|  |  |
| --- | --- |
| **Name** | **Shubhan Singh** |
| **UID no.** | **2022300118** |
| **Experiment No.** | **7** |

|  |  |
| --- | --- |
| **Program 1** | |
| **PROBLEM STATEMENT :** | *Write a program to count the number of vowels, consonants, total characters and words in the given string.* |
| **ALGORITHM:** | Step 1: Start  Step 2: Initialize 4 variables vowels, words, consonants and characters to 0,1,0 and 0 respectively.  Step 3: Initialize a string str.  Step 4: Read the sentence from input.  Step 5: initialize i to 0.  Step 6: if str[i] equals 32, increment words by 1.  Step 7: if str[i] is in the ascii range for lower/uppercase alphabets, go to step 8, else jump to step 10.  Step 8: if str[i] is in the ascii range for lower/uppercase vowels, increment vowels by 1.  Step 9:else increment consonants by 1.  Step 10: if str[i] lies in the ascii range for special characters, increment characters by 1.  Step 11: print the count of words, vowels, consonants and special characters in the sentence.  Step 12: END |
| **PROGRAM:** | #include <stdio.h>  int main()  {      int vowels = 0, words = 1, consonants = 0, characters = 0;      char str[9999];      printf("Enter the sentence:\n");      gets(str);      for (int i = 0; str[i] != 0; i++)      {          if (str[i] == 32){words++;}          if (str[i] >= 65 && str[i] <= 90 || str[i] >= 97 && str[i] <= 122){              if (str[i] == 97 || str[i] == 101 || str[i] == 105 || str[i] == 111 || str[i] == 117 || str[i] == 65 || str[i] == 69 || str[i] == 73 || str[i] == 79 || str[i] == 85)              {  vowels++;}              else              {consonants++;}          }          if (str[i] > 32 && str[i] <= 47 || str[i] >= 58 && str[i] <= 64 || str[i] >= 91 && str[i] <= 96 || str[i] >= 123 && str[i] <= 126)          {characters++;}      }      printf("Number of vowels : %d\nNumber of words : %d\nNumber of consonants : %d\nNumber of special characters : %d\n", vowels, words, consonants, characters);      return 0;  } |
| **RESULT:** | |
| **Program 2** | |
| **PROBLEM STATEMENT :** | ***Write a Menu driven Program to***  ***i)****copy one string to another character by character.*  ***ii)****Find the string length*  ***iii)****compare two strings*  ***iv)****reverse the string*  ***v)****Concatenate one string to another string.*  ***vi)****lower case to upper*  *(Do not use library functions)* |
| **ALGORITHM:** | *Algorithm for function to find string length:* ***int stringlength(char\* str1)***  *Arguments: a string str1*.  Step 1: initialize variables I and length to 0.  Step 2: if str[i] id not equal to ‘\0’, continue to step 3, else jump to step 5.  Step 3: increment length and i by 1,  Step 4: return to step 2.  Step 5: return value of length.  *Algorithm for function to copy one string to another:* ***void strcopy(char\* str1,char\* str2)***  *Arguments: source string str2, destination string str1*  Step 1: initialize a variable len to stringlength(str1).  Step 2: increment len by 1.  Step 3: initialize a variable i to 0  Step 4: set str1[i]=str2[i]  Step 5: increment i by 1  Step 6: if i<len, return to step 4.  *Algorithm for function to reverse a string:* ***void stringrev(char\* str1)***  *Arguments: a string str1*  Step 1: initialize a variable len to stringlength(str1)  Step 2: declare a variable temp  Step 3: initialize a variable i to 0  Step 4: set temp=str1[i]  Step 5: set str1[i]=str1[len-i-1]  Step 6: set str1[len-i-1]=temp  Step 7: increment i  Step 8: if i<len/2, return to step 4  *Algorithm for function to concatenate two strings:* ***void concatentates(char \* str1,char \* str2)***  *Arguments: main string str1, appended string str2*  Step 1: initialize two variables len1 and len2 and set them to stringlength(str1) and stringlength(str2) respectively.  Step 2: reallocate size of str1 to len1+len2+1  Step 3: initialize a variable i to len1  Step 4: set str1[i] = str2[i-len1]  Step 5: increment i by 1  Step 6: if i<len1+len2, return to step 4  Step 7: set str1[len1+len2]=’\0’  *Algorithm for function to convert lowercase characters in a string to uppercase:*  ***void capitalisestr(char \*str)***  *Arguments: a string str*  Step 1: initialize an integer len to stringlength(str)  Step 2: initialize a variable i to 0  Step 3: if 97<=str[i]<=122, set str[i]=str[i]-32, else jump to step 4  Step 4: increment i  Step 5: if i<len, return to step 3  *Algorithm to compare two string lexicographically:* ***void stringcompare(char \*str1,char\*str2)***  *Arguments: two strings str1 and str2*  Step 1: declare three variables res, len1 and len2  Step 2: set len1 and len2 to stringlength(str1) and strniglength(str2) respectively.  Step 3: initialize a variable min to the lesser value among len1 and len2.  Step 4: initialize a variable i to 0  Step 5: if str1[i]>str2[i], set res=1 and jump to step 8, else continue to step 6  Step 6: if str[i]>str2[2], set res to -1 and jump to step 8, else continue to step 7  Step 7: if len1>len2, set res=1, else if len1<len2, set res=-1, else set res=0  Step 8: print if res=0, that the strings are equal, else if res=1, then string 1 is lexicographically greater, else string 2 is lexicographically greater. |
| **PROGRAM:** | #include<stdio.h>  #include<stdlib.h>  int stringlength(char\* *str1*){      int i=0,length=0;      while((\*(*str1*+i))!='\0'){          length++;          i++;      }      return length;  }  void strcopy(char\* *str1*,char\* *str2*){      int len=stringlength(*str1*);      len++;      for(int i=0;i<len;i++){  *str1*[i]=*str2*[i];      }  }  void stringrev(char\* *str1*){      int len=stringlength(*str1*);      int temp;      for(int i=0;i<len/2;i++){          temp=*str1*[i];  *str1*[i]=*str1*[len-i-1];  *str1*[len-i-1]=temp;      }  }  void concatentates(char \* *str1*,char \* *str2*){      int len1=stringlength(*str1*);      int len2=stringlength(*str2*);  *str1*=(char \*)realloc(*str1*,(len1+len2+1)\*sizeof(char));      for(int i=len1;i<len1+len2;i++){  *str1*[i]=*str2*[i-len1];      }  *str1*[len1+len2]='\0';  }  void capitalisestr(char \**str*){      int len=stringlength(*str*);      for(int i=0;i<len;i++){          if(*str*[i]>=97 && *str*[i]<=122){  *str*[i]=(char)*str*[i]-32;          }      }  }  void stringcompare(char \**str1*,char\**str2*){      int res;      int len1=stringlength(*str1*);      int len2=stringlength(*str2*);      int min=len1>len2?len1:len2;      for(int i=0;i<min;i++){          if(*str1*[i]>*str2*[i]){              res=1;              break;          }          if(*str1*[i]<*str2*[i]){              res=-1;              break;          }      }      if(len1>len2){res=1;}      else if(len1<len2){res=-1;}      else{res=0;}      if(res==0){          printf("The strings are equal\n");      }      else if(res==1){          printf("The first string comes first lexicographically\n");      }      else{          printf("The second string comes first lexicographically\n");      }  }  int main(){      int n,code,temp;      char \*str;      char \*str2;      str=(char \*)malloc(250\*sizeof(char));      str2=(char \*)malloc(250\*sizeof(char));      printf("Enter first string:\n");      scanf("*%*[^\n]%\*c",str);      printf("Enter second string:\n");      scanf("*%*[^\n]%\*c",str2);      str=(char \*)realloc(str,(stringlength(str)+1)\*sizeof(char));      str2=(char \*)realloc(str2,(stringlength(str2)+1)\*sizeof(char));      printf("Legend for actions:\n 1:Print length of string\n 2:copy strings\n 3:reverse strnig\n 4:Concatenate string\n 5:Convert to uppercase\n 6:Compare strings\n");      printf("Enter number of actions to execute:\n");      scanf("%d",&n);      while(n--){          printf("Enter action number: ");          scanf("%d",&code);          switch(code)          {              case 1:printf("Enter number of string whose length you want to find: ");                      scanf("%d",&*temp*);                      if(temp==1){                          printf("Length of string 1 is %d characters\n",stringlength(str));                      }                      else if(temp==2){                          printf("Length of string 2 is %d characters\n",stringlength(str2));                      }                      else{printf("invalid input!\n");}                      break;              case 2:printf("Enter number of source string: ");                      scanf("%d",&*temp*);                      if(temp==1){                          strcopy(str2,str);                          printf("string 1 is: %s\n",str);                          printf("string 2 is: %s\n",str2);                      }                      else if(temp==2){                          strcopy(str,str2);                          printf("string 1 is: %s\n",str);                          printf("string 2 is: %s\n",str2);                      }                      else{printf("invalid input!\n");}                      break;              case 3:printf("Enter string number to reverse: ");                      scanf("%d",&*temp*);                      if(temp==1){                          stringrev(str);                          printf("string 1 is: %s\n",str);                          printf("string 2 is: %s\n",str2);                      }                      else if(temp==2){                          stringrev(str2);                          printf("string 1 is: %s\n",str);                          printf("string 2 is: %s\n",str2);                      }                      else{printf("invalid input!\n");}                      break;              case 4:printf("Enter string number of string to concatenate to: ");                      scanf("%d",&*temp*);                      if(temp==1){                          concatentates(str,str2);                          printf("string 1 is: %s\n",str);                          printf("string 2 is: %s\n",str2);                      }                      else if(temp==2){                          concatentates(str2,str);                          printf("string 1 is: %s\n",str);                          printf("string 2 is: %s\n",str2);                      }                      else{printf("invalid input!\n");}                      break;              case 5:printf("Enter string number to capitalize: ");                      scanf("%d",&*temp*);                      if(temp==1){                          capitalisestr(str);                          printf("string 1 is: %s\n",str);                          printf("string 2 is: %s\n",str2);                      }                      else if(temp==2){                          capitalisestr(str2);                          printf("string 1 is: %s\n",str);                          printf("string 2 is: %s\n",str2);                      }                      else{printf("invalid input!\n");}                      break;              case 6:stringcompare(str,str2);          }      }      free(str);      free(str2);      return 0;  } |
| **RESULT:** | |
| **Program 3** | |
| **PROBLEM STATEMENT:** | *Write a program to find and replace a particular word from the string.* |
| **PROGRAM:** | #include<stdio.h>  #include<stdlib.h>  #include<string.h>  int main(){      char \*\*strarr;      char findstr[250];      char replacestr[250];      char tempstr[250];      int i=0,templen,lenreplace;      printf("Enter the string (all words before newline will be read):\n");      scanf("%s",tempstr);      templen=strlen(tempstr);      strarr=(char\*\*)malloc(sizeof(char\*));      strarr[i]=(char\*)malloc((templen+1)\*sizeof(char));      strcpy(strarr[i],tempstr);      i++;      while(1){          if(getchar()=='\n'){              break;          }          strarr=(char\*\*)realloc(strarr,(i+1)\*sizeof(char\*));          scanf("%s",tempstr);          templen=strlen(tempstr);          strarr[i]=(char\*)malloc((templen+1)\*sizeof(char));          strcpy(strarr[i],tempstr);            i++;      }      printf("Enter word to find:\n");      scanf("%s",findstr);      printf("Enter word to replace:\n");      scanf("%s",replacestr);      lenreplace=strlen(replacestr);      for(int j=0;j<i;j++){          if(strcmp(strarr[j],findstr)==0){              strarr[j]=(char\*)calloc((lenreplace+1),sizeof(char));              strcpy(strarr[j],replacestr);          }      }      for(int k=0;k<i;k++){          printf("%s ",strarr[k]);      }      printf("\n");      free(strarr);      return 0;  } |
| **RESULT:** | |