

<b><u>SURNAME:</u></b>		<b><u>NAME:</u></b>		<b><u>C1</u></b>
<b><u>STUDENT ID:</u></b>				
<b><u>PROFESSOR:</u></b>				

**Question 1**

Given two integers,  $n_1$  and  $n_2$ , expressed in base 16, represent them in two's complement (**2C**) using 8 bits. Compute the addition (always in **2C**) and verify an eventual overflow.

$n_1 = 73h$

$n_2 = 7Fh$

*Binary result:*

$n_1$  in 2C:

$n_2$  in 2C:

Addition:

Overflow:

*Please, show the most significant steps to determine the result*

**Question 2**

Given two Boolean functions  $X = (b + c') * a'$ ,  $Y = [(b + c')' * a] + b$ , demonstrate their equivalence.

Note: ' = NOT; +=OR; \*=AND;

*Please, show the most significant steps to determine the result*

**Question 3**

Describe the main features of the system bus in a computer based system.

**Question 4 (PROGRAMMING)**

General *von Weyrother* wants to convince General *Kutuzov* of the usefulness of a strategic plan that precisely manages the movements of the troops. To this aim, he needs a program that, given the map of the initial deployment and the movements of the troops, generates the final positions.

Write a **C** program that manages this process. The program must receive from the command line the next elements in order: **(1)** the name of the file with the initial map, **(2)** the name of the file holding the movements, and **(3)** the name of the file to write the final map on.

The program must write to the file, specified in **(3)**, the final map generated by the movements of the troops. The initial and final maps represent the battlefield as a  $N \times N$  square (**N** is a predefined symbolic **constant** defined by a **#define**).

Each element of the matrix represents the status of the corresponding area: the empty spaces correspond to the **'0'** symbol, while the position of each troop is represented by an alphabetical symbol that uniquely distinguishes it. A troop occupies a sequence of cells either in a row or in a column (see the example). The troop movement file has the following format: each line represents a movement:

`<troop> <direction> <movement>`

`<troop>` is a **character** associated with a troop, `<direction>` is a **character** pointing the movement direction: **'N'** (up), **'S'** (down), **'E'** (right), **'W'** (left). Finally, `<movement>` is an **integer** representing the number of boxes whose troops are to be moved.

One space separates all values. The number of lines in the file is **NOT** known a priori.

If a part or all of a troop leaves the battlefield, the leaked part cannot return. If a troop leaves the battlefield completely, print it on the screen.

Assume that the contents of the files are always correct and that collisions between troops cannot occur.

**Example of the input files: (N=8)**

map1.txt	movement1.txt	movement2.txt
00000000	a E 2	b E 1
000aaa00	a S 1	c W 2
00b00000	b N 2	
00b00000	c S 3	
00b00000		
00b00000		
00000cc0		
00000000		

**Examples of program execution:**

**Examples of generated files:**

<pre>c:\&gt;move.exe map1.txt movement1.txt map2.txt</pre> <p>Troop c has left the battlefield.</p>	<pre>map2.txt 00b00000 00b00000 00b00aaa 00b00000 00000000 00000000 00000000 00000000</pre>
<pre>c:\&gt;move.exe map1.txt movement2.txt map3.txt</pre>	<pre>map3.txt 00000000 000aaa00 000b0000 000b0000 000b0000 000b0000 000cc000 00000000</pre>

<b><u>SURNAME:</u></b>		<b><u>NAME:</u></b>		<b><u>C2</u></b>
<b><u>STUDENT ID:</u></b>				
<b><u>PROFESSOR:</u></b>				

Question 1	
<p>Given two integers, <math>n_1</math> and <math>n_2</math>, expressed in base 16, represent them in two's complement (<math>CA_2</math>) with 8 bits, compute the addition (always in <math>CA_2</math>) and verify an eventual overflow.</p> <p><math>n_1 = 19</math>  <math>n_2 = D9</math></p>	<p><i>Binary Result:</i></p> <p><math>n_1</math> in <math>CA_2</math>:  <math>n_2</math> in <math>CA_2</math>:          Addition:          Overflow:</p>
<p><i>Please, show the most significant steps to determine the result</i></p>	

Question 2
<p>Given two boolean functions <math>X = (b + c') * a'</math>, <math>Y = [a * (b + c')'] + b'</math>, demonstrate their equivalence.</p> <p><i>Please, show the most significant steps to determine the result</i></p>

Question 3
<p>Describe the main logical blocks composing the CPU briefly.</p>

**QUESTION 4 (PROGRAMMING)**

General *von Weyrother* wants to convince General *Kutuzov* of the usefulness of a strategic plan that precisely manages the movements of the troops. To this aim, he needs a program that, given the map of the initial deployment and the movements of the troops, generates the final positions.

Write a program in **C** language that can manage this process. The program must receive from the command line the next elements in order: **(1)** the name of the file with the initial map, **(2)** the name of the file holding the movements, and **(3)** the name of the file to write the new map on.

The program must write to the file, specified in **(3)**, the new map generated by the movements of the troops. The initial and final maps represent the battlefield as a square  $N \times N$  (with **constant**  $N$  known a priori and defined by a **#define**).

Each element of the matrix represents the status of the corresponding area: the empty spaces correspond to the **'0'** symbol, while the position of each troop is represented by an alphabetical symbol that uniquely distinguishes it. A troop occupies a sequence of squares either in a row or in a column (see the example). The troop movement file has the following format: each line represents a movement like:

<troop> <movement> <direction>

<troop> is the **character** associated with a troop. <movement> is an **integer** number representing the number of boxes whose troops are to be moved, and <direction> is a **character** pointing the movement direction: **'u'** (up), **'d'** (down), **'r'** (right), **'l'** (left). All values are separated by one space. The number of lines in the file is **NOT** known a priori.

If a part or all of a troop leaves the battlefield, the leaked part cannot return. Assume that the contents of the files are correct and that collisions between troops cannot occur.

If a troop leaves the battlefield completely, print it on the screen.

**Example of input files: (N=8)**

map1.txt	movement1.txt	movement2.txt
00000000	a 2 r	b 1 r
000aaa00	a 1 d	c 2 l
00b00000	b 2 u	
00b00000	c 3 d	
00b00000		
00b00000		
00000cc0		
00000000		

**Example of program execution:**

**Example of generated files:**

c:\>move.exe map1.txt movement1.txt map2.txt	map2.txt
Troop c has left the battlefield.	00b00000 00b00000 00b00aaa 00b00000 00000000 00000000 00000000 00000000
c:\>move.exe map1.txt movement2.txt map3.txt	map3.txt
	00000000 000aaa00 000b0000 000b0000 000b0000 000b0000 000cc000 00000000