|  |  |  |
| --- | --- | --- |
| **Name: Jacob V Sanoj** | **SRN: PES1UG20EC083** | **Section: F1** |
| **Date: 8/06/2021** | **Week Number: 5** |

|  |  |
| --- | --- |
| **1** | 1)Write functions to  a) Reverse a string.  b) Check for equality of strings.  **Input1:**  Enter string  abbcbba  **Output1:**  Reversed string is = abbcbba  Given string is abbcbba is palindrome  **Input2:**  Enter string  hi  **Output2:**  Reversed string is = ih  Given string is hi is not palindrome |
|  | Program:  Client1.c  #include <stdio.h>  #include "Question1\_palindrome.c"  int main()  {  char s[20];  char rev[20];  printf("Enter the string\n");  scanf("%s", s);  reverse\_string(s, rev);  int r = check\_equal(s, rev);  if (r == 0)  printf("%s is a palindrome\n");  else  printf("%s is not a palindrome\n");  }  Server1.c  #include<stdio.h>  void reverse\_string(const char \*s1,char \*s2)  {  int len=0;  while(\*s1!='\0')  {  ++s1;  ++len;  }  printf("%d",len);  while(len>0)  {  \*s2++=\*(--s1);  len--;  }  \*s2='\0';  }  int check\_equal(const char\*s1,const char \*s2)  {  while(\*s1 && \*s2 && \*s1==\*s2)  {  s1++;  s2++;  }  return \*s1-\*s2;  }  1.h  void reverse\_string(const char \*s1,char \*s2);  int check\_equal(const char\*s1,const char \*s2);  #include <stdio.h>  void reverse\_string(const char \*s1, char \*s2)  {  int len = 0;  while (\*s1 != '\0')  {  ++s1;  ++len;  }  printf("%d\n", len);  while (len > 0)  {  \*s2++ = \*(--s1);  len--;  }  \*s2 = '\0';  }  int check\_equal(const char \*s1, const char \*s2)  {  while (\*s1 && \*s2 && \*s1 == \*s2)  {  s1++;  s2++;  }  return \*s1 - \*s2;  } |
|  | **Output Screenshot:** |
| **2** | Write function to find all occurrences of a character in a string and use this function to replace all occurences of a character by specific character.  Input1:  Enter the string : Welcome to C programming  Enter a character to replace: o  Enter character to replace with r : @  Output1:  Before replace: Welcome to C programming  After replace: Welc@me t@ C pr@gramming |
|  | Program:  #include <stdio.h>  #include <stdlib.h>  int main()  {  char character;  char element;  char a[20];  printf("Enter the string\n");  scanf("%s", a);  printf("Enter the character to be replaced\n");  scanf("%s", &element);  printf("Enter the character that replaces the above character\n");  scanf("%s", &character);  for (int i = 0; a[i] != '\0'; i++)  {  if (a[i] == element)  {  a[i] = character;  }  }  printf("%s\n", a);  return 0;  } |
|  | **Output Screenshot:** |
| **3** | Write a function to remove all repeated characters from a given string and display the string without duplicate characters.  **Input 1:**  Enter any string: hello world  **Output 1:**  String before removing duplicates: hello world  String after removing duplicates: helo wrd  **Input 1:**  Enter any string: programming in c  **Output 1:**  String before removing duplicates: programming in c  String after removing duplicates: progamin c |
|  | Program:  Client3.c  #include <stdio.h>  #include "Question3\_duplicate.c"  int main()  {  char str[20];  printf("Enter the string\n");  scanf("%[^\n]s", str);  printf("string before removing duplicates is %s\n", str);  removeduplicates(str);  printf("string after removing duplicates is %s\n", str);  return 0;  }  Server3.c  #include <stdio.h>  void removeall(char \*str, char remove, int index)  {  int i;  while (str[index] != '\0')  {  if (str[index] == remove)  {  i = index;  while (str[i] != '\0')  {  str[i] = str[i + 1];  i++;  }  }  else  index++;  }  }  void removeduplicates(char \*str)  {  int i = 0;  while (str[i] != '\0')  {  removeall(str, str[i], i + 1);  i++;  }  } |
|  | **Output Screenshot:** |
| 4 | Write function to Concatenate two strings and use this to concatenate n (i.e, say 2) strings.  **Input 1:**  Enter 1st string  pes  Enter 2nd string  university  Enter number of times u want to append  1  **Output1:**  Concatenated string is pesuniversity  **Input2:**  Enter 1st string  pes  Enter 2nd string  university  Enter number of times u want to append  2  **Output2:**  Concatenated string is pesuniversityuniversity |
|  | Program:  #include <stdio.h>  #include "Question4\_repeat.c"  int main()  {  char s1[100], s2[100];  int n;  printf("enter the 1st string\n");  scanf("%s", s1);  printf("enter the 2nd string\n");  scanf("%s", s2);  printf("Enter the number of times you want to append\n");  scanf("%d", &n);  printf("Concatenated string is %s\n", my\_strncat(s1, s2, n));  return 0;  }  #include <stdio.h>  void my\_strcat(char \*d, const char \*s)  {  while (\*d)  {  d++;  }  while (\*d++ = \*s++);  }  char \*my\_strncat(char \*d1, const char \*s1, int n)  {  for (int i = 0; i < n; i++)  {  my\_strcat(d1, s1);  }  return d1;  } |
|  | Output Screenshot: |
| 1 | **Practice Programs**  Write a function to count the number of occurrences of a given character. Use this to find the number of occurrences of every character in a word.  **Input:**  pesit pes!  **Output:**  i occurs is 1 times  t occurs is 1 times  occurs is 1 times  p occurs is 2 times  e occurs is 2 times  s occurs is 2 times  ! occurs is 1 times |
|  | Program:  #include <stdio.h>  #include "practice1\_counting.c"  int main()  {  char s[20];  printf("Enter the string: \n");  scanf("%[^\n]s", s);  countchar(s);  return 0;  }  #include <stdio.h>  #include <string.h>  int countchar(char \*s)  {  int i, j, count;  int len = strlen(s);  for (i = 0; i < len; i++)  {  count = 0;  for (j = 0; j < len; j++)  if (s[i] == s[j] && s[j] != '\0')  count++;  if (count > 1)  {  for (j = 0; j < len; j++)  if (s[i] == s[j] && i != j)  s[j] = '\0';  }  if (s[i] != '\0')  {  printf("%c occurs %d times.\n", s[i], count);  }  }  } |
|  | Output Screenshot: |
| 2 | Write the function strend (s , t ), which returns 1 if the string t occurs at the end of the string s, and zero otherwise.  **Input1:**  hello world!  world  **Output 1:**  0  **Input2:**  hello world! world  world  **Output 2:**  1 |
|  | Program:  #include <stdio.h>  #include "practice2\_tcheck.c"  int strend(char \*s, char \*t);  int main()  {  char s[50], t[50];  int result;  printf("Enter the string 1: \n");  scanf("%[^\n]s", s);  printf("Enter the string 2: \n");  scanf("%s", t);  result = strend(s, t);  printf("Result is: %d\n", result);  return 0;  }  #include <string.h>  int strend(char \*s, char \*t)  {  int i, j, len\_s, len\_t, res, count = 0, flag = 0;  len\_s = strlen(s);  len\_t = strlen(t);  for (i = len\_s; i > 0; i--)  {  if (s[i] == ' ')  {  count++;  for (j = 0; j < len\_t; j++)  {  if (s[i + 1] == t[j])  {  i++;  }  else  {  flag++;  }  }  res = i;  }  if (count > 0)  {  i = 0;  }  }  if (count > 0 && flag > 0 && res == len\_s)  return 0;  else if (count > 0 && flag == 0 && res == len\_s)  return 1;  else  return 0;  } |
|  | Output Screenshot: |