Numbers

# **Count of Digits**

**int** countOfDigits(**int** number){  
 **int** counter=0;  
 **while**(number){  
 counter++;  
 number/=10;  
 }  
 **return** counter;  
}

Count of digits 2

**int** countOfDigits(**int** number){  
 **int** counter=1;  
 **while**(number){  
 counter\*=10;  
 number/=10;  
 }  
 counter/=10;  
 **return** counter;  
}

Reverse number

**int** reverseNumber(**int** number){  
 **int** reversNumber=0;  
 **while**(number){  
 **int** lastDigit=number%10;  
 reversNumber= reversNumber\*10+lastDigit;  
 }  
 **return** reversNumber;  
}

Number of divisors

**int** numberOfDivisors(**int** number){  
 **int** numberOfDivisors=0;  
 **for**(**int** i=1 ; i<=number ; i++){  
 **if**(number%i==0){  
 numberOfDivisors++;  
 }  
 }  
 **return** numberOfDivisors;  
}

Sum of divisors

**int** sumOfDivisors(**int** number){  
 **int** sumOfDivisors=0;  
 **for**(**int** i=1 ; i<=number ; i++){  
 **if**(number%i==0){  
 sumOfDivisors+=i;  
 }  
 }  
 **return** sumOfDivisors;  
}

Strings

Shifting a string by X places

**void** transform(**char** \*str,**int** X){  
 **for**(**int** i=0 ; i< strlen(str) ; i++){  
 **if**(islower(str[i])){  
 **if**(str[i]+X>**'z'**){  
 **int** s=**'z'**-str[i];  
 **int** shift=X-s;  
 str[i]=**'a'**-1+shift;  
 } **else**{  
 str[i]=str[i]+X;  
 }  
 } **else if**(isupper(str[i])){  
 **if**(str[i]+X>**'Z'**){  
 **int** s=**'Z'**-str[i];  
 **int** shift=X-s;  
 str[i]=**'A'**-1+shift;  
 } **else**{  
 str[i]=str[i]+X;  
 }  
 }  
 }  
}

Clean a string

**void** clearStr(**char** \*str){  
 **if**(str[strlen(str)-1]==**'\n'**){  
 str[strlen(str)-1]=**'\0'**;  
 }  
}

Check if a given character is vowel

**int** isVowel(**char** c){  
 **return** tolower(c)==**'a'** || tolower(c)==**'e'** || tolower(c)==**'i'** || tolower(c)==**'o'** || tolower(c)==**'u'**;  
}

The ratio digits/letters

**float** ratio(**char** \*str){  
 **int** countL=0, countD=0;  
 **for**(**int** i=0 ; i< strlen(str) ; i++){  
 **if**(isdigit(str[i])){  
 countD++;  
 }  
 **if**(isalpha(str[i])){  
 countL++;  
 }  
 }  
 **return** (**float**)countD/(**float**)countL;  
}

Sum of all numbers in a string

**int** sumOfNumbers(**char** \*str){  
 **int** sum=0;  
 **int** currentNumber=0;  
 **for**(**int** i=0 ; i< strlen(str) ; i++){  
 **if**(isdigit(str[i])){  
 currentNumber=10\*currentNumber+(str[i]-**'0'**);  
 } **else**{  
 sum+=currentNumber;  
 currentNumber=0;  
 }  
 }  
 **return** sum;  
}

Check if a word/string is a palindrome

**int** isPalindrome(**char** \*word){  
 **for**(**int** i=0 ; i< strlen(word) ; i++){  
 **if**(tolower(word[i])!= tolower(word[strlen(word)-1-i])){  
 **return** 0;  
 }  
 }  
 **return** 1;  
}

Ratio uppercase/lowercase letters in a word/string

**float** ratioUpperLower(**char** \*str){  
 **float** countUpper=0, countLower=0;  
 **for**(**int** i=0 ; i< strlen(str) ; i++){  
 **if**(isalpha(str[i])){  
 **if**(isupper(str[i])){  
 countUpper++;  
 } **else**{  
 countLower++;  
 }  
 }  
 }  
 **return** countUpper/countLower;  
}

Recursive

Triangle

**void** triangle (**int** n,**int** m){  
 **if**(n==0){  
 **return**;  
 } **else**{  
 printf(**"%d"**,m-n+1);  
 **return** triangle(n-1,m);  
 }  
}  
  
**void** triangle2 (**int** n,**int** m){  
 **if**(n==0){  
 **return**;  
 } **else**{  
 triangle(m-n+1,m-n+1);  
 printf(**"\n"**);  
 **return** triangle2(n-1,m);  
 }  
}