

שאלות בנוגע לתרגיל נא להפנות דרך פורום הקורס שנפתח במיוחד לשם כך:

<https://piazza.com/biu.ac.il/fall2018/89110>

אם לא נענתה תשובה תוך 24 שעות, נא לשלוח אלי (דור) מייל עם לינק לדיון הרלוונטי ואענה. המייל הוא: [nisimdor@gmail.com](mailto:nisimdor@gmail.com). בכל מייל יש לציין שם, שם משתמש, מס' קורס, וקבוצת תרגול.

- מועד פרסום: 11.11.2018
- מועד אחרון להגשה: 18.11.2018
- יש לשלוח את הקבצים באמצעות האתר: <http://help.cs.biu.ac.il/submit.htm>
- שם ההגשה של התרגיל: ex3
- יש להקפיד מאוד על כל הוראות עיצוב הקלט והפלט, כמפורט בכל סעיף וסעיף. על הפלט להיראות בדיוק כמו בדוגמאות. אין להוסיף או להשמיט רווחים או תווים אחרים ואין להחליף אותיות גדולות בקטנות או להיפך ® אי-הקפדה על פרטים אלה עלולה לגרור ירידה משמעותית ביותר בציון התרגיל עד כדי 0. ראו הוזהרתם!
- להזכירכם, העבודה היא אישית. "עבודה משותפת" דינה כהעתקה.
- אין להדפיס למסך שום דבר מעבר למה שנתבקש בתרגיל.
- יש לוודא שהתרגיל מתקמפל ורץ על השרתים באוניברסיטה (u2) ללא שגיאות/אזהרות.
- אתם יכולים לעבוד עם כל עורך טקסטואלי שאתם מעדיפים. להזכירכם – pico בשרתי linux שבמעבדות; notepad ב-windows; או בסביבת פיתוח ייעודית.

**הקפידו על כתיבה לפי קובץ ה-Coding-Style שבאתר הקורס!!**

## הוראות כלליות לתרגיל

כתבו תכנית בעלת הזרימה הבאה:

1. הדפס: Please enter the mission number:
2. קלוט מספר שלם (ניתן להניח שיתקבל מספר שלם)
3. כל עוד המספר בין 1 ל-4 בצע:
  - i. אם התקבל 1 בצע את **משימה 1**.
  - ii. אם התקבל 2 בצע את **משימה 2**.
  - iii. אם התקבל 3 בצע את **משימה 3**.
  - iv. אם התקבל 4 בצע את **משימה 4**.
- b. הדפס: Please enter the mission number:
- c. קלוט מספר שלם (ניתן להניח שיתקבל מספר שלם)
4. צא מהתוכנית

ניתן להניח שכל קלט המתקבל במשימות הינו **תקין** אלא אם כן נאמר אחרת!  
**יש להשתמש בלבד בחומר הנלמד עד כה בתרגולים (עד לולאות + BitWise כולל).**

## שים/י ♥

כאשר מתרחשת שגיאה בקוד (משמע, הודפסה הודעת שגיאה) עליכם לחזור לתוכנית הראשית ולא לבצע את המשימה מחדש!

## משימה 1 – Full Adder

We would like your help to create a basic adder. However, this adder, should work in any base (from base 2 to base 10).

The first line of input contains an integer  $b$ , the base.

The remaining 2 lines will contain two non-negative numbers expressed in the given base. (You can assume that the numbers are indeed in base  $b$ ).

You must output the solution for the given addition problem, as shown in the sample input and output.

### Constraints

(For every constraint you must include **#define** in your code)

- $2 \leq b \leq 10$

(In the code, the following lines are expected to be found:

```
#define MAX_BASE 10
```

```
#define MIN_BASE 2)
```

If the input does not contain a base in the given constraint you **must output** the following:

```
ERROR - The base <base> is not between 2 and 10 <newline>
```

For example, if we are given  $b = 12$  we will output

```
ERROR - The base 12 is not between 2 and 10 <newline>
```

Where <newline> is the printing '\n' in the printf function

- The numbers to be added can contain up to 8 digits.  
(There is no need to check if a given number contains more than 8 digits)  
There is no need to include a specific **#define** statement in the code, only use the appropriate variable type.

### Input Format

The input format will be as follows:

Please enter the base of the problem:

<b>

Please enter the first number in base <b>:

<num1>

Please enter the second number in base <b>:

<num2>

Please make sure that the output messages are as described above. (The system is case-sensitive).

- <b> - The base of the problem (remember that  $2 \leq b \leq 10$ )
- <num1> <num2> - The two numbers in base <b>.

## Output Format

The output format will be as follows:

The result of  $\langle \text{num1} \rangle + \langle \text{num2} \rangle$  in base  $\langle b \rangle$  is  **$\langle \text{result} \rangle$**   $\langle \text{newline} \rangle$

- **$\langle \text{result} \rangle$**  - The solution to the addition problem (The result of adding the operands under the given base) .

## Sample Input

Please enter the base of the problem:

**8**

Please enter the first number in base  $\langle b \rangle$ :

**715**

Please enter the second number in base  $\langle b \rangle$ :

**501**

## Sample Output

The result of  $715 + 501$  in base 8 is **1416**  $\langle \text{newline} \rangle$

## Explanation

This is a base-8 problem.

The numbers are 461 and 321 in decimal (i.e. base 10).

When adding these numbers, the result is 1416 (in base 8), which is 782 in decimal.

Thus the expected output is:

$715 + 501 = \mathbf{1416}$  in base 8  $\langle \text{newline} \rangle$

## Sample Input

Please enter the base of the problem:

**12**

## Sample Output

ERROR - The base 12 is not between 2 and 10  $\langle \text{newline} \rangle$

## Explanation

We are given a base 12, which is out of the permitted range.

## Sample Input

Please enter the base of the problem:

**2**

Please enter the first number in base <b>:

**100111**

Please enter the second number in base <b>:

**01010**

### **Sample Output**

The result of 100111 + 1010 base 2 is **110001**<newline>

### **Explanation**

This is a base-2 problem (binary).

The numbers are 39 and 10 in decimal (i.e. base 10).

When adding these numbers, the result is 110001 (in base 2), which is 49 in decimal.

Bob has a *strange counter*. At the first second, it displays the number 3. Each second, the number displayed by the counter decrements by 1 until it reaches 1.

The counter counts down in cycles. In next second, the timer resets to  $2 \times$  the initial number for the prior cycle and continues counting down. The diagram below shows the counter values for each time  $t$  in the first three cycles:

| time | value | time | value | time | value |
|------|-------|------|-------|------|-------|
| 1    | 3     | 4    | 6     | 10   | 12    |
| 2    | 2     | 5    | 5     | 11   | 11    |
| 3    | 1     | 6    | 4     | 12   | 10    |
|      |       | 7    | 3     | 13   | 9     |
|      |       | 8    | 2     | 14   | 8     |
|      |       | 9    | 1     | 15   | 7     |
|      |       |      |       | ...  | ...   |
|      |       |      |       | 21   | 1     |

Find and print the value displayed by the counter at time  $t$ .

### Input Format

The input format will be as follows:

Please enter the time:

<t>

Please make sure that the output messages are as described above. (The system is case-sensitive).

- <t> - A single integer denoting the value of  $t$  (The time).

### Constraints

(For every constraint you must include **#define** in your code)

- $1 \leq t \leq 10^7$

If the input is a negative time or zero you **must output** the following:

ERROR - The time <t> is invalid<newline>

For example, if we are given  $t = -12$  we will output

ERROR - The time -12 is invalid<newline>

Where <newline> is the printing '\n' in the printf function

### Output Format

The output format will be as follows:

The time in the timer after <t> seconds is <time> <newline>

- <time> - The value displayed by the strange counter at the given time **t**.

### Sample Input

Please enter the time:

**4**

### Sample Output

The time in the timer after 4 seconds is **6** <newline>

### Explanation

Time **t = 4** marks the beginning of the second cycle. It is double the number displayed at the beginning of the first cycle:  $2 \times 3 = 6$ . This is also shown in the diagram in the *Problem Statement* above.

### Sample Input

Please enter the time:

**15**

### Sample Output

The time in the timer after 15 seconds is **7** <newline>

### Explanation

This is also shown in the diagram in the *Problem Statement* above.

A *modified Kaprekar number* is a positive whole number with a special property. If you square it, then split the number into two integers and sum those integers, you have the same value you started with.

Consider a positive whole number  $n$  with  $d$  digits. We square  $n$  to arrive at a number that is either  $2 \times d$  digits long or  $(2 \times d) - 1$  digits long. Split the square into two parts,  $l$  and  $r$ . The right hand part,  $r$  must be  $d$  digits long. The left is the remaining substring. Convert those two substrings back to integers, add them and see if you get  $n$ .

For example, if  $n = 5$ ,  $d = 1$  then  $n^2 = 25$ . We split that into two numbers 2 and 5. We test  $2 + 5 = 7 \neq 5$ , so this is not a modified Kaprekar number. If  $n = 9$ , still  $d = 1$ , and  $n^2 = 81$ . This gives us  $1 + 8 = 9$  the original  $n$ .

**Note:**  $r$  may have leading zeros.

Given two positive integers  $p$  and  $q$  where  $p$  is lower than  $q$ , write a program to print the modified Kaprekar numbers in the range between  $p$  and  $q$ , inclusive. The numbers must be printed in an ascending order.

### **Input Format**

The input format will be as follows:

Please enter the lower limit of the range (p):

<p>

Please enter the upper limit of the range (q):

<q>

Please make sure that the output messages are as described above. (The system is case-sensitive).

- <p> - The lower integer limit .
- <q> - The upper integer limit.

**Note:** Your range should be inclusive of the limits.

### **Constraints**

(For every constraint you must include **#define** in your code)

$$0 < p < q \leq 10000$$

If  $p$  and  $q$  are not in the given range, or if  $p \geq q$ , you must output the following line



ERROR - The range <p> to <q> isn't valid<newline>

For example, if we are given **p=7 and q=1** we will output

ERROR - The range 7 to 1 isn't valid<newline>

Where <newline> is the printing '\n' in the printf function

### Output Format

The output format will be as follows:

The numbers between <p> and <q> are:

<n<sub>1</sub>>

<n<sub>2</sub>>

...

<n<sub>k</sub>><newline>

- <n<sub>i</sub>> - The i<sup>th</sup> number between <p> and <q> that is *modified Kaprekar number*

If no modified Kaprekar numbers exist in the given range, print

NO NUMBERS BETWEEN <p> AND <q><newline>

### Sample Input

Please enter the lower limit of the range (p):

**1**

Please enter the upper limit of the range (q):

**100**

### Sample Output

The numbers between 1 and 100 are:

**1**

**9**

**45**

**55**

**99**<newline>

### Explanation

1,9,45,55 and 99 are the modified Kaprekar Numbers in the given range.

### Sample Input

Please enter the lower limit of the range (p):

**2**

Please enter the upper limit of the range (q):

**8**

**Sample Output**

NO NUMBERS BETWEEN 2 AND 8<newline>

**Explanation**

There are no modified Kaprekar Numbers in the given range.

**Sample Input**

Please enter the lower limit of the range (p):

**100**

Please enter the upper limit of the range (q):

**300**

**Sample Output**

The numbers between 100 and 300 are:

**297<newline>**

**Explanation**

The number 297 is the only Kaprekar Numbers between 100 and 300.

If  $n = 297$  then  $n^2 = 88209$  and  $d = 3$ . We split that into two numbers 88 and 209.

We test  $88 + 209 = 297 = 297$ , so this is a modified Kaprekar number.

You must solve this mission using Bitwise **Only!**

# The use of **\***, **+**, **-** and **/** is forbidden!!

Louise and Richard have developed a numbers game. They pick a number and check to see if it is a power of 2. If it is, they divide it by 2. If not, they reduce it by the next lower number which is a power of 2. Whoever reduces the number to 1 wins the game. Louise always starts.

Given an initial value, print all the stages in the game and determine who wins the game.

As an example, let the initial value  $n = 132$ . It's Louise's turn so she first determines that 132 is not a power of 2. The next lower power of 2 is 128, so she subtracts that from 132 and passes 4 to Richard. 4 is a power of 2, so Richard divides it by 2 and passes 2 to Louise. Likewise, 2 is a power so she divides it by 2 and reaches 1. She (Louise) wins the game.

If they initially set counter to 1, Richard wins. Louise cannot make a move so she loses.

## Input Format

The input format will be as follows:

Please enter the initial value for the game:

<n>

Please make sure that the output messages are as described above. (The system is case-sensitive).

- <n> - An integer, the initial value for the game.

No need to check if <n> is valid.

## Constraints

- $1 \leq n \leq 2^{30}$

## Output Format

The output format will be as follows:

Louise: <n<sub>1</sub>>

Richard: <n<sub>2</sub>>

Louise: <n<sub>3</sub>>

...

The winner is: <winner><newline>

- <n<sub>i</sub>> - The  $i^{\text{th}}$  number played by Louise/Richard.
- <winner> - The winner of the Game. Can only be: **Richard** or **Louise**.

### Sample Input

Please enter the initial value for the game:

**6**

### Sample Output

Louise: **6**

Richard: **2**

Louise: **1**

The winner is: **Richard**

### Explanation

- 6 is not a power of 2 so Louise reduces it by the largest power of 2 less than 6:  
 $6 - 4 = 2$ .
- 2 is a power of 2 so Richard divides by 2 to get 1 and wins the game.

### Sample Input

Please enter the initial value for the game:

**132**

### Sample Output

Louise: **132**

Richard: **4**

Louise: **2**

Richard: **1**

The winner is: **Louise**

### Explanation

The initial value  $n = 132$ . It's Louise's turn so she first determines that 132 is not a power of 2. The next lower power of 2 is 128, so she subtracts that from 132 and passes 4 to Richard. 4 is a power of 2, so Richard divides it by 2 and passes 2 to Louise. Likewise, 2 is a power so she divides it by 2 and reaches 1. She wins the game.

Please enter the mission number:

1

Please enter the base of the problem:

8

Please enter the first number in base 8:

715

Please enter the second number in base 8:

501

The result of  $715 + 501$  in base 8 is 1416

Please enter the mission number:

1

Please enter the base of the problem:

12

ERROR - The base 12 is not between 2 and 10

Please enter the mission number:

1

Please enter the base of the problem:

2

Please enter the first number in base 2:

100111

Please enter the second number in base 2:

01010

The result of  $100111 + 01010$  in base 2 is 110001

Please enter the mission number:

2

Please enter the time:

4

The time in the timer after 4 seconds is 6

Please enter the mission number:

2

Please enter the time:

15

The time in the timer after 15 seconds is 7

Please enter the mission number:

2

Please enter the time:

-10

ERROR - The time -10 is invalid

Please enter the mission number:

3

Please enter the lower limit of the range (p):

1

Please enter the upper limit of the range (q):

100

The numbers between 1 and 100 are:

1

9

45

55

99

Please enter the mission number:

3

Please enter the lower limit of the range (p):

2

Please enter the upper limit of the range (q):

8

NO NUMBERS BETWEEN 2 AND 8

Please enter the mission number:

3

Please enter the lower limit of the range (p):

7

Please enter the upper limit of the range (q):

2

ERROR - The range 7 to 2 isn't valid

Please enter the mission number:

3

Please enter the lower limit of the range (p):

100

Please enter the upper limit of the range (q):

300

The numbers between 100 and 300 are:

297

Please enter the mission number:

4

Please enter the initial value for the game:

6

Louise: 6

Richard: 2

Louise: 1

The winner is: Richard

Please enter the mission number:

**4**

Please enter the initial value for the game:

132

Louise: 132

Richard: 4

Louise: 2

Richard: 1

The winner is: Louise

Please enter the mission number:

**5**