programming and structures using c

Assignment 8

1. read from a terminal using scanf function and print using printf function.   
  
#include <stdio.h>  
#include<string.h>  
int main() {  
    char name[50];  
    printf("please enter your name:\n");  
    scanf("%s",name);  
    printf("your name is %s",name);  
    return 0;  
}  
Output-  
please enter your name:  
sagarika behera  
Your name is sagarika  
  
  
2.read a lines of text from a terminal using fgets function and print using puts function.   
  
#include  <stdio.h>  
#include  <string.h>     
int main(){  
    char  name[50];  
    printf("please enter your name:\n");  
     fgets(name,sizeof(name),stdin);  
     printf("your name is:");   
     puts(name);         
     return  0;       
}  
Output-  
please enter your name:  
Sagarika behera  
Your name is: sagarika behera  
  
3. convert   
a. Upper case to Lower case   
b. Lower case to Upper case   
  
#include<stdio.h>    
#include <string.h>    
  int main(){     
char str[20]="sagarika";  
printf("String is: %s",str);    printf("UpperString is:%s",strupr(str));   
  printf("\nLowerString is: %s",strlwr(str));   
   return 0;    
  }      
Output-  
UpperString is: SAGARIKA  
LowerString is:sagarika  
  
  
4. perform String Concatenation (With and Without String Handling Functions).   
#include<stdio.h>     
#include  <string.h>     
int main(){     
    int l,i=0;  
    char a;  
    char ch1[20]="sagarika";  
    char ch2[20]={' ','b','e','h','e','r','a','\0'};         
    l=strlen(ch1);  
    printf("please choose how you want to concatenate the strings.\n  1)with library function type 'y'\n 2)without library function type 'n'\n");  
    scanf("%c",&a);  
    switch(a)  
    {   
     case'y':  
    while(ch2[i]!='\0')  
    {  
        ch1[l]=ch2[i];  
        i++;  
        l++;  
    }  
    printf("the final string  is:  %s",ch1);     
    break;  
    case'n':  
    strcat(ch1,ch2);    
    printf("the final string  is:  %s",ch1);    
    break;  
    default:  
    printf("error!!!");  
    }  
    return  0;     
      
    }         
Output-  
please choose how you want to concat  
enate the strings.                    
  1)with library function type 'y'    
2)without library function type 'n'  
y                                     
the final string  is:  sagarika behera 

5. perform String Reversal (With and Without String Handling Functions).   
#include<stdio.h>  
#include <string.h>  
int main(){  
    char a;  
    int i=0,j=0;  
char str1[20]="sagarika",str2[20];  
printf("please choose how you want to reverse the string.\n  1)with library function type 'y'\n 2)without library function type 'n'\n");  
    scanf("%c",&a);  
    switch(a)  
    {  
      case 'y':  
printf("\nReverse String is: %s",strrev(str1));  
break;  
      case 'n':  
      j=strlen(str1);  
      while(str1[i]!='\0')  
      {  
          str2[j-1]=str1[i];  
          i++;  
          j--;  
            
      }  
      printf("\nReverse String is %s");  
(j=0;j<strlen(str1);j++)  
      printf("%c",str2[j]);   
                break;  
          default:  
          printf("error!!!");  
}  
return 0;  
}  
Output-  
"please choose how you want to reverse the strings.  
with library function type 'y'  
without library function type 'n'  
Reverse String is:akiragas

6. perform Substring Extraction (With and Without String Handling Functions).   
#include <stdio.h>  
void main()   
{  
   char str[100], ext[100];  
   int pos, l, c = 0;  
       printf("enter the string : \n");  
       fgets(str, sizeof(str), stdin);  
  
   printf("enter the position to start extraction :\n");  
   scanf("%d", &pos);  
     
   printf("enter the length of substring :\n");  
   scanf("%d", &l);  
  
   while (c < l)   
   {  
      ext[c] = str[pos+c-1];  
      c++;  
   }  
   ext[c] = '\0';  
  
   printf("The substring retrieved from the string is :  %s", ext);  
  
}  
Output-  
Enter the string:  
Sagarika  
enter the position to start extraction:  
4  
Enter the length of substring:  
3  
The substring retrieved from the string is:ari

Q7. Copy one string into another and count the no of elements copied. (With and Without String Handling Functions).  
  
• Without string handling function:  
#include<stdio.h>  
        int main()  
                {  
        char str1[20], str2[20];  
         int i;  
         printf("Input a string: ");  
         scanf("%s", str2);  
         for(i=0; str2[i]!='\0'; i++)  
          {str1[i]=str2[i];  
          }  
          str1[i]="\0";  
         printf("Original string: %s", str1);  
         printf("\nNumber of characters = %d\n", i);  
         return 0;  
         }  
Output-  
Input a string: saga                  
Original string: saga              
Number of characters = 4              
                             
  
          
•using string handling functions:  
#include <stdio.h>  
#include<string.h>  
int main()  
{  
char text1[100];  
char text2[100];  
int i;  
printf("Enter any string: ");  
Scanf("%s",text1);  
Strcpy(text2,text1);  
printf("First string = %s\n", text1);  
printf("Second string = %s\n", text2);  
printf("Total characters copied = %d\n"strlen(text2) );  
return 0;  
}  
Output-  
Enter any string: saga                
First string = saga                   
Second string = saga                  
Total characters copied = 4           
  
Q8. Read a string and prints if it is a palindrome or not.   
#include <stdio.h>  
#include <string.h>  
int main()  
{  
    char string1[20];  
    int i, length;  
    int flag = 0;  
    printf("Enter a string:");  
    scanf("%s", string1);  
    length = strlen(string1);  
    for(i=0;i < length ;i++)  
{  
        if(string1[i] != string1[length-i-1])  
{  
            flag = 1;  
            break;  
   }  
}  
    if (flag) {  
        printf("%s is not a palindrome", string1);  
    }      
    else {  
        printf("%s is a palindrome", string1);  
    }  
    return 0;  
}       
Output-  
Enter a string:illi                   
illi is a palindrome  

Q9. Read a line of text and count all occurrences of particular word.   
#include<stdio.h>  
#include <string.h>  
int main()  
{  
    char s[100],w[100];    
    int n,a[100],i,j,k=0,l,found=0,t=0;  
    printf("Enter  the string : ");  
    gets(s);  
    printf("Enter word to be searched: ");  
    gets(w);  
    for(i=0;s[i];i++)  
    {  
    if(s[i]==' ')  
{  
a[k++]=i;  
}  
}  
a[k++]=i;  
j=0;  
for(i=0;i<k;i++)  
{  
n=a[i]-j;  
if(n==strlen(w))  
{  
t=0;  
for(l=0;w[l];l++)  
{  
if(s[l+j]==w[l])  
{  
t++;  
}  
if(t==strlen(w))  
{  
found++;  
}  
}  
j=a[i]+1;  
}  
printf("word '%s' is occurred count=%d ",w,found);  
return 0;  
}  
Output-  
Enter  the string : saga rika            
Enter word to be searched: rika  
Word rika is  
occurred count=6  
                  
                          
Q10. Read a string and rewrite it in the alphabetical order.  
#include <stdio.h>  
#include <string.h>  
void main()  
{  
char str[100],ch;  
int i,j,l;  
printf("Input the string : ");  
fgets(str, sizeof str, stdin);  
l=strlen(str);  
for(i=1;i<l;i++)  
for(j=0;j<l-i;j++)  
if(str[j]>str[j+1])  
{  
ch=str[j];  
str[j] = str[j+1];  
str[j+1]=ch;  
}  
printf("After sorting the string appears like : \n");  
printf("%s\n\n",str);  
}  
Output-  
Input the string : sagarika           
After sorting the string appears like :                                                                    
aaagikrs                           

Q12. Delete All Repeated Words in the line of text.  
#include <stdio.h>  
#include <string.h>  
int main()  
{  
char str[100];  
int i, j, k;  
printf("\n Please Enter any String :  ");  
gets(str);  
for(i = 0; i < strlen(str); i++)  
{  
for(j = i + 1; str[j] != '\0'; j++)  
{  
if(str[j] == str[i])    
{  
for(k = j; str[k] != '\0'; k++)  
{  
str[k] = str[k + 1];  
}  
}  
}  
}  
printf("\n The Final String after Removing All Duplicates = %s ", str);  
return 0;  
}  
Output-  
Please Enter any String :  sagarika  
                                      
The Final String after Removing All  
Duplicates = sagrik                  
                           
  
  
  
  
  
  
  
  
  
  
  
  
  
                                  
                       