**QUESTION 1.**

In a system designed to work out the tax to be paid: An employee has $4000 of salary tax free. The next $1500 is taxed at 10%. The next $28000 is taxed at 22%. Any further amount is taxed at 40%. Which of these groups of numbers would fall into the same equivalence class?

1. $4800; $14000; $28000
2. $5200; $5500; $28000
3. $28001; $32000; $35001
4. $5800; $28000; $32000

**QUESTION 2**

For the example from the previous question, provide test cases required to ensure boundary testing.

**QUESTION 3.**

Which values must X and Y have in the table below to make it a L4(23) orthogonal array? Explain your answer.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 | 2 | 3 |
| 1 | 1 | 1 | 1 |
| 2 | 1 | 2 | 2 |
| 3 | 2 | 1 | X= |
| 4 | 2 | 2 | Y= |

Explanation:

**QUESTION 4**

Suppose you have a web-based system which supports the following environments:

* Browser: Internet Explorer, Mozilla Firefox
* Client operating system: Windows7, Mac iOS
* Server operating system: Windows, Linux

1. How many test cases would be needed to test this system using pairwise testing? How many test cases are needed for exhaustive testing of the system? Explain your answer
2. For this system, make a table of inputs of test cases for pairwise testing of the possible configurations with a minimum number of test cases.

**QUESTION 5.**

Use the following scenario to answer the questions below:

A marketing company wishes to develop a software system that will recommend products to customers based on the following their characteristics (of the customers): Gender, City Dweller, and age group: Young (under 30), Middle Age (between 30 and 60), and Old (over 60). The company has four products (W, X, Y and Z) to test market. Product W will appeal to female city dwellers. Product X will appeal to young females. Product Y will appeal to Male middle aged shoppers who do not live in cities. Product Z will appeal to all but older females

1. Suggest a testing strategy and justify your answer.
2. Develop test cases using the proposed approach. Cleary highlight your tests.

**QUESTION 6.**

The following program is used in a hypothetical retail situation. The owner of a shop has decided that her staff can have a 10 percent discount on all their purchases. If they spend more than $15, then the total discount is increased by 50 cents. The price of each item being purchased is input into the program. When -1 is entered, the total price is displayed, as well as the calculated discount and the final price to pay. For example, the values $5.50, $2.00 and $2.50 are input, equaling $10.00. The total discount would equal $1.00 (10% of $10.00), with the total price to pay equaling $9.00. A second example would have purchases of $10.50 and $5.00, equaling $15.50. In this case, as the total value is over $15, the discount would be $2.05 (10% of $15.50 is $1.55, plus 50 cents as the original total is over $15), meaning that the total price to pay would be $13.45.

The source code, written in pseudo code, for a program which has been written to perform the task described above, is shown below:

1 program Example()

2 var staffDiscount, totalPrice, finalPrice, discount, price

3 staffDiscount = 0.1

4 totalPrice = 0

5 input(price)

6 while(price != -1) {

7 totalPrice = totalPrice + price

8 input(price)}

9 print("Total price: " + totalPrice)

10 if(totalPrice > 15.00) then

11 discount = (staffDiscount \* totalPrice) + 0.50

12 else

13 discount = staffDiscount \* totalPrice

14 print("Discount: " + discount)

15 finalPrice = totalPrice – discount

16 print("Final price: " + finalPrice)

17 endprogram

1. Draw a control flow graph (CFG) to represent the program’s code
2. Write down the minimal set of tests to achieve statement coverage.
3. What is the maximum number of test cases needed to ensure branch coverage of this code? **Explain your answer.**
4. Provide a minimal set of test cases to ensure branch coverage.
5. Annotate your CFG with “definition” and “use” information of each module variable.
6. Write down the test cases necessary to ensure definition-use testing with respect to variable “price.”