Write an application that performs few operations on a sentence. Implement the application by defining the functions given below. You can define a fixed size character array in main function to hold the sentence.

Requirement Tag	Requirement Description	Comments
STR/01	Write a function to accept a sentence (that is a line) from the user.	Hint: Use fgets function
STR/02	Write a function which accepts the sentence and prints one word in a line. A word is a sequence of non-whitespace characters	The function returns the number of words
STR/03	Write a function to print the longest word in the sentence (assume all words are of different length).	The function returns the size of the longest word.
STR/04	Write a function to prompt and read a search sub string, search for the first word containing this substring in input sentence, remove the word and display the sentence after update.	Hint: Use strstr() to search.
STR/05	Print all words in one line without printing any whitespace	
STR/06	Write a function to replace all the whitespaces with "!". The function returns the number of replacement done.	int replacews(mystr, myreplacement)

Answer:

```
#include <stdio.h>

#include <stdlib.h>
int main()
{
 char str[20];
 fgets(str,20, stdin);
 printf("%s", str);

 return 0;
}
```

Note: The stdin is the short form of the “standard input”, in C programming the term “stdin” is used for the inputs which are taken from the keyboard either by the user or from a file. The “stdin” is also known as the pointer because **the developers access the data from the users or files and can perform an action on them**.

Answer:

```
#include <stdio.h>
#include<string.h>
#define MAX_SIZE 100
int main() {
 // Write C code here
 char str[MAX_SIZE];
 printf("ente a sentence");
 fgets(str,MAX_SIZE,stdin);
 int i,end, start=0;
 for(i=0;i<strlen(str);i++)
 {
 if(str[i] == ' '|| i == strlen(str) - 1)
 {
 end=i;
 printf("%.*s\n",(end - start),str + start);
 start= i+1;
 }
 }
 return 0;
}
```

Answer:

```
#include<stdio.h>
int main()
{
 char string[100]="Tom went to department";
 int i,start=0, longest=0, longest_p=0;
 for(i=0; string[i]!='\0';i++)
 {
 if(string[i]==' ')
 {
 start = i+1;
 }
 else
 {
 if(i - start > longest)
 {
 longest = i - start;
 longest_p = start;
 }
 }
 }
 printf("longest word: %d letters,'%.*s'\n", longest, longest, string +
longest_p);  return 0;
}
```

Answer:5

```
 #include<stdio.h>
#include<string.h>
int main()
{
 char a[1000];
 int i,k=0;
 printf("enter the string");
 gets(a);
 for(i=0;a[i];i++)
 {
 a[i]=a[i+k];
 if(a[i]==' '||a[i]=='\t')
 {
 k++;
 i++;
 }
 }
 printf("string after removing all blank spaces:");
 printf("%s",a);
 return 0;
}
```

Answer:

```
#include <stdio.h>
#include<string.h>

int main()
{
 char s[1000];
 int i,k=0;
 printf("enter the string":);
 gets(s);
 for(i=0;s[i];i++)
 {
 s[i]=s[i+k];
 if(s[i] == ' '|| s[i]=='\t')
 {
 k++;
 i--;

 }
 }

 printf("%s",s);
 return 0;
}
```