**Assignment-18-solution**

1.Write a function to calculate length of the string.

#include<stdio.h>

#include<string.h>

int length(char str[])

{

   int i,count=0;

   for(i=0;str[i];i++)

      {

       count++;

      }

   return count-1;

}

int main()

{

   char str[200];

   printf("Enter a string: ");

   fgets(str,200,stdin);

   printf("Length is %d",length(str));

}

2. Write a function to reverse a string.

#include<stdio.h>

#include<string.h>

void reverse(char str[])

{

   int i,count=0;

   char str2[200];

   for(i=0;str[i];i++)

      {

       str2[i]=str[strlen(str)-1-i];

      }

      printf("%s",str2);

}

int main()

{

   char str[200];

   printf("Enter a string: ");

   fgets(str,200,stdin);

   reverse(str);

}

3. Write a function to compare two strings.

#include<stdio.h>

#include<string.h>

int compare(char str[], char str2[])

{

   return strcmp(str,str2);

}

int main()

{

   int cmp;

   char str[200],str2[200];;

   printf("Enter both strings: ");

   fgets(str,200,stdin);

   fgets(str2,200,stdin);

   cmp=compare(str,str2);

   if(cmp==0)

    printf("Both strings are equal.");

   if(cmp==1)

    printf("string 2 comes first.");

   if(cmp==-1)

    printf("string 1 comes first.");

}

4. Write a function to transform string into uppercase.

#include<stdio.h>

#include<string.h>

void upper(char str[])

{

   strupr(str);

   printf("Upper string is: %s",str);

}

int main()

{

   char str[200];

   printf("Enter string: ");

   fgets(str,200,stdin);

   upper(str);

}

5. Write a function to transform a string into lowercase.

#include<stdio.h>

#include<string.h>

void lower(char str[])

{

   strlwr(str);

   printf("lower string is: %s",str);

}

int main()

{

   char str[200];

   printf("Enter string: ");

   fgets(str,200,stdin);

   lower(str);

}

6. Write a function to check whether a given string is an alphanumeric string or not. (Alphanumeric string must contain at least one alphabet and one digit).

#include<stdio.h>

#include<string.h>

int alpnum(char str[])

{

   int i,j=0;

   for(i=0;str[i];i++)

   {

      if((str[i]>='a' && str[i]<='z') || (str[i]>='A' && str[i]<='Z'))

        if(str[i+1]>='0' && str[i+1]<='9')

        {

           j=1;

           break;

        }

   }

   if(j==1)

     return j;

   else

   return 0;

}

int main()

{

   int alp;

   char str[200];

   printf("Enter string: ");

   fgets(str,200,stdin);

   alp=alpnum(str);

   if(alp==1)

     printf("yes alphanumeric.");

   else

    printf("No not an alphanumeric.");

}

7. Write a function to check whether a given string is palindrome or not.

#include<stdio.h>

#include<string.h>

void compare(char str[])

{

   int l,k=1,i;

   l=strlen(str);

   for( i=0;i<l/2;i++)

   {

     if(str[i]!= str[l-2-i])

       {

         printf("not palindrom");

         k=0;

         break;

       }

   }

   if(k==1)

     printf("string is palindrom");

}

int main()

{

   int cmp;

   char str[200];

   printf("Enter both strings: ");

   fgets(str,200,stdin);

   compare(str);

}

8. Write a function to count words in a given string.

#include<stdio.h>

#include<string.h>

void word(char str[])

{

  int i,count=0;

  for(i=0;str[i];i++)

  {

   if(str[i]==' ' && str[i+1]!=' ')

    {

       count=count+1;

    }

  }

  printf("%d word",count);

}

int main()

{

   int cmp;

   char str[200];

   printf("Enter both strings: ");

   fgets(str,200,stdin);

   word(str);

}

9.