ASSIGNMENT - 08

Q1. Read from a terminal using scanf function and printf using printf function.

```
#include <stdio.h>
int main() {
int testInteger;
printf("Enter an integer: ");
scanf("%d", &testInteger);
printf("Number = %d",testInteger);
return 0;
}
```

OUTPUT :

```
Enter an integer: 5
Number = 5
```

Q2. Read a line of text from a terminal using fgets function and print using puts function.

```
#include <stdio.h>
int main() {
char name[50];
 printf("Enter your name :");
gets(name);
printf("Your name is :");
puts(name);
return 0;
}
```

OUTPUT :

```
Enter your name :SAIMA NASIM
Your name is :SAIMA NASIM
```

Q3. Convert : a)upper case to

```
#include <stdio.h>
#include<string.h>int main() {
char str[25];
int i;
printf("Enter the string: ");
scanf("%s",str);
for(i=0;i<=strlen(str);i++){
if(str[i]>=65&&str[i]<=90)
str[i]=str[i]+32;    }
printf("\nLower Case String is: %s",str);
return 0;
}
```

OUTPUT :

```
Enter a string : SAIMANASIM

String in Lower Case = saimanasim
```

b) lowercase to upper case

```
#include <stdio.h>
#include<string.h>
int main() {
char str[100];
int i;
printf("\nEnter a string : ");
gets(str);
for (i = 0; str[i]!='\0'; i++) {
if(str[i] >= 'a' && str[i] <= 'z') {
str[i] = str[i] - 32;      }   }
printf("\nString in Upper Case = %s", str);
return 0;}
```

OUTPUT :

```
Enter a string : saimanasim

String in Upper Case = SAIMANASIM
```

C) toggle case

```
#include <stdio.h>
#include<string.h>
int main() {
char Str1[100];
int i;
printf("\n Please Enter any String to Toggle : ");
gets(Str1);
for (i = 0; Str1[i]!='\0'; i++)        {
if(Str1[i] >= 'a' && Str1[i] <= 'z')                    {
Str1[i] = Str1[i] - 32;                 }
else if(Str1[i] >= 'A' && Str1[i] <= 'Z')               {
Str1[i] = Str1[i] + 32;                 }          }
printf("\n The Given String after Toggling Case of all Characters = %s", Str1);
return 0;}
```

OUTPUT :

```
Please Enter any String to Toggle : jaVATpoINT

 The Given String after Toggling Case of all Characters = JAvatPOint
```

d) sentence case.

```
#include <stdio.h>
#include <string.h>
void StrToSentence(char * string){
int length=0,i=0;
length = strlen(string);
for(i=0;i<length;i++)     {
if( (i==0) && (string[i]>='a' && string[i]<='z'))                {
string[i] = string[i] - 32;                                  }
else if(string[i]=='.')                        {
if(string[i+1] == ' ')                                  {
if(string[i+2]>='a' && string[i+2]<='z')                                                  {
string[i+2] = string[i+2] - 32; }}      else                                  {
  if(string[i+1]>='a' && string[i+1]<='z')                                                  {
string[i+1] = string[i+1] - 32;}}}}/
 functionint  main(){
char string[50]={0};
int length=0,i=0,j=0,k=0;
printf("\nEnter  the string : ");
gets(string);
StrToSentence(string);
printf("Final string is : %s",string);
return 0;}
```

OUTPUT :

```
Enter the string : hello world.how are
you
Final string is : Hello world.How are you
```

Q4. perform string concatenation (with and without string handling function)

```
#include <stdio.h>
#include<string.h>
int main() {
char str1[100] = "javat", str2[100] = "point";
char str3[100];
int i = 0, j = 0;
printf("\nFirst string: %s", str1);
printf("\nSecond string: %s", str2);
while (str1[i] != '\0') {
str3[j] = str1[i];
i++;
 j++;     }
 i = 0;
while (str2[i] != '\0') {
str3[j] = str2[i];
i++;
j++;     }
str3[j] = '\0';
printf("\nConcatenated string: %s", str3);
 return 0; }
```

OUTPUT :

```
First string: javat
Second string: point
Concatenated string: javatpoint
```

Q5. Perform string reversal(with and without string handling functions).

```
#include <stdio.h>
#include<string.h>
int main() {
 char str[100], temp;
int i, j = 0;
printf("\nEnter the string :");
gets(str);
i = 0;
j = strlen(str) - 1;
while (i < j) {
temp = str[i];
str[i] = str[j];
str[j] = temp;
i++;
j--;   }
printf("\nReverse string is :%s", str);
 return 0;
}
```

OUTPUT :

```
Enter the string :string
Reverse string is :gnirts
```

Q6. Perform substring extraction(with and without string handling function).

```
#include <stdio.h>
#include<string.h>
int main() {
char string[1000], sub[1000];
int position, length, c = 0;
printf("Input a string\n");
gets(string);
printf("Enter the position and length of substring\n");
scanf("%d %d", &position, &length);
while (c < length) {
sub[c] = string[position+c-1];
c++;   }
sub[c] = '\0';
printf("Required substring is \"%s\"\n", sub);
return 0;
}
```

OUTPUT :

```
Input a string :c program
Enter the position and length of substring
3
8
Required substring is "program"
```

Q7. Copy one string into another and count the no of element copied.(with and without string handling function)

```
#include <stdio.h>
#include<string.h>
int main() {
char s1[100], s2[100];
int i;
printf("\nEnter the string :");
gets(s1);
i = 0;
while (s1[i] != '\0') {
s2[i] = s1[i];
 i++;   }
s2[i] = '\0';
printf("\nCopied String is :%s ", s2);
printf("\nNumber of characters = %d\n", i);
return (0);
}
```

OUTPUT :

```
Enter the string :program
Copied String is :program
Number of characters = 7
```

Q8. Read a string and prints if it is a palindrome or not.

```
#include <stdio.h>
#include <string.h>
int main(){
char s[1000];
int i,n,c=0;
printf("Enter  the string : ");
gets(s);
n=strlen(s);
for(i=0;i<n/2;i++)      {
if(s[i]==s[n-i-1])
c++;    }
if(c==i)
printf("string is palindrome");
else
printf("string is not palindrome");
return 0;}
```

OUTPUT :

```
Enter  the string : madam
string is palindrome
Enter  the string : program
string is not palindrome
```

Q9. Read a line of text and count all occurrences of particular word.

```
#include<stdio.h>
#include<string.h>
int main(){
char s[1000],w[1000];
int n,a[1000],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 : hello world hello earth
Enter word to be searched: hello
word 'hello' is occurred count=2
```

Q10. Read a string and rewrite it in the alphabetical order.

```
#include <stdio.h>
int main(){
char str[100],temp;
int i,j;
printf("Enter the string :");
gets(str);
printf("%s in ascending order is :",str);
for(i=0;str[i];i++){
for(j=i+1;str[j];j++){
if(str[j]<str[i]){
temp=str[j];
str[j]=str[i];
str[i]=temp;}}}
printf("%s\n",str);
return 0;}
```

OUTPUT :

```
Enter the string :  string
String  in ascending order is : ginstr
```

Q11. Print the word ending with letter s.

```
#include <stdio.h>
char str[100];
void main(){
 int i, t, j, len;
printf("Enter a string : ");
scanf("%[^\n]s", str);
len = strlen(str);
str[len] = ' ';
for (t = 0, i = 0; i < strlen(str); i++)    {
 if ((str[i] == ' ') && (str[i - 1] == 's'))        {
for (j = t; j < i; j++)
 printf("%c", str[j]);
t = i + 1;
printf("\n");         }
else        {
 if (str[i] == ' ')            {
  t = i + 1;            }        }
 }}
```

OUTPUT :

```
Enter a string : let start the class
class
```

Q12. Delete all repeated word with in the line of text.

```
#include <stdio.h>
int main(){
char string[] = "big black bug bit a big black dog on his big black nose";
char words[100][100];
int i = 0, j = 0, k, length, count;
for(k=0; string[k]!='\0'; k++){
if(string[k] != ' ' && string[k] != '\0'){
words[i][j++] = tolower(string[k]);          }
else{
words[i][j] = '\0';
i++;
 j = 0;          }       }
length = i+1;
printf("Duplicate words in the given string: \n");
 for(i = 0; i < length; i++){
 count = 1;
 for(j = i+1; j < length; j++){
if(strcmp(words[i], words[j]) == 0 && (strcmp(words[j],"0") != 0)){
count++;
  strcpy(words[j],"0");            }          }
   if(count > 1 )
 printf("%s\n", words[i]);      }
 return 0;  }
```

OUTPUT :

```
Duplicate words in the given
string:
big
black
```