INSTITUT TEKNOLOGI SEPULUH NOPEMBER
Department of Information Systems
Undergraduate Program

SEMESTER 2 MIDTERM EXAMINATION 2022/23

IS184203 – Algorithm and Programming (3 credits)

Course Convenors:
Ahmad Muklason
Renny Pradina Kusumawardani
Rizal Risnanda Hutama

Answer and Marking Scheme

27 March 2023 TIME ALLOWED: 2 HOURS

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains **SEVEN (7)** questions
2. Answer strictly **FIVE (5)** questions only, i.e. questions numbers 1, either 2 or 3, 4, 5, and either 6 or 7. The marks for each question are indicated at the beginning of each question.
3. Answer the questions in any order.
4. This is **CLOSED BOOK** exam.
5. When justification is required, you **MUST** write down systematically the steps in the workings.
6. Please work independently, and **Good Luck!**

Question 1. (15 marks)

The code below is created to determine whether a toddler is stunting or not. A toddler is said stunting if his/her z-score is less than the standard, i.e. **0.45**. The z-score is counted using the following formula: $z-score = 2-\frac{w-h}{y}$, whereas w, h, y are weight (in 100 gram), height (in cm), and age (in months) respectively. However the code contains **five** lines causing errors. Fix, the code to know whether a toddler with 100 cm height, 15 kg weight, and 36 month old is stunting or not!

```
public class Stunting{
    public static void main(String[] args) {
        int todAge= 36;
        ___ todHeight=___
        int todWeight=15;
        ___ zScore =  ___
        boolen isStunting= ___
        if (___) ____
        if (___) System.out.println("stunting");
        else System.out.println("normal");
    }
}
```

Answer

```
public class Stunting{
    public static void main(String[] args) {
        int todAge= 36;
        int todHeight=100;
        int todWeight=15;
        double zScore =  2.0-(todWeight*10.0-todHeight)/todAge;
        boolean isStunting=false;
        if (zScore<0.45) isStunting=true;
        if (isStunting) System.out.println("stunting");
        else System.out.println("normal");
    }
}
```

Marking Scheme
There are **five lines** should be corrected.

- Each corrected line counts for **3 out of 15 marks.**
- For any partially correct answer, please mark evenly.
- If the total mark contains fractions, please round up to the nearest integer.

Question 2. (15 marks)
Suppose a method is defined as follow:

```
public static void secretMethod(int a, int b){
 char code [] = {'h','e','a','r','t'};
 int c=0;
 do {
     System.out.print("#");
     if (c<5) c++;
     if (a%3==0) a = a+3;
     else a++;
     System.out.print(code[c]);
 } while (a<b);

 int i = (a+b)%5;
 if (i<0) i=0;
 System.out.println(a+""+b+code[i]);
}
```

What is the output from the following method call?Briefly justify your answer!

(a) `secretMethod(5,2)`

(b) `secretMethod(1,4)`

Answer

(a) `secretMethod(5,2)`

`initial variables value: a=5, b=2, c=0`
`the variables value after each iteration:`

iter	a	b	c	console
1	6	2	1	#e
	i			
	3			#e62r

`So, the output is: #e62r`

(b) `secretMethod(1,4)`

`initial variables value: a=1, b=4, c=0`
`the variables value after each iteration:`

iter	a	b	c	console
1	2	4	1	#e
2	3	4	2	#e#a
3	6	4	3	#e#a#r
	i			
	0			#e#a#r64h

`So, the output is: #e#a#r64h`

Marking Scheme

- Part (a) counts for **5 out of 15 marks.**
- Part (b) counts for **10 out of 15 marks.**
- For any partially correct answer, please mark evenly.
- Halve the mark if no justification is provided (i.e. showing the output only)
- If the total mark contains fractions, please round up to the nearest integer.

Question 3. (15 marks)

Suppose a method is defined as follow:

```
public static void mystery(int a, int b){
 char code [] = {'l','o','v','e'};
 boolean whenever [] ={true,false,true,true,false};
 for(int i=0;i<5;i+=2){
     if(!whenever[i]) System.out.print("*");
     if (a<b){
         int c= (a+b)%4;
         System.out.print(code[c]);
         a=a+2;
     }
 }
 System.out.println(a+b+""+b+"*");
}
```

What is the output from the following method call?Briefly justify your answer!

(a) `mystery(2,5)`

(b) `mystery(10,2)`

Answer

(a) `mystery(2,5)`

`initial variables value: a=2, b=5, i=0`
`the variables value after each iteration:`

iter	i	a	b	c	console
1	2	4	5	3	e
2	4	6	5	1	eo
3	6	6	5	1	eo*
					eo*115*

`So, the output is: eo*115*`

(b) `mystery(10,2)`

`initial variables value: a=10, b=2, i=0`
`the variables value after each iteration:`

iter	i	a	b	c	console
1	2	10	2	-	
2	4	10	2	-	
3	4	10	2	-	*
					122

`So, the output is: *122*`

Marking Scheme

- Part (a) counts for **10 out of 15 marks.**
- Part (b) counts for **5 out of 15 marks.**
- For any partially correct answer, please mark evenly.
- Halve the mark if no justification is provided (i.e. showing the output only)
- If the total mark contains fractions, please round up to the nearest integer.

Question 4. (20 marks)

Complete the following method definition:

```
public static void squareStars(int n){
  //your code goes here

}
```

In order to generate expected output with its respective method call as follow:

```
Method Call:    squareStars(1)      squareStars(2)      squareStars(3)
Exp.Output:         ***                 *****               *******
                    * *                 *   *               *     *
                    ***                 *   *               *     *
                                        *   *               *     *
                                        *****               *     *
                                                            *     *
                                                            *******
```

Please note that your code has to work with parameter n, any integers greater than 0.

Answer

```
public static void squareStars(int n){
   //step 1
    for (int i=0;i< (2*n+1);i++) System.out.print("*");
    System.out.println();
    //step 2
    for (int i=0;i< (2*n-1);i++) {
        //step 3
        System.out.print("*");
        //step 4
        for (int j=0;j< 2*n-1;j++) System.out.print(" ");
        //step 5
        System.out.print("*");
        System.out.println();
    }
    //step 6
    for (int i=0;i< (2*n+1);i++) System.out.print("*");

}
```

Marking Scheme

The code contains **6 (six)** steps.

- Each step except step 3 and step 5 counts for **4 out of 20 marks.**
- Each step of steps 3 and 5 counts for **2 out of 20 marks.**
- For any partially correct answer, please mark evenly.
- If the total mark contains fractions, please round up to the nearest integer.

Question 5. (20 marks)
Given an input file namely **hujan_bulan_juni.txt** shown below:

```
Hujan Bulan Juni
tak ada yang lebih tabah
dari hujan bulan Juni
dirahasiakannya rintik rindunya
kepada pohon berbunga itu
tak ada yang lebih bijak
dari hujan bulan Juni
dihapusnya jejak-jejak kakinya
yang ragu-ragu di jalan itu
tak ada yang lebih arif
dari hujan bulan Juni
dibiarkannya yang tak terucapkan
diserap akar pohon bunga itu
(1989)
Sapardi Djoko Damono
```

Create a complete java code, namely **Hujan.java** with a main method to read the file and print out the number of times the word "hujan" occurs in the text, regardless of capitalization.

Answer

```
Filename:Hujan.java
//step 1
import java.util.Scanner;
public class Hujan{
    //step 2
    public static void main(String [] args){
        //step 3
        Scanner read = new Scanner(Main.class.getResourceAsStream
        ("hujan_bulan_juni.txt"));
        int count=0;
        //step 4
        while (read.hasNext()){
```

```
            //step 5
            String line= read.nextLine();
            //step 6
            String [] splitLine= line.split(" ");
            //step 7
            for(int i=0;i<splitLine.length;i++){
                if (splitLine[i].equalsIgnoreCase("hujan")) count++;
            }
        }
        //step 8
        System.out.println(count);

    }
}
```

Marking Scheme

The code contains **8 (eight)** steps.

- Each step except step 7 counts for **2 out of 20 marks.**
- Step 7 counts for **6 out of 20 marks.**
- For any partially correct answer, please mark evenly.
- If the total mark contains fractions, please round up to the nearest integer.

Question 6. (30 marks)

To foster a competitive atmosphere, **Waskito School**, a private school in Surabaya, initiate to open Quick Express class. Selected students in this class will finish their senior high school within 1 year only. Therefore, the school carried out IQ test to the new students. The students will pass the fast track class entry exam if their IQ $\geq$ 120 and greater than the average IQ of their peers. To automate the selection process, you are asked to help the school developing a complete java program. namely **Waskito.java** allow the school to enter the students names and their IQ Score, count how many

students passed the Quick Express Class, and spot the student with highest IQ score. More specifically, your code should:

a. Ask the users how many students taking the entry test

b. For each student, ask the user to enter the student name and his/her IQ score

c. Store the student names and their respective IQ score into two separated array variables

d. Count the average and highest IQ score

e. Count the number of students passed the Quick Express Class

f. Print the number of students pass the Quick Express Class, the name of student with the highest IQ score, and the highest IQ score

Please note that you should ask the user input from console (System.in).

Answer

```
Filename:Waskito.java
//step 1
import java.util.Scanner;
public class Waskito{
    //step 2
    public static void main(String [] args){
        //step 3
        Scanner read = new Scanner(System.in);
        System.out.print("Enter number of students:");
        //step 4
        int n= Integer.parseInt(read.nextLine());
        //step 5
        String studentName [] = new String[n];
        //step 6
        int studentIQScore []= new int[n];
```

```
//step 7
int sumIQScore=0;

//step 8
for (int i=0;i<n;i++){
    //step 9
    System.out.print("name student "+(i+1)+":");
    studentName[i] =read.nextLine();
    //step 10
    System.out.print("IQ Score student "+(i+1)+":");
    studentIQScore[i] =Integer.parseInt(read.nextLine());
    //step 11
    sumIQScore+=studentIQScore[i];
}

//step 12
double avgIQScore=(double)sumIQScore/n;
//step 13
int countPass=0;
//step 14
int highestIQScore=0;
String highestIQName="";
//step 15
for (int i=0;i<n;i++){
    //step 16
    if (studentIQScore[i]>120 &&
    studentIQScore[i]>avgIQScore) countPass++;
    //step 17
    if (studentIQScore[i]>highestIQScore){
        highestIQScore =studentIQScore[i];
        highestIQName=studentName[i];
    }
}
//step 18
System.out.println("Avg IQ:"+avgIQScore+"; Number of
Pass:"+countPass+"; Highest IQ:"+highestIQScore+"
("+highestIQName+")");
```

```
    }
}
```

Marking Scheme
The code contains **18 (eighteen)** steps.

- For steps 1-15, each step counts for **1 out of 30 marks.**
- For steps 16 and step 18, each step counts for **3 out of 30 marks.**
- Step 17 counts for **9 out of 30 marks.**
- For any partially correct answer, please mark evenly.
- If the total mark contains fractions, please round up to the nearest integer.

Question 7. (30 marks)
Today is very defining moment for Juminten, a final year, computer science student in Coding University. She finally can make her dream, being accepted as exchange student in The University of Edinburgh, Scotland, really comes true. This morning She is going to catch her very first flight in her life. Using Emprit Airways, she will be leaving Soekarno Hatta Airport in Jakarta to Heathrow Airplot in London and having connecting flight to Edinburgh afterward. She brings three travel bags, labeled as A, B, and C. Since she is allowed to bring exactly one bag to the cabin, she has to check in the other two bags. However, the Emprit Airways has restriction that the total weight of checked baggage is not allowed to exceed D Kgs and the maximum weight of cabin bag is E Kgs. Given the **baggage.txt** file as input, help Juminten to check whether she can bring all her bags or not.

```
Sample Input:
5
1 1 1 15 5
8 7 6 15 5
8 5 7 15 6
19 1 5 20 5
9 5 7 15 6
```

```
Sample expected Output:
YES
NO
YES
YES
NO
```

The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows. Each test case contains a single line of input, five space separated integers A,B,C,D,E. Create a complete java program, namely **Emprit.java.**

Answer

```
Filename:Emprit.java
//step 1
import java.util.Scanner;
public class Waskito{
    //step 2
    public static void main(String [] args){
        //step 3
        Scanner read= new Scanner(Main.class.getResourceAsStream
        ("baggage.txt"));
        //step 4
        int t= read.nextInt();
        //step 5
        for(int i=0;i<t;i++){
            //step 6
            int a= read.nextInt();
            //step 7
            int b= read.nextInt();
            //step 8
            int c= read.nextInt();
            //step 9
            int d= read.nextInt();
            //step 10
```

```
            int e= read.nextInt();
            //step 11
            System.out.println( a<=e && b+c<=d || b<=e && a+c<=d
            || c<=e && a+b<=d?"YES":"NO");
        }

    }
}
```

Marking Scheme

The code contains **11 (eleven)** steps.

- For steps 1-10, each step counts for **1 out of 30 marks.**
- Step 11 counts for **20 out of 30 marks.**
  - Step for printing "YES" and "NO" count for **2 marks**
  - The logical test counts for **18 marks**
- For any partially correct answer, please mark evenly.
- If the total mark contains fractions, please round up to the nearest integer.