

<b><u>SURNAME:</u></b>		<b><u>NAME</u></b>		<b><u>D1</u></b>
<b><u>STUDENT ID:</u></b>				
<b><u>PROFESSOR:</u></b>				

Question 1	Answers
<p>Given <b>X</b> and <b>Y</b>, two numbers of 8 bits in 2's Complement (2C), perform the operations: <b>X + Y</b> and <b>X - Y</b>. In each case, indicate if the operation generates overflow.</p> <p><b>X</b> = 00111111<sub>2c</sub>  <b>Y</b> = 11000001<sub>2c</sub></p>	<p>X+Y = _____ Overflow: _____</p> <p>X-Y = _____ Overflow: _____</p>
<p><i>Report the most relevant steps</i></p>	

Question 2
<p>Determine the truth table for the Boolean function: <b>f(A, B, C) = A × (B' × C)</b></p>
<p><i>Report the most relevant steps</i></p>

Question 3
<p>Describe the main characteristics of the main memory (or internal memory) and secondary memory (or mass or external storage), indicating memory size and access time relatively.</p>

#### Question 4 (PROGRAMMING)

Write a C program that allows the management of a commercial warehouse.

The program must process the information contained in 2 different files:

- **warehouse:** It contains the list of available products in the warehouse using the format:  

**<Product Name> <Quantity>**
- **requests:** It contains the list of requests for the products from the warehouse. It contains one  

**<Product Name>**

 per line

The filenames are passed as command line arguments. For example:

```
c:\> myprog warehouse.txt requests.txt
```

Where, **the first argument is the filename of the warehouse** and the second argument is **the filename of the requests**.

Consider the following assumptions:

- In the **warehouse** file, **<Product Name>** is maximum 15 characters without space, **<Quantity>** is an integer. They are separated by one space. In the **requests** file, **<Product Name>** is maximum 15 characters without space.
- The number of products in the **warehouse** is not known a priori, but at most it is 100.
- The number of requests in the **requests** file is not known a priori, but contains maximum 200 **distinct** products, one for each line.
- The product indicated in **requests** file may not be present in **warehouse** file, and the files content is always correct.

The program must:

- 1) Take into account the incoming requests from the **requests** file and update (reduce) the quantity of available products in the warehouse.
- 2) If a product is requested for a number of times higher than the warehouse stock, display a message showing the **additional quantity** of the product required by the stock for that product.
- 3) Check if there are products that are not present in the warehouse (i.e., whose name does not appear in the first file), displaying them in a list without quantities.
- 4) Replace the **warehouse** file with the new quantities of the products at the end of the process.

For example, in the files contain the following content:

<b>warehouse.txt</b>	<b>requests.txt</b>		<b>warehouse.txt (After the execution)</b>
Book 10 Toy 20 Bike 3 Ball 100 Tablet 20 Guitar 1	Book Toy Bike Ball Guitar Ski Bike Ski Bike Book Bike Guitar Guitar Backsword		Book 8 Toy 19 Bike 0 Ball 99 Tablet 20 Guitar 0

Output from the previous example:

The excess quantity of Bike required is: 1

The excess quantity of Guitar required is: 2

The following products are not in the warehouse stock: Ski, Backsword.

<b><u>SURNAME:</u></b>		<b><u>NAME</u></b>		<b><u>D2</u></b>
<b><u>STUDENT ID:</u></b>				
<b><u>PROFESSOR:</u></b>				

Question 1	Answers
<p>Given <b>X</b> and <b>Y</b>, two 8 bits numbers in Sign &amp; Magnitude (SM), perform the operations: <b>X + Y</b> and <b>X - Y</b>. In each case, indicate if the operation generates overflow.</p> <p><b>X</b> = 00111111<sub>SM</sub></p> <p><b>Y</b> = 11000001<sub>SM</sub></p>	<p>X+Y = _____ Overflow: _____</p> <p>X-Y = _____ Overflow: _____</p>
<p><i>Report the most relevant steps</i></p>	

Question 2
<p>Determine the truth table for the Boolean function: <b>f(A, B, C) = A × (B' × C)'</b></p>
<p><i>Report the most relevant steps</i></p>

Question 3
<p>List and briefly explain the steps of the instruction execution cycle in a microprocessor.</p>

**Question 4 (PROGRAMMING)**

Write a C program that allows the management of a commercial warehouse.

The program must process the information contained in 2 different files:

- **warehouse:** It contains the list of available products in the warehouse using the format:  

**<Product Name> <Quantity>**
- **accesses:** It contains the list, in order of time, of withdrawals and supplies of products at the warehouse, through a format that provides for each row:
  - **Withdraw** of a product: 

**<Product Name>**
  - **Supply** of a product: 

**+<Product Name>**

The filenames are passed as command line arguments. For example:

```
c:\> myprog warehouse.txt accesses.txt
```

Where, the first argument is the filename of the **warehouse** and the filename of **accesses** is the second.

Consider the following assumptions:

- In **warehouse** file, **<Product Name>** is maximum 15 characters without space, **<Quantity>** is an integer. They are separated by one space. In **accesses** file, **<Product Name>** is maximum 15 characters without space and there is **no** space between the **'+'** symbol **indicating supply of a product and the <Product Name>**.
- The number of products is not known a priori, but at most it is 100.
- The number withdraws and supplies in the **accesses** file is not known a priori. But the **accesses** file contains a maximum of 200 **distinct** products.
- The **<Product Name>** indicated in **accesses** file may not be present in **warehouse** file, and the files content is always correct.

The program must update the **warehouse** file appropriately:

- 1) Taking into account the incoming requests from the **accesses** file and update the quantity of available products according to the request is **withdraw** or **supply**.
- 2) Displaying the total number per product of unsuccessful withdraws in the case of insufficient quantity in the warehouse.
- 3) In case of supply after a failed withdraw, the quantity of the product must be updated without altering the previous failed withdraws.
- 4) Withdraws and supplies requests for **NOT** present products the warehouse must be ignored.
- 5) Replace the **warehouse** file with the final products and quantities after all withdraws and supplies in **accesses** file are processed.

For example, in the files contain the following content:

warehouse.txt	accesses.txt	EFFECT of each row in accesses.txt	warehouse.txt (After the execution)
Book 10	Book	Book 10 -> 9	Book 8
Toy 20	+Toy	Toy 20 -> 21	Toy 21
Bike 3	Book	Book 9 -> 8	Bike 1
Ball 100	Bike	Bike 3 -> 2	Ball 100
tablet 20	Bike	Bike 2 -> 1	tablet 20
Guitar 2	Bike	Bike 1 -> 0	Guitar 0
	+Backsword	(Supply Ignored)	
	Bike	Failed withdraw	
	Bike	Failed withdraw	
	+Bike	Bike 0 -> 1	
	Guitar	Guitar 2 -> 1	
	Ski	(Withdraw Ignored)	
	Guitar	Guitar 1 -> 0	
	Guitar	Failed withdraw	

Output from the previous example:

Failed requests for the Bike: 2

Failed requests for the Guitar: 1