
Print “The total sum of all numbers is ” , sum
if (negative=true)
        Print “Negative numbers were entered”
end if
```

3. Sample Input and Output:

```
run:
Enter numbers: 3 5 2 5 -3 5 5 0
The largest number is 5
The occurrence count of the largest number is 4
The second-largest number is 3
The occurence count of the second-largest number is 1
The total sum of all numbers is 22
Negative numbers were entered
BUILD SUCCESSFUL (total time: 7 seconds)
```

```
run:
Enter numbers: 4 4 4 0
The largest number is 4
The occurrence count of the largest number is 3
There was no valid second-largest number.
The total sum of all numbers is 12
BUILD SUCCESSFUL (total time: 3 seconds)
```

```
run:
Enter numbers: 5 0
The largest number is 5
The occurrence count of the largest number is 1
There was no valid second-largest number.
The total sum of all numbers is 5
BUILD SUCCESSFUL (total time: 2 seconds)
```

4. Source code:

```
7  import java.util.Scanner;
8
9  public class VivaQ6 {
10
11
12     public static void main(String[] args)
13     {
14         Scanner sc=new Scanner(System.in);
15         int number,countSecondLarge=0,countMax=0,sum=0;
16         int max=Integer.MIN_VALUE,secondLarge=Integer.MIN_VALUE;
17         boolean negative=false;
18
19         System.out.print("Enter numbers: ");
20
21         while(true)
22         {number=sc.nextInt();
23
24         if(number==0)
25             break;
```

```
31             if(number>max)
32                 {secondLarge=max;
33                 max=number;
34                 countSecondLarge=(secondLarge!=Integer.MIN_VALUE)?countMax:0 ;
35                 countMax=1;}
36
37             else if(number==max)
38                     countMax++;
39
40             else if(number<max && number>secondLarge)
41                 {secondLarge=number;
42                 countSecondLarge=1;}
43
44             else if(number==secondLarge)
45                 countSecondLarge++;
46
47             if(number<0)
48                 negative=true;
```

```
45
46          sum+=number;
47          }
48
49
50          System.out.println("The largest number is "+max);
51          System.out.println("The occurrence count of the largest number is "+countMax);
52
53          if(countSecondLarge>0)
54              {System.out.println("The second-largest number is "+secondLarge);
55              System.out.println("The occurrence count of the second-largest number is "+countSecondLarge);}
56
57          else
58              System.out.println("There was no valid second-largest number. ");
59
60          System.out.println("The total sum of all numbers is "+sum);
61
62          if(negative==true)
63              System.out.println("Negative numbers were entered");
64
65      }
66
67  }
68
```