

Due to your code isn't fully received in the letter, here is right code with similar implementation of class BigInteger.

```
#include <iostream>
#include <cstring>
#include <stdio.h>

using namespace std;

int charIntToInt(char c) {
    return c - '0';
}

class BigInteger {
private:
    char Num1[1000];
    char Num2[1000];
    char Num3[1001];
public:
    void SetNumber(char A[], char B[]);
    int Add();
    void getSumm(char* summ);
    BigInteger();
};

BigInteger::BigInteger() {
    for (int i = 0; i < 1001; i++) {
        Num3[i] = 0;
    }
}

void BigInteger :: SetNumber(char A[], char B[]) {
    //everyint length(A) >= length(B)
    //to add 0 to B number in th beginning
    for (unsigned int i = 0; i < strlen(A); i++) {
        Num1[i] = A[i];
    }
    for (unsigned int i = 0; i < strlen(B); i++) {
        Num2[i] = B[i];
    }

    //swap Num1 and Num2 if length(Num1) < length(Num2)
    if (strlen(Num1) < strlen(Num2)) {
        char temp_number[1000];
        strcpy(temp_number, Num1);
        strcpy(Num1, Num2);
        strcpy(Num2, temp_number);
    }

    //write zero to the beginning of Num2
    char new_Num2[1000];
    int i = 0;
    for (; i < strlen(Num1) - strlen(Num2); i++) {
        new_Num2[i] = '0';
    }
}
```

```

    for (int j = 0; j < strlen(Num2); j++, i++) {
        new_Num2[j] = Num2[j];
    }
    strcpy(Num2, new_Num2);

//    printf ("%s\n", Num1);
//    printf ("%s\n", Num2); --uncomment to see debug output
}

int BigInteger :: Add() {
    int decimal_memory = 0;
    for (int i = strlen(Num1) - 1; i >= 0; i--) {
        int x = (charToInt(Num1[i]) + charToInt(Num2[i]) +
                decimal_memory) % 10;
        char c = x + '0';
        Num3[i] = c;
        decimal_memory = (charToInt(Num1[i]) + charToInt(Num2[i]) +
                decimal_memory) / 10;
    }

    //exception
    //in case like 999+1; 5+5; and etc
    if (decimal_memory == 1) {
        char new_Num3 [1000];
        new_Num3[0] = '1';
        for (int i = 0; i < strlen(Num3); i++) {
            new_Num3 [i + 1] = Num3 [i];
        }
        strcpy(Num3, new_Num3);
    }
}

//it's not ok to return pointer!!!
//ok is to store some value into function param //such as address of summ void BigInteger ::
getSumm(char* summ) {
    strcpy(summ, Num3);
}

int main() {
    //it is some example
    //and next 2 strings cause warnings, but it's ok
    /*
    char* A = "999999999999999";
    char* B = "1";
    BigInteger bi;
    bi.SetNumber(A,B);
    bi.Add();
    debug
    char res [1001];
    bi.getSumm(res);
    printf ("%s\n", res);
    */
    return 0;
}

```

<= uncomment these lines for