Project # 1

Name: ____________________ Registration No: _________________

Marks: 100 Due Date & Time: **Monday January 29, 2024 before 11 : 00 PM**

Instructions:

SUBMIT *.CPP FILE + This project *.PDF document ONLY.

Use any concept, but you would be able to defend VIVA VOICE.

No marks will be awarded if you will do any of the following.

- Compress File Submission (**ZIP/ RAR**)
- Use of **SYSTEM("PAUSE")**
- Submission of **Wrong File/ Empty File/ Commented File**
- **PLAGIARISM** (Zero Tolerance)
- Submission from the **same PC**
- ALL other ways to cheat are strictly prohibited.

15-digit calculator

Note: Design a Menu Based program that can repeat in a loop until we want to continue

Standard integers are only 32 bits (4 bytes) in size which means they can only safely take values that are a maximum of 8 digits long. You must write a C++ program that is capable of handling numbers that can be up to 15 digits long. In order to do this you need to store the number in an array where each digit is placed at individual indices in the array. By doing this you will be able to store larger numbers than what standard limitations allow you to store. You must implement the following mathematical operations in your program:

- Addition
- Subtraction
- Multiplication
- Integer division (i.e. result is always a whole number, you can ignore point values)
- Modulus

You can take number input as separate integers, as integer separation will not be possible because you cannot store the entire number. Note that your result may not always be 15 digits long, so plan accordingly!

Sample case for addition:

Number 1 = 564378642115001

Number 2 = 123456789123456

Num1:

5	6	4	3	7	8	6	4	2	1	1	5	0	0	1

Num2:

1	2	3	4	5	6	7	8	9	1	2	3	4	5	6

Sum:

6	8	7	8	3	5	4	3	1	2	3	8	4	5	7

Sample case for subtraction:

Number 1 = 564378642115001

Number 2 = 123456789123456

Num1:

5	6	4	3	7	8	6	4	2	1	1	5	0	0	1

Num2:

1	2	3	4	5	6	7	8	9	1	2	3	4	5	6

Diff:

4	4	0	9	2	1	8	5	2	9	9	1	5	4	5

Sample case for multiplication:

Number 1 = 564378642115001

Number 2 = 999 (Max Value for Multiply/ divide/ modulus)

Num1:

5	6	4	3	7	8	6	4	2	1	1	5	0	0	1

Num2:

												9	9	9

product:

5	6	3	8	1	4	2	6	3	4	7	2	8	8	5	9	9	9