8a. Strings a. Printing Tokens Objective: print each word of the sentence in a new line https://www.hackerrank.com/challenges/printing-tokens- /problem?isFullScreen=true

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {

   char s[100];
   int i;
   gets(s);
   for(i=0;i<strlen(s);i++)
   {
      if(s[i]==' ')
         printf("\n");
      else
         printf("%c",s[i]);
   }
   return 0;
}
```

OUTPUT:

```
S (stdin)
This is C

Your Output (stdout)
This
is
C

Expected Output
This
is
C{-truncated-}
```

8b. Count number of alphabets (lowercase, uppercase, consonants, vowels) and digits

Objective:

```
#include <stdio.h>
#include <ctype.h>

int main()
{
   char s[100];
   int lowercase = 0, uppercase = 0, vowels = 0, consonants = 0, digits = 0;

   printf("Enter a string: ");
   gets(s);
   for (int i = 0; s[i] != '\0'; i++)
   {
      if (isalpha(s[i]))
         {
            if (islower(s[i]))
               {
                  lowercase++;

                    if (s[i] == 'a' || s[i] == 'e' || s[i] == 'i' || s[i] == 'o' || s[i] == 'u' ||
                      s[i] ==  'A' || s[i] == 'E' || s[i] == 'I' || s[i] == 'O' || s[i] == 'U')
                         {
                              vowels++;
                         }
                     else
                          {
                              consonants++;
                          }
                    }
               else if (isupper(s[i]))
                    {
                        uppercase++;
                   if (s[i] == 'A' || s[i] == 'E' || s[i] == 'I' || s[i] == 'O' || s[i] == 'U')
                        {
                            vowels++;
                         }
                   else
                          {
                            consonants++;
                          }
                     }
```

```
        }
            else if (isdigit(s[i]))
              {
                  digits++;
              }
    }

    printf("Lowercase letters: %d\n", lowercase);
    printf("Uppercase letters: %d\n", uppercase);
    printf("Vowels: %d\n", vowels);
    printf("Consonants: %d\n", consonants);
    printf("Digits: %d\n", digits);

    return 0;
}
```

Output:

```
Enter a string: Hello, World! 123
Lowercase letters: 8
Uppercase letters: 2
Vowels: 3
Consonants: 7
Digits: 3
```

8c.Lowercase to Uppercase, Uppercase to Lowercase, Toggle case, Sentential case

Objective:

```
#include <stdio.h>
#include <string.h>

void toUpperCase(char *str)
{
    for (int i = 0; str[i]; i++)
    {
        if (str[i] >= 'a' && str[i] <= 'z')
        {
            str[i] = str[i] - 32; // Convert to uppercase
        }
    }
}

void toLowerCase(char *str)
{
```

```
    for (int i = 0; str[i]; i++)
    {
        if (str[i] >= 'A' && str[i] <= 'Z')
        {
            str[i] = str[i] + 32; // Convert to lowercase
        }
    }
}

void toggleCase(char *str)
{
    for (int i = 0; str[i]; i++)
    {
        if (str[i] >= 'a' && str[i] <= 'z')
        {
            str[i] = str[i] - 32; // Convert to uppercase
        } else if (str[i] >= 'A' && str[i] <= 'Z')
        {
            str[i] = str[i] + 32; // Convert to lowercase
        }
    }
}

void sententialCase(char *str)
{
    int capitalize = 1; // Start with a capital letter
    for (int i = 0; str[i]; i++)
    {
        if (str[i] >= 'a' && str[i] <= 'z' && capitalize)
        {
            str[i] = str[i] - 32; // Convert to uppercase
            capitalize = 0;
        } else if (str[i] >= 'A' && str[i] <= 'Z' && !capitalize)
        {
            str[i] = str[i] + 32; // Convert to lowercase
            capitalize = 1;
        }
    }
}

int main()
{
    char input[100];
```

```
    printf("Enter a string: ");
    fgets(input, sizeof(input), stdin);
    input[strcspn(input, "\n")] = '\0'; // Remove the newline character from input

    char output[100];
    strcpy(output, input); // Copy the input to output for each case conversion

    toUpperCase(output);
    printf("Uppercase: %s\n", output);

    strcpy(output, input);
    toLowerCase(output);
    printf("Lowercase: %s\n", output);

    strcpy(output, input);
    toggleCase(output);
    printf("Toggle Case: %s\n", output);

    strcpy(output, input);
    sententialCase(output);
    printf("Sentential Case: %s\n", output);

    return 0;
}
```

Output:

```
Enter a string: This is a Sample String
Uppercase: THIS IS A SAMPLE STRING
Lowercase: this is a sample string
Toggle Case: tHIS IS A sAMPLE sTRING
Sentential Case: This is A sample String
```

8d. Digit Frequency

Objective: find the frequency of each digit in the given string.

https://www.hackerrank.com/challenges/frequency-of-digits1/problem?isFullScreen=true

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
```

```c
int main() {

    /* Enter your code here. Read input from STDIN. Print output to STDOUT */

    char s[1000];
    int freq[10] = {0};

    scanf("%[^\n]", s);
    for (int i = 0; i < strlen(s); i++) {
        if (s[i] >= '0' && s[i] <= '9') {
            freq[s[i] - '0']++;
        }
    }

    for (int i = 0; i < 10; i++) {
        printf("%d ", freq[i]);
    }
    return 0;
}
```

OUTPUT:

```
Input (stdin)
a11472o5t6

Your Output (stdout)
0 2 1 0 1 1 1 1 0 0

Expected Output
0 2 1 0 1 1 1 1 0 0
```

8e. Find string length, concatenate 2 strings, reverse a string using built-in and without built-in string functions.

```c
#include <stdio.h>
#include <string.h>

// Function to find the length of a string without using built-in function
```

```
int customStringLength(const char *str)
{
   int length = 0;
   while (str[length] != '\0')
   {
      length++;
   }
   return length;
}

// Function to concatenate two strings without using built-in function
void customStringConcat(char *destination, const char *source)
 {
   int destLength = customStringLength(destination);
   int sourceLength = customStringLength(source);

   for (int i = 0; i < sourceLength; i++)
   {
      destination[destLength + i] = source[i];
   }
   destination[destLength + sourceLength] = '\0';
}


// Function to reverse a string without using built-in function
void customStringReverse(char *str)
 {
   int length = customStringLength(str);
   for (int i = 0; i < length / 2; i++)
   {
      char temp = str[i];
      str[i] = str[length - i - 1];
      str[length - i - 1] = temp;
   }
}

int main()
{
   char str1[100], str2[100], concatenated[200];

   printf("Enter a string: ");
   scanf("%s", str1);
```

```
    printf("Enter another string: ");
    scanf("%s", str2);

    // Finding the length of str1
    int length1 = strlen(str1);
    printf("Using built-in function: Length of str1 = %d\n", length1);

    // Concatenating str1 and str2 using built-in function
    strcpy(concatenated, str1);
    strcat(concatenated, str2);
    printf("Using built-in function: Concatenated string = %s\n", concatenated);

    // Reversing str1 using built-in function
    strcpy(concatenated, str1);
    strrev(concatenated); // Note: strrev is not a standard C function, but some
compilers provide it.
    printf("Using built-in function: Reversed str1 = %s\n", concatenated);

    // Finding the length of str1 without using built-in function
    length1 = customStringLength(str1);
    printf("Without built-in function: Length of str1 = %d\n", length1);

    // Concatenating str1 and str2 without using built-in function
    strcpy(concatenated, str1);
    customStringConcat(concatenated, str2);
    printf("Without built-in function: Concatenated string = %s\n",concatenated);

    // Reversing str1 without using built-in function
    strcpy(concatenated, str1);
    customStringReverse(concatenated);
    printf("Without built-in function: Reversed str1 = %s\n", concatenated);
    return 0;
}
```

Output:

```
Enter a string: Hello

Enter another string: World

Using built-in function: Length of str1 = 5

Using built-in function: Concatenated string = HelloWorld

Using built-in function: Reversed str1 = olleH
```

```
Without built-in function: Length of str1 = 5
Without built-in function: Concatenated string = HelloWorld
Without built-in function: Reversed str1 = olleH
```