**Problem 01 (minmaxsum.c):** Find the sum of the minimum and the maximum of n numbers.

e.g.

### Sample Input:

Enter n: **5**

Enter number 1: **10**

Enter number 2: **-7**

Enter number 3: **9**

Enter number 4: **37**

Enter number 5: **24 Sample Output:**

#### Sum = 30

**Problem 02 (house.c):** Take the wall height ***h*** of the house as input and print the house. The width ***w*** of the house is twice its height ***h*** and the roof (triangle) height ***rh*** is 1 less than the height ***h***. You must ensure user enters a height not less than 2 (i.e. do input validation). (note: there’s space b/w 2 stars in a row)

e.g.

### Sample Input:

Enter n: **5 Sample Output:**

### \* \*

**\* \* \* \* rh = 4 stars (roof)**

**\* \* \* \* \* \***

**\* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* h = 5 stars (wall)**

**\* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \***

**w = 10 stars (width)**

e.g.

### Sample Input:

Enter n: **8 Sample Output:**

### \* \*

**\* \* \* \***

**\* \* \* \* \* \***

**\* \* \* \* \* \* \* \* rh = 7 stars (roof)**

**\* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* h = 8 stars (wall)**

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**w = 16 stars (width)**

**Problem 03 (digitsum.c):** Find the sum of digits of an input number **n**. **n** can range from 0 to +2 billion and so we don’t know how many digits the number has. You may do this using a for loop as well since 2 billion means 2000000000 i.e. in the worst case there may be 10 digits. Find the sub-problem for this larger problem; it might be as simple as finding the least significant digit may be! Do validation for input i.e. n must be **>=0**. You are also allowed to use while loop.

e.g.

### Sample Input:

Enter n: **1234 Sample Output:**

#### Sum of digits = 10

**Sample Input:**

Enter n: **27834289 Sample Output:**

#### Sum of digits = 43

**Sample Input:**

Enter n: **3 Sample Output:**

#### Sum of digits = 3

**Problem 04 (characters.c):** Input n characters and then in the end print the summary of the inputs i.e. count of lower-case alphabets, upper-case alphabets, vowels, digits, special characters (all characters excluding alphabets and digits). n must be **> 0** – validation!

### Sample Input:

Enter n: **5**

Enter character 1: **a** Enter character 2: **T** Enter character 3: **3**

Enter character 4: **9** Enter character 5: **P Sample Output:**

Upper-case alphabets = 2 Lower-case alphabets = 1 Vowels = 1

Digits = 2

Special = 0

### Sample Input:

Enter n: **5**

Enter character 1: **#** Enter character 2: **1** Enter character 3: **d** Enter character 4: **x** Enter character 5: **+ Sample Output:**

Upper-case alphabets = 0 Lower-case alphabets = 2 Vowels = 0

Digits = 1

Special = 3

### Sample Input:

Enter n: **3**

Enter character 1: **A** Enter character 2: **u** Enter character 3: **4 Sample Output:**

Upper-case alphabets = 1 Lower-case alphabets = 1 Vowels = 2

Digits = 1

Special = 0

**Problem 05 (divisors.c):** Write a program in which the user inputs two numbers n and x. n is the number of input values that the user will be prompted next. Find out how many of these n numbers are divisible by x.

### Sample Input:

Enter n: **5**

Enter x: **3**

Enter number 1: **10**

Enter number 2: **21**

Enter number 3: **6**

Enter number 4: **16**

Enter number 5: **30**

### Sample Output:

**Count = 3**

**Sample Input:**

Enter n: **3**

Enter x: **10**

Enter number 1: **35**

Enter number 2: **42**

Enter number 3: **6**

### Sample Output:

**Count = 0**

#### String Problems

# Problem #1: Cases (cases.c):

Write a program that inputs a sentence from the user and converts it into four different cases i.e.

**UPPER CASE** i.e. change all alphabets to upper case,

**lower case** i.e. change all alphabets to lower case,

**Proper Case** i.e. change the first alphabet of every word to upper case and the remaining alphabets of that word to lower, and

**Sentence case** i.e. only the first alphabet of the sentence is capitalized.

e.g.:

Input: I love PROGRAMMING. also, mathematics. Output:

Upper Case: I LOVE PROGRAMMING. ALSO, MATHEMATICS.

Lower Case: i love programming. also, mathematics. Proper Case: I Love Programming. Also, Mathematics. Sentence Case: I love programming. Also, mathematics.

You may verify your sample inputs/outputs from here: <http://caseconverter.com/>

The function prototypes shall be:

**void toUpperCase(char str[]); void toLowerCase(char str[]); void toProperCase(char str[]); void toSentenceCase(char str[]);**

**Note: you are not allowed to use any library other than stdio.h and stdlib.h. hence no string.h!**

# Problem #2: Check parentheses (parenthesis.c):

Write a program that finds whether a given sequence of parenthesis is balanced or not. For example:

Input: ((()))()()(())

Output: Balanced Input: ()()()()()()()

Output: Balanced Input: ((())()()(())

Output: Not Balanced Input: ()()()()()()(

Output: Not Balanced Input: )(

Output: Not Balanced

The function prototype shall be:

**int isBalanced(char str[]);**

***Bonus: your program should also check strings having characters other than the parentheses:***

***e.g. Input: if((a>b) && (b>c)) Output: balanced***

*Note: it’s not as simple as counting the # of opening and closing parenthesis since*

*)( is considered as not balanced!*

# Problem #3: String Statistics (stringstats.c):

Write a program to find out the number of vowels(a,e,i,o,u), consonants(that are not vowels in english alphabets), digits(0-9) and special characters(!,-,\*, , ) in an input string. For

example:

Input: Hello World 123!!! Output:

# of vowels: 3

# of consonants: 7 # of digits: 3

# of special characters: 5

The function prototype shall be:

**void stringStats(char str[], int\* vowels, int\* consonants, int\* digits, int\* special); Note: all parameters above are being passed by reference!**