1. A car-park charges **5.50** Saudi Riyals per hour or any fraction of an hour. Write a **64-bit** x86 assembly language program that prompts for and read the number of hours and the number of minutes [both integers] a car has been parked. It then calculates and displays the parking charge in Saudi Riyals.

Use a procedure **computeCharge** to calculate the charge.

**Hint:** Your **computeCharge** procedure must use the standard C mathematical function **ceil**

|  |  |  |  |
| --- | --- | --- | --- |
| C function | Mathematical Notation | Example | Comment |
| ceil(x) | ⎡ x ⎤ | ceil(45.1) = 46.0  ceil(-7.9) = -7.0 | Returns the smallest integral value (of type double) greater or equal to x |

**Note:**

* The **computeCharge** procedure must not contain input/output statements such as **scanf\_s** or **printf\_s**; the printing of the results must be in the **main** procedure.
* The **main** procedure must display an appropriate error message and terminate if the input is invalid [Assume that the user will not enter a floating-point value].

Sample program runs:

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

1. Write a **32-bit** x86 program that prompts for and reads, row-wise, an **unsigned** 2D integer array of size 4\*6. It then passes the array to a **macro** that returns the index of the first column whose sum is the maximum of the column sums. The main procedure then displays the returned index.

**Sample program run:**

Enter the elements of a 4\*6 integer array row-wise:

1 4 2 0 5 0

7 3 5 4 8 0

5 4 1 1 3 2

3 6 8 5 4 5

The index of the first column with maximum sum is: 4

Note: Your macro must be general; it must work for any unsigned integer 2D-array of size 4 \* 6