Week 5: Programs on Strings,Pointers and Multiple files

2021

Name:AYUSH SINGH	SRN:PES2UG20CS081	Section:B
	Date:09-06-2001	Week Number:5

1	1)Write functions to a) Reverse a string. b) Check for equality of strings. **Input1:** Enter string abbcbba **Output1:** Reversed string is = abbcbba Given string is abbcbba is palindrome **Input2:** Enter string hi **Output2:** Reversed string is = ih Given string is hi is not palindrome

	Program:
	bruh1.c
	#include<stdio.h>
	#include<stdbool.h>
	#include<string.h>
	
	/*
	functions needed strcompare,revstring
	*/
	//prototyping
	void revstring(char *input,char *output,int strlen);
	int strcompare(char *input,char *output,int strlen);

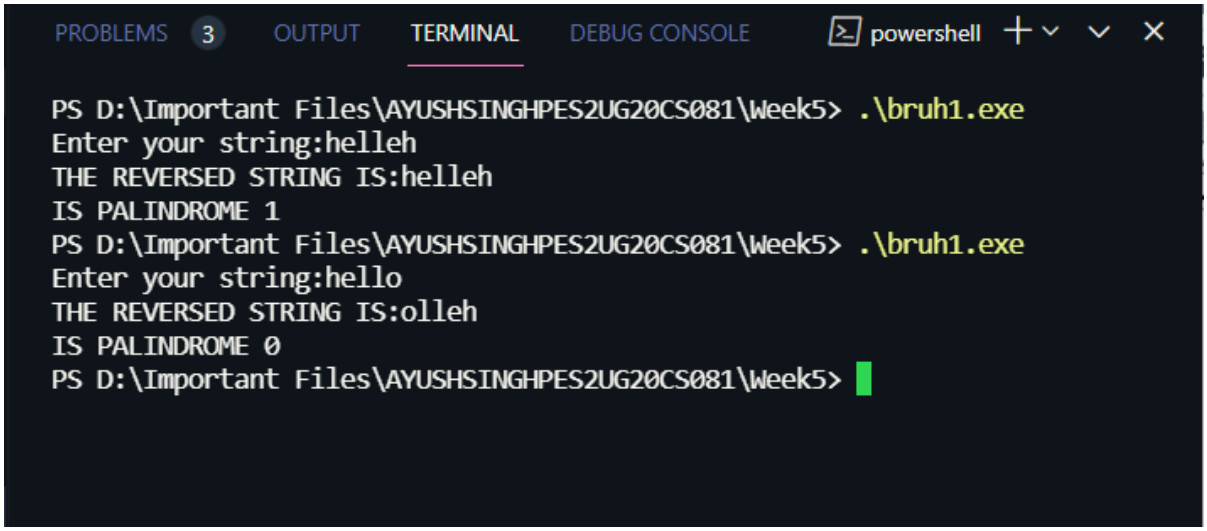
```
int main(){
   char input[256];



   printf("Enter your string:");
   scanf("%[^\n]%*c",input);

   int strlen = 0;
   for (int i = 0; i < 256; i++)
   {

      if (input[i]=='\0')break;
      strlen++;
   }



   char revstr[strlen+1];
   revstr[strlen]='\0';

   revstring(input,revstr,strlen);
   printf("THE REVERSED STRING IS:%s\n",revstr);

   printf("IS PALINDROME %d",strcompare(input,revstr,strlen));

   return 0;

}
```

monke1.c

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

void revstring(char *input,char *output,int strlen){


   for (int i = 0; i < strlen; i++)
   {
      output[i]=input[strlen-1-i];
   }
}

int strcompare(char *input,char *output,int strlen){
```

```
    for (int  i = 0; i < strlen; i++)
    {
       if(input[i]!=output[i])return 0;
    }
    return 1;

}
```

Output Screenshot:



2 Write function to find all occurrences of a character in a string and use this function to replace all occurences of a character by specific character.

Input1:
Enter the string : Welcome to C programming
Enter a character to replace: o
Enter character to replace with r : @
Output1:
Before replace: Welcome to C programming
After replace: Welc@me t@ C pr@gramming

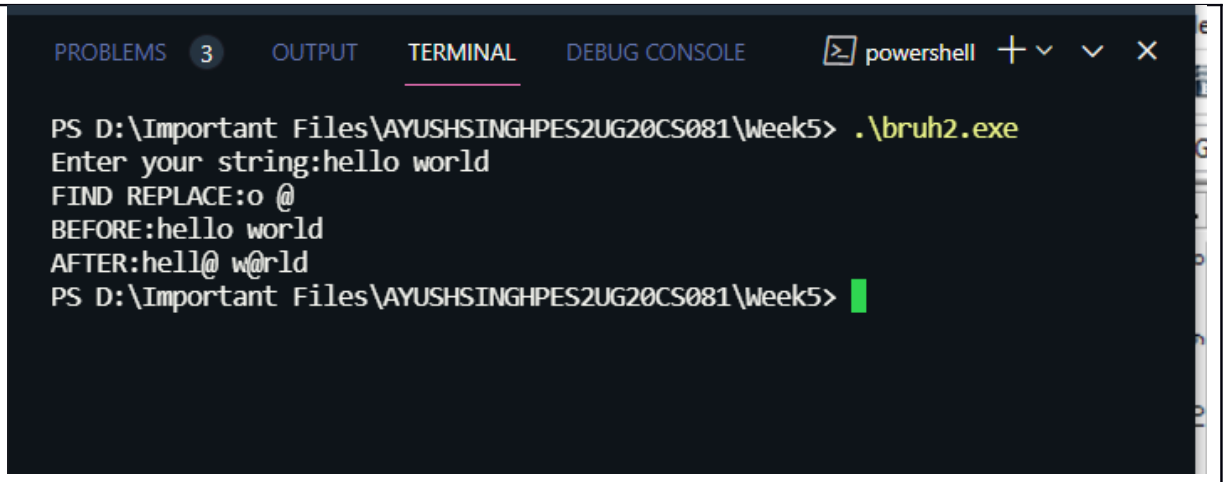
Program:
bruh2.c

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

/*
functions needed strcompare,revstring
*/
void replacechr(char *input,char find,char replace,int strlen);

int main(){
   char input[256];
   printf("Enter your string:");
   scanf("%[^\n]%*c",input);

   int strlen = 0;
   for (int i = 0; i < 256; i++)
   {

      if (input[i]=='\0')break;
      strlen++;
   }

   char find;
   char replace;

   printf("FIND REPLACE:");
   scanf("%c %c",&find,&replace);

   printf("BEFORE:%s\n",input);
   replacechr(input,find,replace,strlen);
   printf("AFTER:%s",input);


   return 0;
}
```

monke2.c

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

void replacechr(char *input,char find,char replace,int strlen){
   for (int i = 0; i < strlen; i++)
   {
      if (input[i]==find)input[i]=replace;
   }
}
```

Output Screenshot:



3

Write a function to remove all repeated characters from a given string and display the string without duplicate characters.

Input 1:

Enter any string: hello world

Output 1:

String before removing duplicates: hello world

String after removing duplicates: helo wrd

Input 1:

Enter any string: programming in c

Output 1:

String before removing duplicates: programming in c

String after removing duplicates: progamin c

Program:

bruh3.c

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

/*
functions needed strcompare,revstring
*/
void removeduplicates(char *input,int strlen);

int main(){
    char input[256];
```

```
    printf("Enter your string:");
    scanf("%[^\n]%*c",input);

    int strlen = 0;
    for (int i = 0; i < 256; i++)
    {

        if (input[i]=='\0')break;
        strlen++;
    }
    removeduplicates(input,strlen);



}
```

monke3.c

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

int in(char *input,char find,int strlen){
    int i=0;
    while (i<strlen)
    {
        if (input[i]==find)return 1;
        i++;
    }

    return 0;
}

void removeduplicates(char *input,int strlen){
    char dump[strlen];
    int index=0;

    for (int i = 0; i < strlen; i++)
    {

        if (in(dump,input[i],strlen))
        {
            input[i]='\0';
        }
        else{
            dump[index]=input[i];
            index++;
        }

    }
```

```
    char ans[strlen];
    int count=0;
    for (int i = 0; i < strlen; i++)
    {

       if(input[i]!='\0'){
          ans[count]=input[i];
          count++;
       }
    }
    for (int i = 0; i < count; i++)
    {
       printf("%c",ans[i]);
    }



}
```

Output Screenshot:

```
PS D:\Important Files\AYUSHSINGHPES2UG20CS081\Week5> .\bruh3.exe
Enter your string:hello world
h
e
l

o

w

r

d
helo wrd
PS D:\Important Files\AYUSHSINGHPES2UG20CS081\Week5>
```

4	Write function to Concatenate two strings and use this to concatenate n (i.e, say 2) strings. **Input 1:** Enter 1st string pes Enter 2nd string university Enter number of times u want to append 1 **Output1:** Concatenated string is pesuniversity **Input2:** Enter 1st string pes Enter 2nd string university Enter number of times u want to append 2 **Output2:** Concatenated string is pesuniversityuniversity

	Program:
	bru4.c
	#include <stdio.h>
	void concatenateNTimes(char *input1, char *input2, int l1, int l2, int n);
	int main()
	{
	char string1[500], string2[500];
	int length1 = 0, length2 = 0, repeat = 0;
	char temp;
	
	printf("Enter the first string : ");
	while ((temp = getchar()) != EOF && temp != '\n')
	{
	string1[length1] = temp;
	length1++;
	}
	printf("Enter the second string : ");
	while ((temp = getchar()) != EOF && temp != '\n')
	{
	string2[length2] = temp;
	length2++;
	}
	printf("Enter the number of repeats : ");
	scanf("%d", &repeat);
	concatenateNTimes(string1, string2, length1, length2, repeat);
	return 0;
	}

monke4.c

```
#include <stdio.h>
void concatenateNTimes(char *input1, char *input2, int l1, int l2, int n)
{
   // printf("%s %d %s %d %d\n", input1, l1, input2, l2, n);
   int new_length = l1 + (n * l2);
   printf("The new length is %d\n", new_length);
   char new_string[new_length];

   for (int i = 0; i < new_length; i++)
   {
      new_string[i] = '*';
   }

   for (int i = 0; i < l1; i++)
   {
      new_string[i] = input1[i];
   }
   // printf("%s\n", new_string);

   for (int i = 0; i < n; i++)
   {
      int offset = l1 + (i * l2);
      for (int j = 0; j < l2; j++)
      {
         new_string[offset + j] = input2[j];
      }
      // printf("%s\n", new_string);
   }
   printf("The repeated string is > ");
   printf("%s\n", new_string);
}
```

Output Screenshot: