VJWTSS08190012

BOOT CAMP DAY 4

Program :1 //count of words and characters in a string//

Ans :

#include <stdio.h>

#include <string.h>

main()

{

char str[20],\*p;

int i=0, word=0, chr=0;

printf("\nEnter Your String: ");

gets(str);

p=str; //assigning string to pointer

while (\*p != '\0')

{

if (\*p== ' ') //checking with spaces

{

word++;

chr++;

}

else

chr++;

\*p++;

}

printf("\nNumber of characters: %d", chr);

printf("\nNumber of words: %d", word+1);

}

Algorithm :

Step1:Declare all variables and initialize them.

Step2:The pointer variable p pointing to the string str.

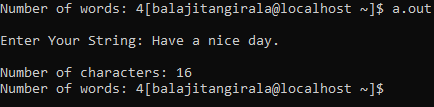
Step3:while loop iterate until the pointer value becomes null.

Step4:The ASCII value of the character at the pointer variable p is equal to white space ASCII value. Then increase the word count and character count by one.

Step5:If step 4 condition is false then character count increment by one.

Step6:Print the number of words and characters present in the string.

Output :



Program :2 a)without predefined functions:

Ans :

#include<stdio.h>

int main()

{

char str[80], search[10];

int count1 = 0, count2 = 0, i, j, flag;

printf("Enter a string:");

gets(str);

printf("Enter search substring:");

gets(search);

while (str[count1] != '\0')

count1++;

while (search[count2] != '\0')

count2++;

for (i = 0; i <= count1 - count2; i++)

{

for (j = i; j < i + count2; j++)

{

flag = 1;

if (str[j] != search[j - i])

{

flag = 0;

break;

}

}

if (flag == 1)

break;

}

if (flag == 1)

printf("string found in the main string\n");

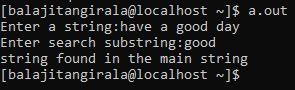
else

printf("string not found in the main string\n");

return 0;

}

Output :



Program :2 b)with using predefined functions:

Ans :

#include <stdio.h>

#include <string.h>

int main()

{

char str[20],search[6];

char \*ptr;

printf("enter a string\n");

gets(str);

printf("enter a substring\n");

gets(search);

ptr = strstr(str, search); /\*this strstr() predefined function is used to find out the first oocurrence of substring\*/

if (ptr != NULL) /\* Substring found \*/

{

printf("string found in the main string\n"); }

else /\* Substring not found \*/

{

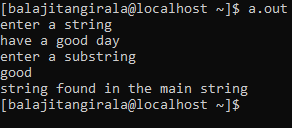
printf("String not found in the main string\n");

}

return 0;

}

Output :



Program :3 reverse of a string.

Ans :

#include<stdio.h>

int string\_length(char\*);

void reverse(char\*);

main()

{

char s[100];

printf("Enter a string\n");

gets(s);

reverse(s);

printf(“Reverse of the string is \n\"%s\".\n", s);

return 0;

}

void reverse(char \*s)

{

int length, c;

char \*begin, \*end, temp;

length = string\_length(s);

begin = s;

end = s;

for (c = 0; c < length - 1; c++)

end++;

for (c = 0; c < length/2; c++)

{

temp = \*end;

\*end = \*begin;

\*begin = temp;

begin++;

end--;

}

}

int string\_length(char \*pointer)

{

int c = 0;

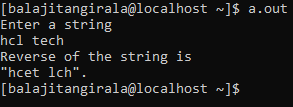
while( \*(pointer + c) != '\0' )

c++;

return c;

}

Output :



Program :4 //Frequency of a character

Ans :

#include<stdio.h>

void frequency(char \*p,char a)

{

int i,count=0;

for(i=0;p[i]!='\0';i++)

{

if(p[i]==a)

count++;

}

printf("frequency of %c is %d",a,count);

}

main()

{

char str[50]="hcl is making a training program";

char ch='a';

frequency(str,ch);

}

Output :



Program :5 //removing spaces and convert “.” to “:”

Ans :

#include<stdio.h>

int main()

{

char str[20],\*p;

int i;

printf("enter a string\n");

gets(str);

for(p=str;p[i]!='\0';i++)

{

if(p[i]==' ')

continue;

else if (p[i]=='.')

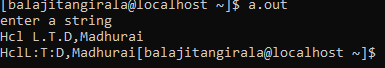
p[i]=':';

printf("%c",p[i]);

}

}

Output :



Program :6 //Anagram program of dada and adda.

Ans :

#include <stdio.h>

#include<string.h>

int check\_anagram(char a[], char b[]);

int main()

{

  char a[100], b[100];

  printf("Enter a string\n");

  gets(a);

  printf("Enter a string\n");

  gets(b);

  if (check\_anagram(a, b) == 1)

    printf("The strings are anagrams.\n");

  else

    printf("The strings aren't anagrams.\n");

  return 0;

}

int check\_anagram(char a[], char b[])

{

  int first[26] = {0}, second[26] = {0}, c=0;

 while(a[c]!='\0')

{

        first[a[c]-'a']++;

        c++;

}

        c=0;

while(b[c]!='\0')

{

        second[b[c]-'a']++;

        c++;

}

for(c=0;c<26;c++)

{

        if(first[c]!=second[c])

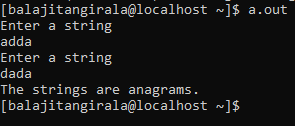
        return 0;

}

return 1;

}

Output :



Program :7// sequence of university.

Ans :

#include<stdio.h>

#include<string.h>

int main()

{

char a[10]="UNIVERSITY";

int i,j,n;

n=strlen(a);

for(i=1;i<n/2;i++,printf("\n"))

{

for(j=0;j<(2\*i);j++)

printf("%c ",a[j]);

}

for(i=n/2;i>=1;i--,printf("\n"))

{

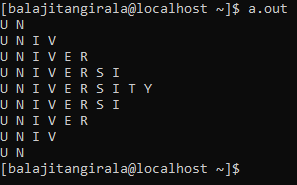
for(j=0;j<(2\*i);j++)

printf("%c ",a[j]);

}

}

Output :



Program :8 //Write a program to copy one string to another string with and without using stringhandling function

Ans :

#include<stdio.h>

#include<string.h>

main()

{

char str[20],copy[20],\*p,\*y;

printf("enter a string:");

gets(str);

p=str;

y=copy;

strcpy(y,p);

printf("copied string is: %s",y);

}

Output :

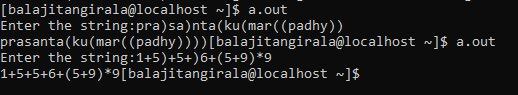


Program :9 //Write a c program to balance a string after removing extra brackets.

Ans :

#include <stdio.h>  
#include<string.h>  
void balancedString(char \*str)  
{  
    int count = 0, i;  
    int n = strlen(str);  
   
    for (i = 0; i < n; i++) {  
   
        if (str[i] == '(') {  
            printf("%c",str[i]);  
            count++;  
        }  
   
        else if (str[i] == ')' && count != 0) {  
            printf("%c",str[i]);   
            count--;  
        }  
        
        else if (str[i] != ')')  
            printf("%c",str[i]);  
    }  
   
    if (count != 0)  
     
        for (i = 0; i < count; i++)  
           printf(")");  
}  
   
int main()  
{  
   
    char str[50];  
 printf("Enter the string:");  
 gets(str);  
     balancedString(str);  
   
    return 0;  
}

Output :



Program :10 // Write a program to compare two input strings, if the input strings are identical print "Both the strings are equal" otherwise print "Input strings are not identical".

Ans :

#include<stdio.h>

int stringCompare(char[],char[]);

int main()

{

char str1[100],str2[100];

int compare;

printf("Enter first string: \n");

scanf("%s",str1);

printf("Enter second string: \n");

scanf("%s",str2);

compare = stringCompare(str1,str2);

if(compare == 1)

printf("Input strings are identical\n");

else

printf("Input strings are not identical\n");

return 0;

}

int stringCompare(char str1[],char str2[])

{

int i=0,flag=0;

while(str1[i]!='\0' && str2[i]!='\0')

{

if(str1[i]!=str2[i])

{

flag=1;

break;

}

i++;

}

if (flag==0 && str1[i]=='\0' && str2[i]=='\0')

return 1;

else

return 0;

}

Output :

