Name: Santosh [CB.EN.U4CCE20053]

Department: Computer and Communication Engineering [CCE]

Subject: 19CSE102 Computer Programming Assignment on Strings

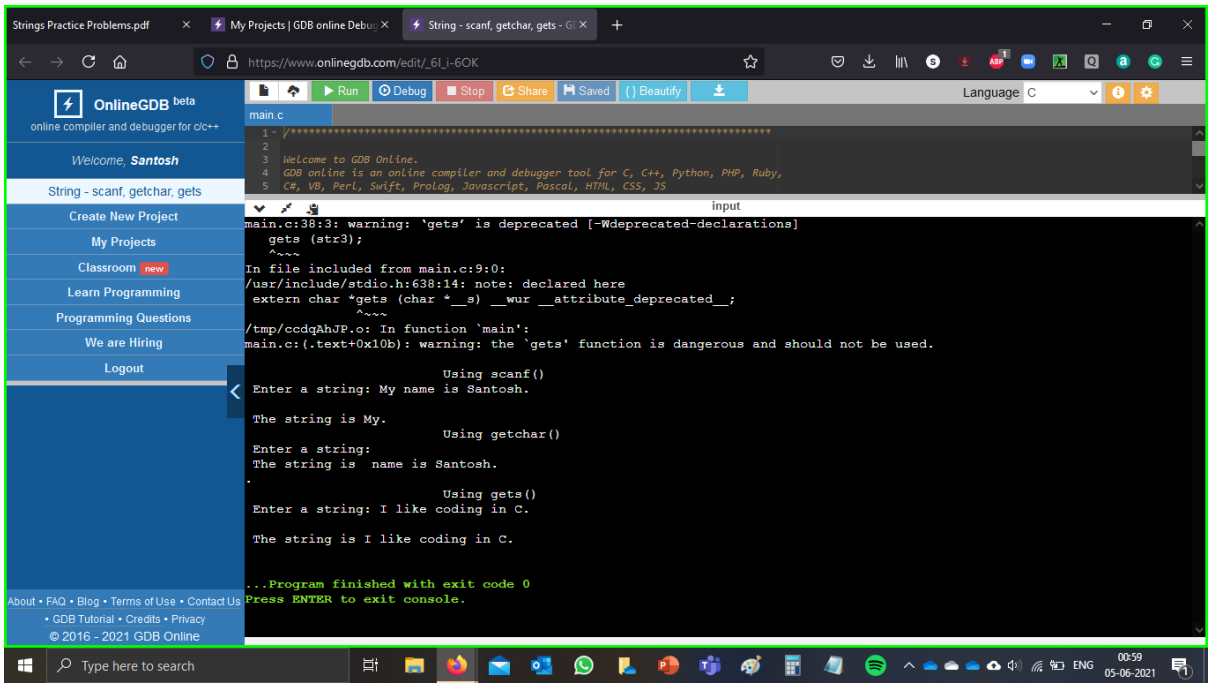
Date: 05/06/2021 – Saturday

Q1. To input a string and print it, using scanf(), getchar() and gets().

A1.

```
#include <stdio.h>

int
main ()
{
  char str1[50], str2[50], str3[50], ch;
  int i = 0;

  //Input and print string using scanf()
  printf ("\n\t\t\t Using scanf()");
  printf ("\n Enter a string: ");
  scanf ("%s", str1);
  printf ("\n The string is %s.", str1);

  //Input and print string using getchar()
  printf ("\n\t\t\t Using getchar()");
  printf ("\n Enter a string: ");
  while (ch != '\n')
    {
      ch = getchar ();
      str2[i] = ch;
      i++;
    }
  str2[i] = '\0';
  printf ("\n The string is %s.", str2);
```

```
 //Input and print string using gets()
 printf ("\n\t\t\t Using gets()");
 printf ("\n Enter a string: ");
 gets (str3);
 printf ("\n The string is ");
 puts (str3);

 return 0;
}
```

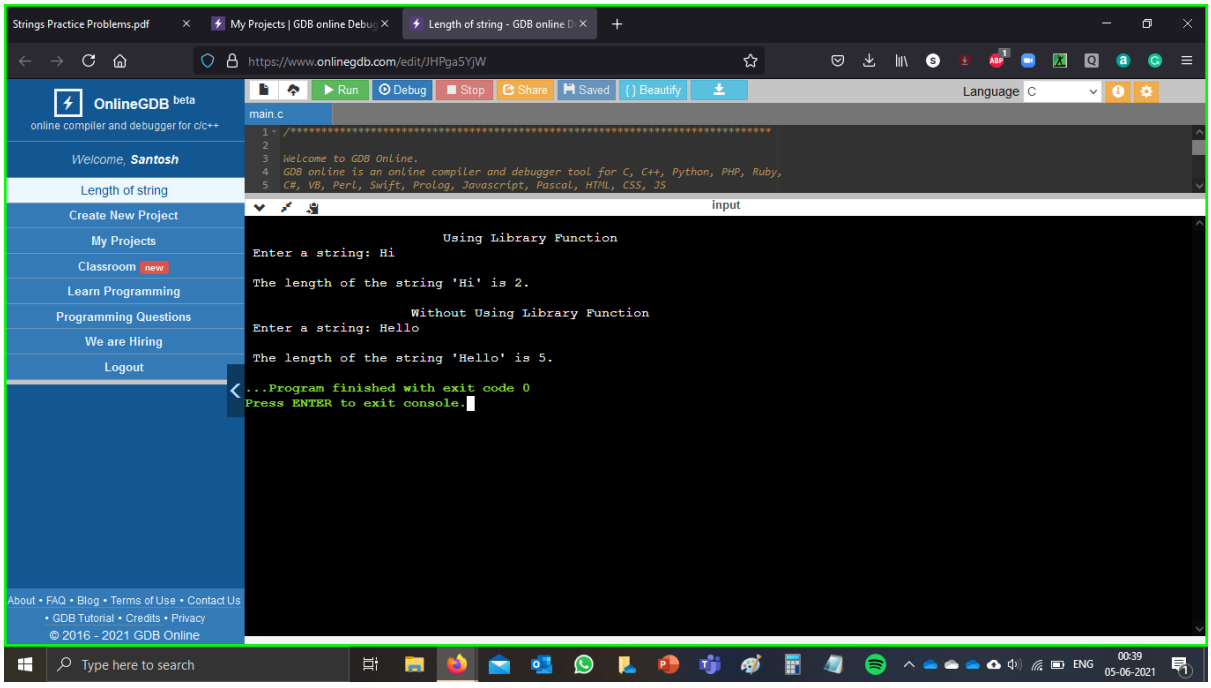


Q2. To find the length of a string with and without using library function.

A2.

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

int
main ()
{
  char str1[50], str2[50];
  int length1 = 0, length2 = 0, i;
```

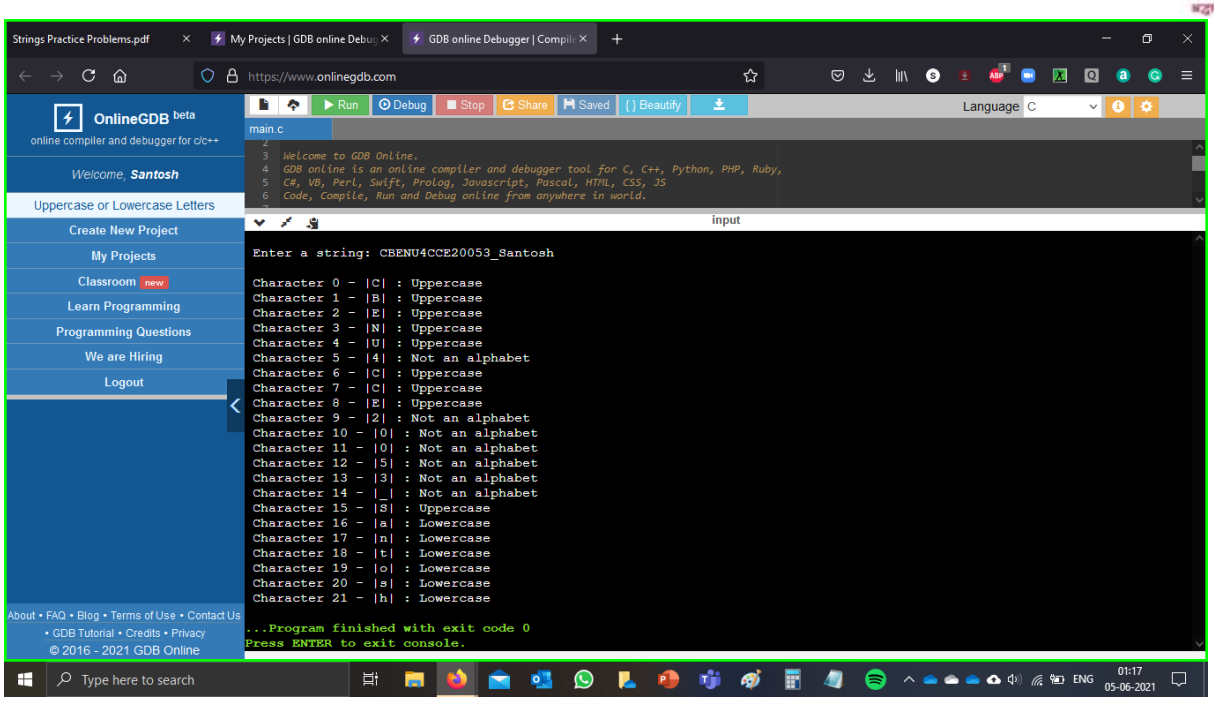
```
//Length using library function
printf ("\n\t\t\t Using Library Function");
printf ("\n Enter a string: ");
scanf ("%s", str1);
length1 = strlen (str1);
printf ("\n The length of the string '%s' is %d.", str1, length1);
printf ("\n");

//Length without using library function
printf ("\n\t\t\t     Without Using Library Function");
printf ("\n Enter a string: ");
scanf ("%s", str2);
for (i = 0; str2[i] != '\0'; i++)
  length2++;
printf ("\n The length of the string '%s' is %d.", str2, length2);

return 0;
}
```



Q3. To separate the individual characters from a string and print whether it is lower-case or upper-case letters.

A3.

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

int
main ()
{
  char str[50];
  int i;
  printf ("\n Enter a string: ");
  scanf ("%s", str);
  for (i = 0; str[i] != '\0'; i++)
    {
      if (islower (str[i]))
            printf ("\n Character %d - |%c| : Lowercase", i, str[i]);
      else if (isupper (str[i]))
            printf ("\n Character %d - |%c| : Uppercase", i, str[i]);
      else
            printf ("\n Character %d - |%c| : Not an alphabet", i, str[i]);
    }
  return 0;
}
```

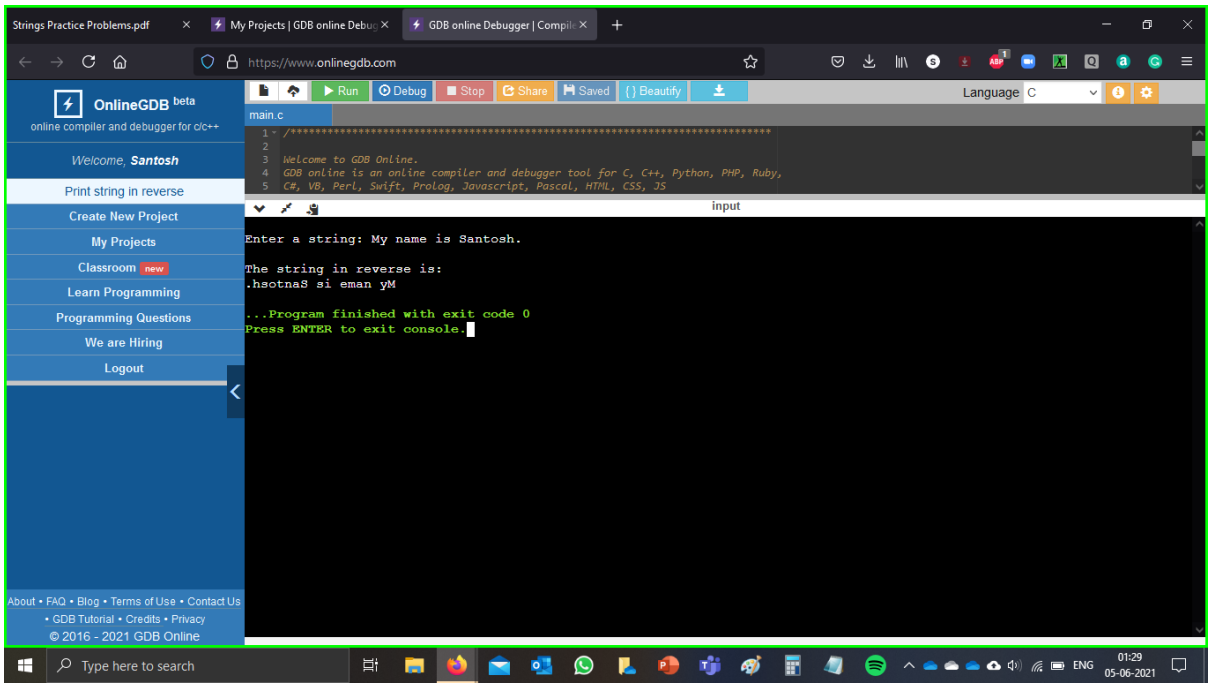


Q4. To count the number of characters in a string including space and print it in reverse order.

A4.

```
#include <stdio.h>

#include <string.h>

#include <stdlib.h>


void

main ()

{

  char str[50];

  int length, i;

  printf ("\nEnter a string: ");

  fgets (str, sizeof str, stdin);

  length = strlen (str);

  printf ("\nThe string in reverse is:");

  for (i = length; i >= 0; i--)

    {

      printf ("%c", str[i]);

    }
```

}

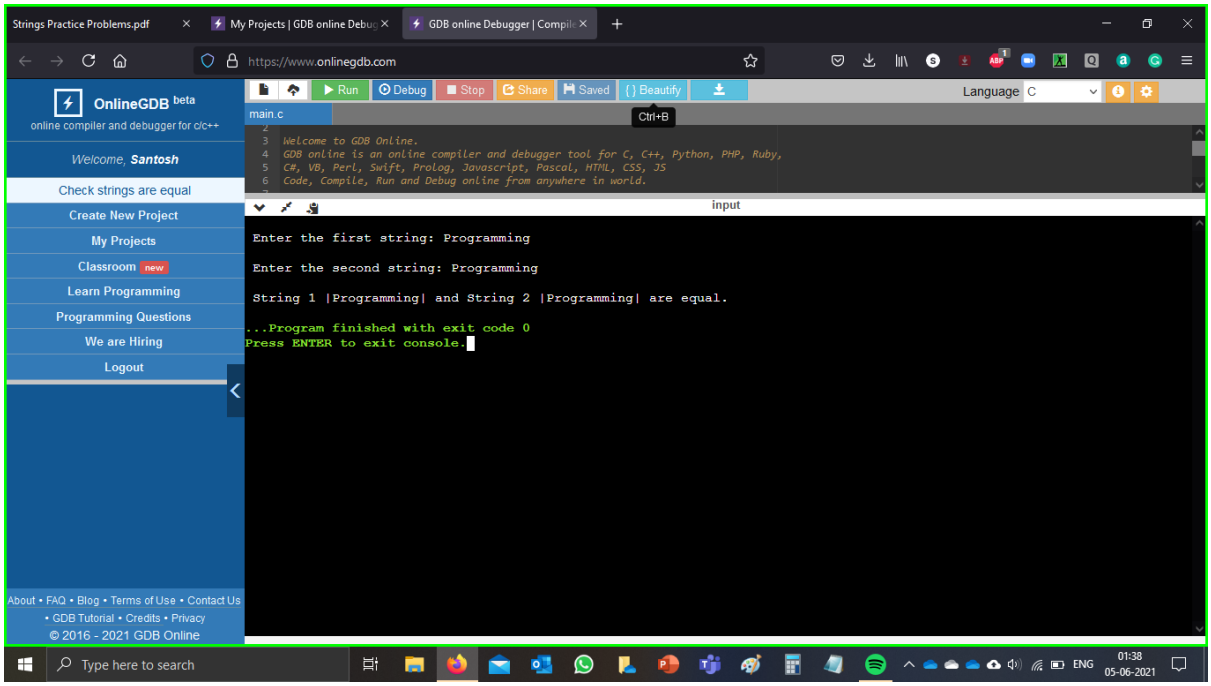


Q5. To check whether two strings are equal or not, using library function.

A5.

```
#include <stdio.h>
#include<string.h>
int
main ()
{
  char str1[25];
  char str2[25];
  int compare;
  printf ("\n Enter the first string: ");
  scanf ("%s", str1);
  printf ("\n Enter the second string: ");
  scanf ("%s", str2);

  //Comparing both the strings using strcmp() function
  compare = strcmp (str1, str2);
  if (compare == 0)
```

```
  printf ("\n String 1 |%s| and String 2 |%s| are equal.", str1, str2);
 else
  printf ("\n String 1 |%s| and String 2 |%s| are not equal.", str1, str2);
 return 0;
}
```



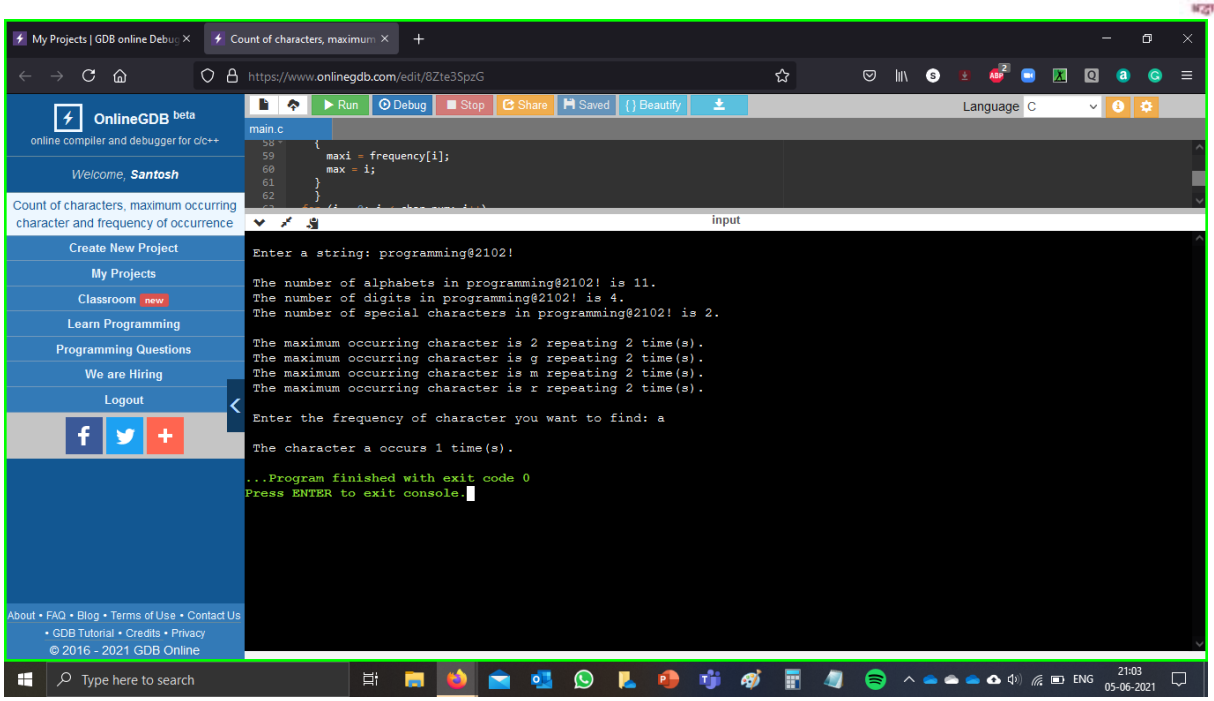
Q6. To count the total number of alphabets, digits and special characters in a string and also the maximum occurring character of the string. Also find the frequency of occurrence of any character of user's choice.

A6.

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

void maximum (char s[50]);
void frequency (char s[50]);

int
main ()
{
  char str[50];
```

```
 int i, count_alpha = 0, count_digit = 0, count_sc = 0;
 printf ("\n Enter a string: ");
 scanf ("%s", str);
 for (i = 0; str[i] != '\0'; i++)
  {
   if (isalpha (str[i]))
        count_alpha++;
   else if (isdigit (str[i]))
        count_digit++;
   else
        count_sc++;
  }
 printf ("\n The number of alphabets in %s is %d.", str, count_alpha);
 printf ("\n The number of digits in %s is %d.", str, count_digit);
 printf ("\n The number of special characters in %s is %d.", str, count_sc);
 printf ("\n");
 maximum (str);
 printf ("\n");
 frequency (str);
 return 0;
}

void
maximum (char s[50])
{
 int i, char_num = 255, frequency[char_num], ascii, max, maxi;
 for (i = 0; i < char_num; i++)
  frequency[i] = 0;
 i = 0;
 while (s[i] != '\0')
  {
   ascii = (int) s[i];
   frequency[ascii] += 1;
   i++;
  }
 max = 0;
```

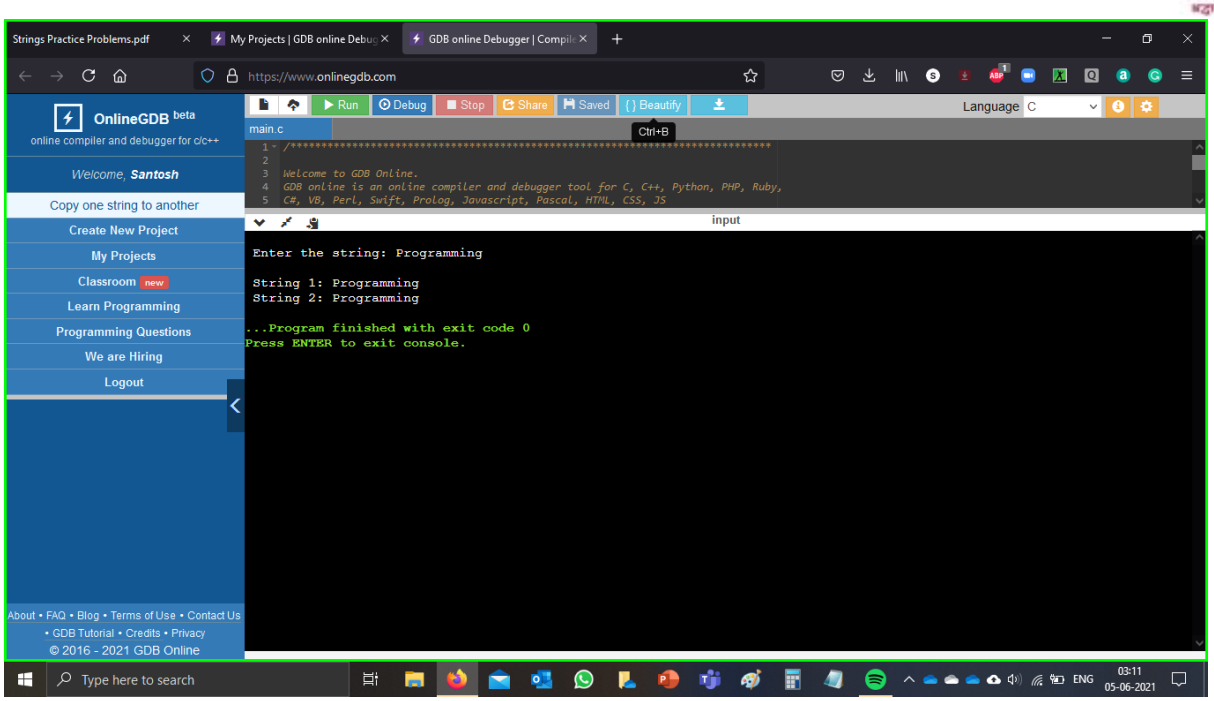
```
  for (i = 0; i < char_num; i++)
    {
      if (frequency[i] > frequency[max])
            {
              maxi = frequency[i];
              max = i;
            }
    }
  for (i = 0; i < char_num; i++)
    {
      if (frequency[i] == maxi)
            printf
              ("\n The maximum occurring character is %c repeating %d time(s).",
               i, maxi);
    }
}

void
frequency (char s[50])
{
  char alpha;
  int i, freq = 0;
  printf ("\n Enter the frequency of character you want to find: ");
  scanf (" %c", &alpha);
  for (i = 0; s[i] != '\0'; i++)
    {
      if (s[i] == alpha)
            freq++;
    }
  printf ("\n The character %c occurs %d time(s).", alpha, freq);
}
```



Q7. To copy one string to another.

A7.

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

int
main ()
{
  char str1[50], str2[50];
  printf ("\n Enter the string: ");
  scanf ("%s", str1);
  strcpy (str2, str1);
  printf ("\n String 1: %s\n String 2: %s", str1, str2);
  return 0;
}
```

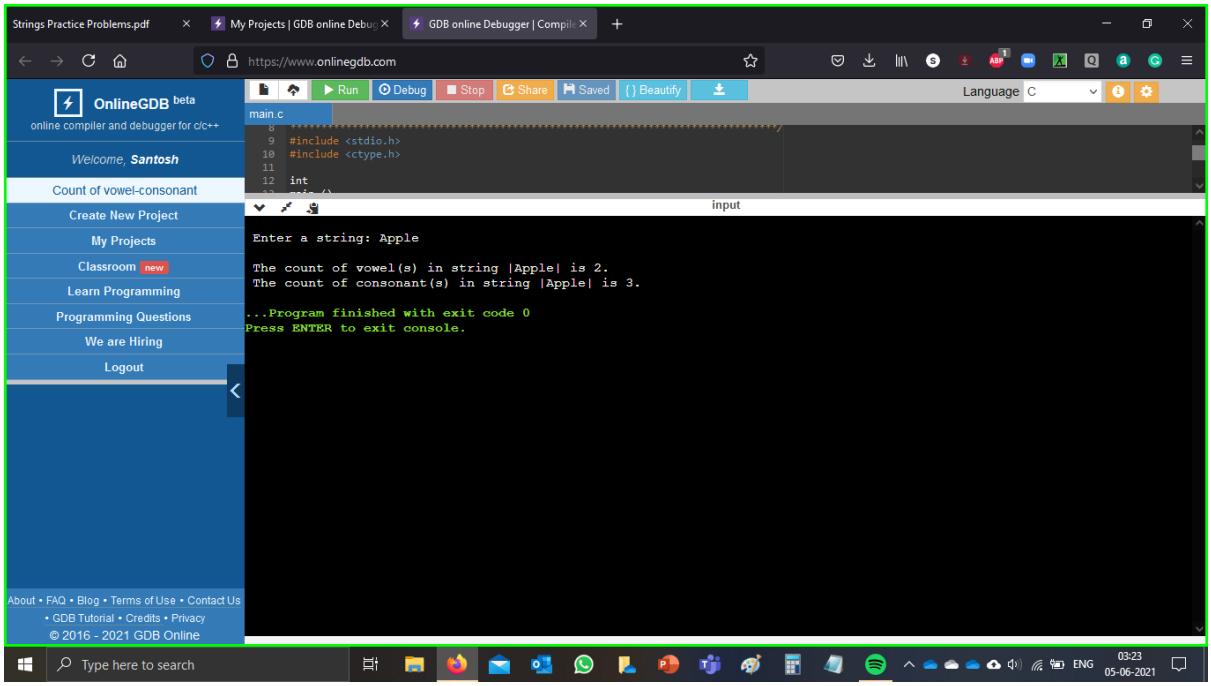


Q8. To count the total number of vowels or consonants in a string.

A8.

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

int
main ()
{
  char str[50];
  int count_vowel = 0, count_consonant = 0, i;
  printf ("\n Enter a string: ");
  scanf ("%s", str);
  for (i = 0; str[i] != '\0'; i++)
    {
      if (isalpha (str[i]))
            {
              if (str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O'
                  || str[i] == 'U' || str[i] == 'a' || str[i] == 'e'
                  || str[i] == 'i' || str[i] == 'o' || str[i] == 'u')
```

```
            count_vowel++;
          else
            count_consonant++;
        }
    }
  printf ("\n The count of vowel(s) in string |%s| is %d.", str, count_vowel);
  printf ("\n The count of consonant(s) in string |%s| is %d.", str,
            count_consonant);
  return 0;
}
```



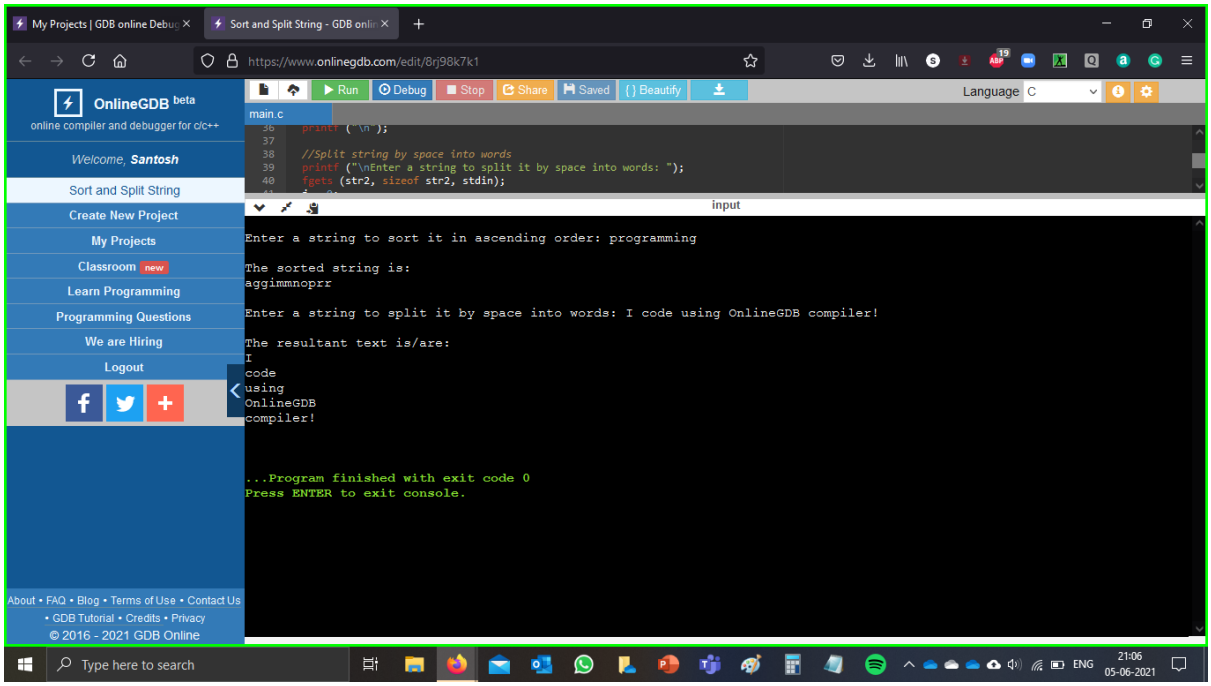
Q9. To sort a string in ascending order and to split string by space into words.

A9.

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

int
main ()
{
```

```
char str1[50], ch, str2[50], result[50][50];
int length1, i, j, counter, length2;

//Bubble Sort String
printf ("\nEnter a string to sort it in ascending order: ");
fgets (str1, sizeof str1, stdin);
length1 = strlen (str1);
for (i = 0; i <= (length1 - 2); i++)
  {
    for (j = 0; j <= (length1 - i - 2); j++)
          {
            if (str1[j] > str1[j + 1])
              {
                ch = str1[j];
                str1[j] = str1[j + 1];
                str1[j + 1] = ch;
              }
          }
  }
printf ("\nThe sorted string is: %s", str1);
printf ("\n");

//Split string by space into words
printf ("\nEnter a string to split it by space into words: ");
fgets (str2, sizeof str2, stdin);
j = 0;
counter = 0;
length2 = strlen (str2);
for (i = 0; i < length2; i++)
  {
    // If space or NULL found, assign NULL into result[counter]
    if (str2[i] == ' ' || str2[i] == '\0')
          {
            result[counter][j] = '\0';
            counter++;                    //for next word
            j = 0;                //for next word, init index to 0
```

```
            }
    else
            {
             result[counter][j] = str2[i];
             j++;
            }
  }
 printf ("\nThe resultant text is/are: \n");
 for (i = 0; i < counter + 1; i++)
   printf ("%s\n", result[i]);
 return 0;
}
```

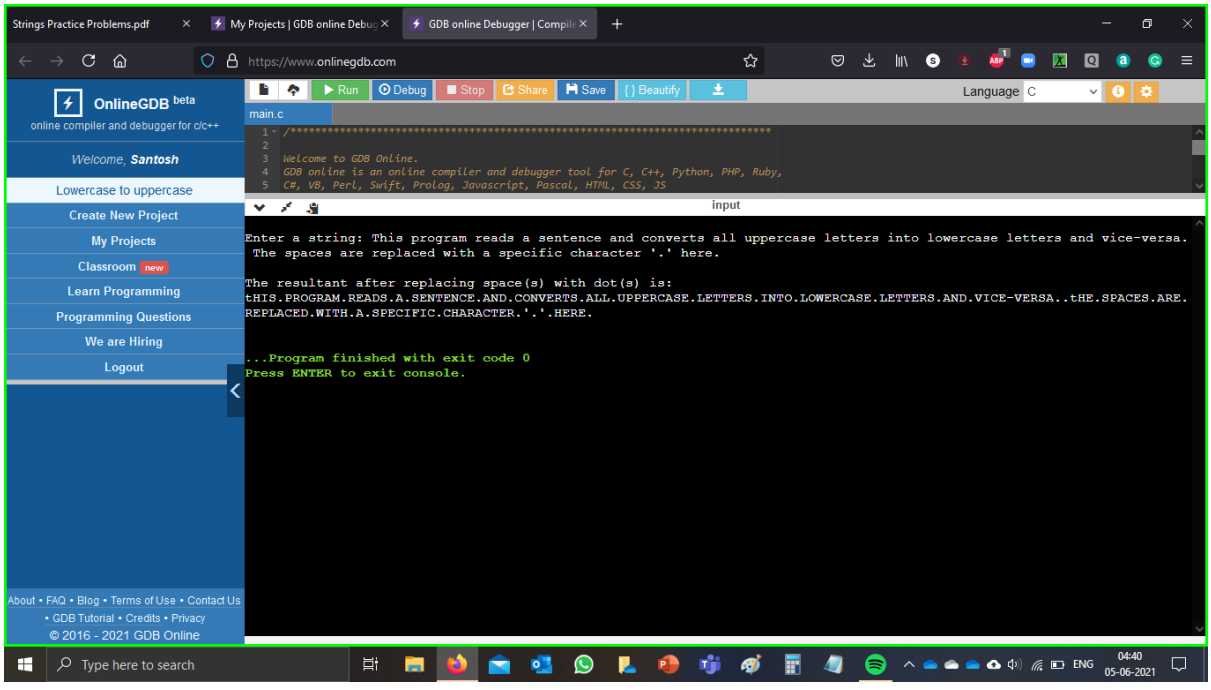


Q10. To read a sentence and replace lowercase letters with uppercase letter and vice versa. Also replace the spaces of a string with a specific character.

A10.

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

int
```

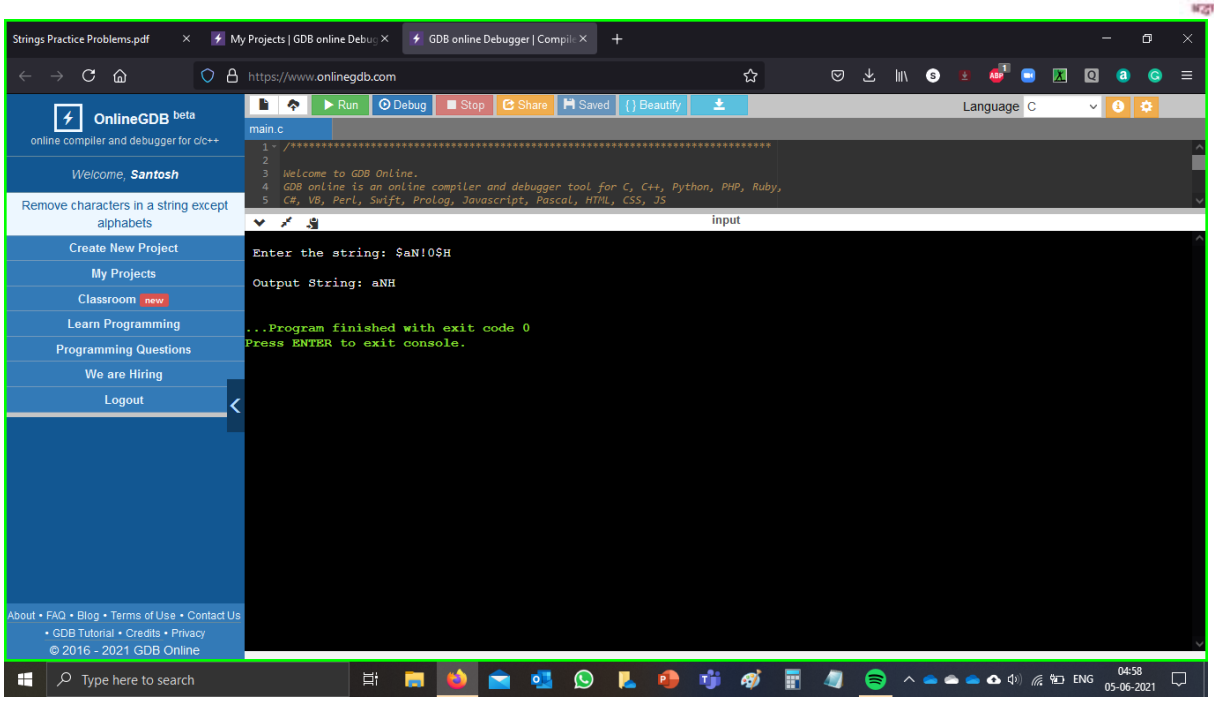
```
main ()
{
  char str[300];
  int i;
  printf ("\nEnter a string: ");
  fgets (str, sizeof str, stdin);
  printf ("\nThe resultant after replacing space(s) with dot(s) is:\n");
  for (i = 0; str[i] != '\0'; i++)
    {
      if (isupper (str[i]))
            putchar (tolower (str[i]));
      else if (islower (str[i]))
            putchar (toupper (str[i]));
      else if (str[i] == ' ')
            putchar (str[i] = '.');
      else
            putchar (str[i]);
    }
  return 0;
}
```



Q11. To remove characters in a string except alphabets.

A11.

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

int
main ()
{
  char str[50];
  int i, j;
  printf ("\n Enter the string: ");
  fgets (str, sizeof (str), stdin);
  for (i = 0; str[i] != '\0'; i++)
    {
      while (!(isalpha (str[i])) && !(str[i] == '\0'))
            {
              for (j = i; str[j] != '\0'; j++)
                str[j] = str[j + 1];
              str[j] = '\0';
            }
    }
  printf ("\n Output String: ");
  puts (str);
  return 0;
}
```

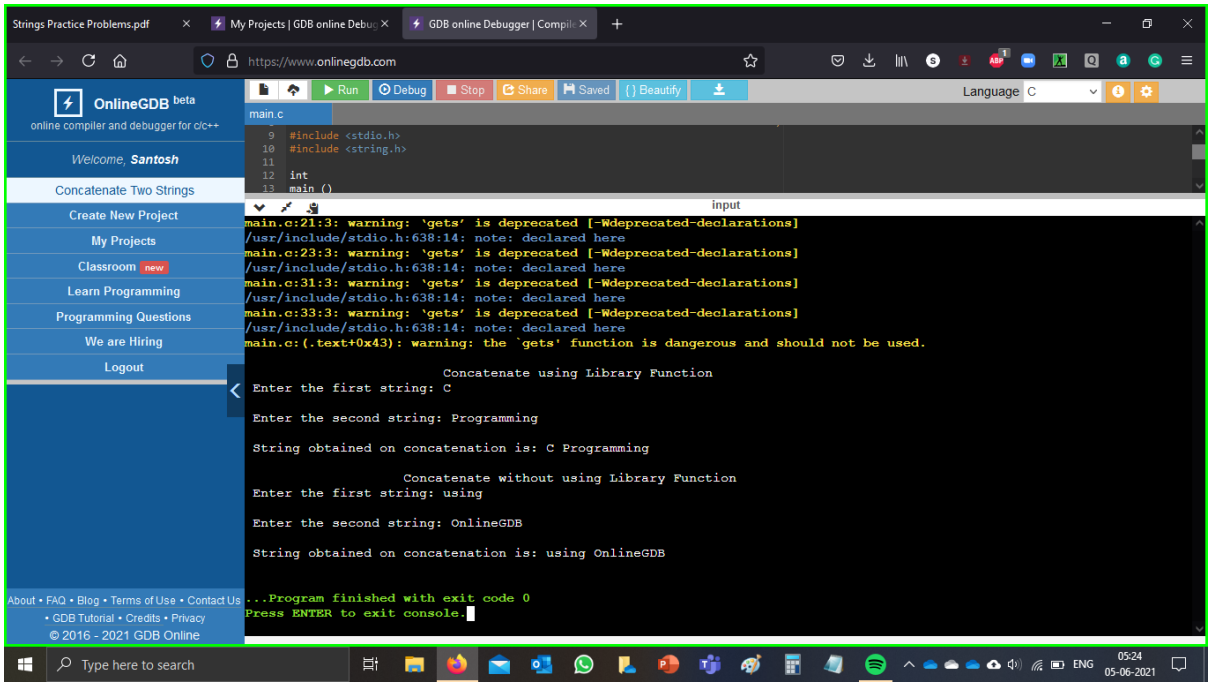


Q12. To concatenate two Strings with and without using library functions.

A12.

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

int
main ()
{
  char str1[50], str2[50], str3[50], str4[50];
  int length, i;

  //Using Library Function
  printf ("\n\t\t\t Concatenate using Library Function");
  printf ("\n Enter the first string: ");
  gets (str1);
  printf ("\n Enter the second string: ");
  gets (str2);
  strcat (str1, str2);
  printf ("\n String obtained on concatenation is: %s", str1);
```

```
printf ("\n");

//Without Using Library Function
printf ("\n\t\t    Concatenate without using Library Function");
printf ("\n Enter the first string: ");
gets (str3);
printf ("\n Enter the second string: ");
gets (str4);
length = 0;
while (str3[length] != '\0')
  length++;
for (i = 0; str4[i] != '\0'; i++, length++)
  str3[length] = str4[i];
str3[length] = '\0';
printf ("\n String obtained on concatenation is: ");
puts (str3);

return 0;
}
```



Thank You!