

|  |
| --- |
| C.P.N.M. LAB REPORT |
|  |
| ASSIGNMNET 5  BCSE FIRST YEAR FIRST SEMESTER  Authored by: SOHAM CHOWDHURY |



**CPNM LAB ASSIGNMENT REPORT**

BCSE FIRST YEAR FIRST SEMESTER 2021-2022

NAME-SOHAM CHOWDHURY

DEPARTMENT-COMPUTER SCIENCE AND ENGINEERING

SECTION-A3.

ROLL NO-002110501145.

# ASSIGNMENT 5

1. Write recursive functions for following tasks. a. Binary equivalent of a number. b. Sum of individual digits of a number passed as argument.

For converting decimal to binary I stored the remainder in n%2,and sent n/2 for next recursion till n becomes 1.

For storing the sum of digit I extracted the digits by n%10 and made another function call n/10 tilln becomes 0.

Program: #include<stdio.h>

#include<math.h>

int dec\_to\_bin(int n)

{

    if(n==1)

    return 1;

    else

    return(n%2+dec\_to\_bin(n/2)\*10);

}

int sum\_of\_digits(int n)

{

    if(n==0)

    return 0;

    else

    return (n%10+sum\_of\_digits(n/10));

}

int main ()

{

    int n;

    char choice;

    printf("enter the number=");

    scanf("%d",&n);

    fflush(stdin);

    printf("enter the choice=");

    scanf("%c",&choice);

    fflush(stdin);

    switch (choice)

    {

    case 'a':

    printf("the binary equivalent of the number is=%d",dec\_to\_bin(n));

    break;

    case 'b':

    printf("the sum of digits is %d",sum\_of\_digits(n));

    break;

    default:

    printf("invalid choice");

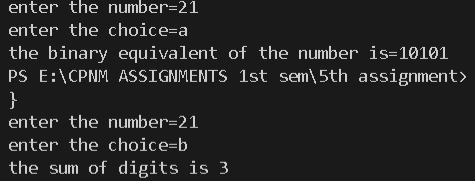
    break;

    }

    return 0;

}

Output:



1. Write a C program using functions which accepts a string from the user and performs the following tasks. a. Counts the number of characters in the string without using string library functions. b. Prints the reverse of the string without using string library functions.

Here I counted the number of characters by comparing each character of the character array with the space and If it is not a space then it has to be a character, for reverse I printed the string in reverse

Program:

#include<stdio.h>

#include<math.h>

int main()

{

    char str[100],choice;

    printf("enter the string:\n");

    gets(str);

    int i=0,c=0,j=0;

    printf("enter the choice=");

    scanf("%c",&choice);

    switch(choice)

    {

        case 'a':

        while(str[i]!='\0')

        {

            if(str[i]!=' ')

            c++;

            i++;

        }

        printf("the total no of characters in the given string is=%d",c);

        break;

        case 'b':

        while(1)

        {

            if(str[j]=='\0')

            {

                break;

            }

            else

            j++;

        }

        printf("the reversed string is:");

        while(j>=0)

        {

            printf("%c",str[j]);

            j--;

        }

        break;

        default:

        printf("invalid choice");

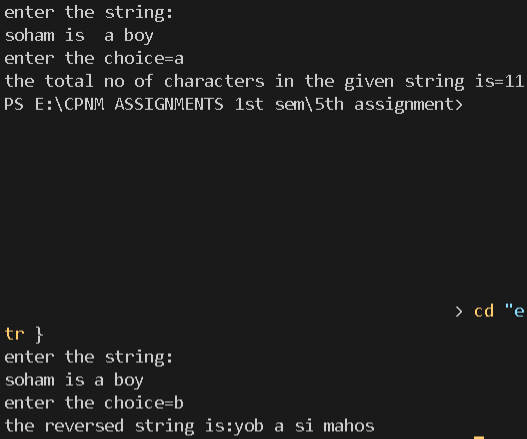
        break;

    }

    return 0;

}

Output:



1. Write a C program which accepts a full name from the user prints the initials. Eg. SRT for Sachin Ramesh Tendulkar.

Took the string as input from the user then added a space at the beginning of the string and if a space occurs before any character just print that.

Program:

#include<stdio.h>

#include<string.h>

int main()

{

    char name[30];

    printf("enter the string:-\n");

    gets(name);

    int l=strlen(name);

    for(int i=l-1;i>=0;i--)

    {

        name[i+1]=name[i];

    }

    name[0]=' ';

    printf("the initials are as follows:");

    for(int i=0;name[i]!='\0';i++)

    {

        if((name[i]==' ')&&(name[i+1]!=' '))

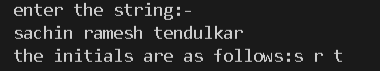
        printf("%c ",name[i+1]);

    }

    return 0;

}

Output:



1. Write a program to count the number of occurrences of any two vowels in succession in a line of text.

Took consecutive elements of the character array using loops and if ith and i+1th elements were vowels then I increased the counter by 1.

Program:

#include<stdio.h>

#include<string.h>

int main()

{

    char str[30];

    int c=0;

    printf("enter the string=");

    gets(str);

    for(int i=0;str[i]!='\0';i++)

    {

        if(((str[i]=='a')||(str[i]=='A')||(str[i]=='e')||(str[i]=='E')||(str[i]=='i')||(str[i]=='I')||(str[i]=='o')||(str[i]=='O')||(str[i]=='u')||(str[i]=='U'))&&((str[i+1]=='a')||(str[i+1]=='A')||(str[i+1]=='e')||(str[i+1]=='E')||(str[i+1]=='i')||(str[i+1]=='I')||(str[i+1]=='o')||(str[i+1]=='O')||(str[i+1]=='u')||(str[i+1]=='U')))

        c++;

    }

    printf("the number of occurences of vowels in succession in a line is=%d",c);

    return 0;

}

Output:



1. Write a program that converts (Do not use any string library function): a. A string like “123” to integer 123. b. An integer like 123 to string “123”.

For string to integer I used the atoi() function that is in the standard library of c. And for converting a user defined integer to string I used the concept of ASCII values.

Program:

#include<stdio.h>

#include<stdlib.h>

int main()

{

    char str[10];

    int num,c=0,sum=0;

    char choice,temp;

    printf("enter the choice=");

    scanf("%c",&choice);

    fflush(stdin);

    switch(choice)

    {

        case 'a':

        printf("enter the string=");

        gets(str);

        num=atoi(str);

        printf("%d",num);

        break;

        case 'b':

        printf("enter the number=");

        scanf("%d",&num);

        for(int i=num;i>0;i/=10)

        {

            sum=sum\*10+i%10;

        }

        for(int i=sum;i>0;i/=10)

        {

            str[c++]=(char)(i%10+48);

        }

        printf("the number in string is=");

        for(int i=0;i<c;i++)

        printf("%c",str[i]);

        break;

        default:

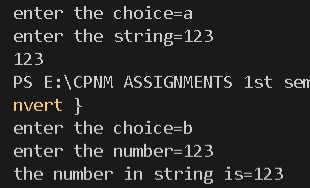
        printf("invalid choice");

    }

    return 0;

}

Output:



1. Write a C program which accepts a string from the user and performs the following tasks. (Do not use any string library function. ) a. Check whether it is palindrome or not. [Example of a palindrome string: "abcba", "abba"] b. Counts the number of characters and words in it.

Traversed the string from beginning and end simultaeneously till the middle of the string and if each and every character matches then the string is palindrome.

Program:

#include<stdio.h>

int main()

{

    char str1[30];

    printf("enter the string=");

    gets(str1);

    char choice;

    int f=0,c=0,w=0,k;

    printf("enter the choice=");

    scanf("%c",&choice);

    switch(choice)

    {

        case 'a':

        for(int i=0;;i++)

        {

            if(str1[i]=='\0')

            {

                k=i;

                break;

            }

        }

        for(int i=0;i<k/2;i++)

        {

            if(str1[i]!=str1[k-i-1])

            {

                f=1;

                break;

            }

        }

        if(f==1)

        printf("the string is not pallindrome");

        else

        printf("the string is pallindrome");

        break;

        case 'b':

        for(int i=0;;i++)

        {

            if(str1[i]=='\0')

            {

                k=i;

                break;

            }

        }

        for(int i=k;i>=0;i--)

        str1[k+1]=str1[k];

        str1[0]=' ';

        for(int i=0;i<k+1;i++)

        {

            if((str1[i]==' ')&&(str1[i+1]!=' '))

            w++;

            if(str1[i]!=' ')

            c++;

        }

        printf("the number of words are %d and characters are %d in the given string",w,c);

        break;

        default:

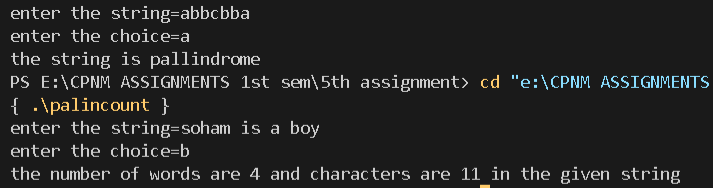
        printf("invalid choice");

    }

    return 0;

}

Output:



1. Write a program in C to store n numbers in an array and print the elements using pointers. Also compute the sum of all elements of that array using pointers.

The array pointer points towards the first element of the array and with increase in value of pointer by 1 we can traverse the entire array and summed up individual elements .

Program:

#include<stdio.h>

int main()

{

    int arr[30];

    int \*ptr;

    int l,sum=0;

    printf("enter the length of the array=");

    scanf("%d",&l);

    printf("enter the elements of the array:\n");

    for(int i=0;i<l;i++)

    scanf("%d",&arr[i]);

    ptr=&arr[0];//or ptr=&arr but error showing;

    for(int i=0;i<l;i++)

    {

        sum=sum+\*(ptr);

        ptr=ptr+1;

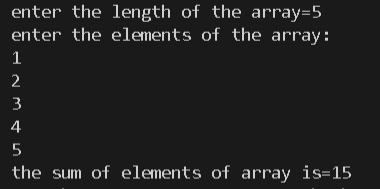
    }

    printf("the sum of elements of array is=%d",sum);

    return 0;

}

Output:



8. Write a C function which accepts a string str1 and returns a new string str2 which is str1 with each word reversed. Do not use any string library function.

For reversing words I just reversed each words by swapping 1st and last element and 2nd and 2nd last element and so on till the middle of the word and traversed every word.

For reversing order of the words I took the words from the end of the array and put them to the otter end of another array.

Program:

#include<stdio.h>

#include<math.h>

char \* rev\_wrd1(char str[])

{

    static char res[50];

    char word[15],temp;

    int c=0;

    int l=0,d;

    for(int i=0;;i++)

    {

        if(str[i]=='\0')

        {

            d=i;

            break;

        }

    }

    str[d]=' ';

    str[d+1]='\0';

    for(int i=0;str[i]!='\0';i++)

    {

        if(str[i]==' ')

        {

            word[c]='\0';

            for(int j=0;j<c/2;j++)

            {

                temp=word[j];

                word[j]=word[c-1-j];

                word[c-1-j]=temp;

            }

            for(int k=0;word[k]!='\0';k++)

            {

                res[l++]=word[k];

            }

            res[l++]=' ';

            c=0;

        }

        else

        {

            word[c++]=str[i];

        }

    }

    return (res);

}

char \* rev\_wrd2(char str[])

{

    static char res[50];

    int d,c=0;

    char word[15];

    for(int i=0;;i++)

    {

        if(str[i]=='\0')

        {

            d=i;

            break;

        }

    }

    str[d]=' ';

    str[d+1]='\0';

    res[d--]='\0';

    for(int j=0;str[j]!='\0';j++)

    {

        if(str[j]==' ')

        {

            word[c]='\0';

            int l=c-1;

            for(int f=l;f>=0;f--)

            {

                if(d>=0)

                {

                    res[d--]=word[f];

                }

            }

            res[d--]=' ';

            c=0;

        }

        else

        {

            word[c++]=str[j];

        }

    }

    return (res);

}

int main()

{

    char str[50];

    char \* ptr;

    printf("enter the string=");

    gets(str);

    int choice;

    printf("enter the choice=");

    scanf("%d",&choice);

    if(choice==1)

    ptr=rev\_wrd1(str);

    if (choice==2)

    ptr=rev\_wrd2(str);

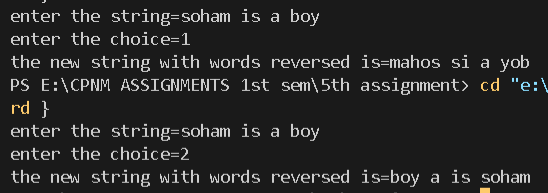
    printf("the new string with words reversed is=");

    puts(ptr);

    return 0;

}

Output:



1. Write a function squeeze(s,c) which removes all occurrences of the character c from the strings.

As soon as we found that the character c is in a character array then just delete that element from that string.

Program:

#include<stdio.h>

#include<string.h>

char\* squeeze(char s[30],char c)

{

    for(int i=0;s[i]!='\0';i++)

    {

        if(s[i]==c)

        {

            for(int j=i;s[j]!='\0';j++)

            {

                s[j]=s[j+1];

            }

            i--;

        }

    }

    return s;

}

int main()

{

    char str[30];

    char \*ptr;

    char c;

    printf("enter the string=");

    gets(str);

    printf("enter the character to be squeezed=");

    scanf("%c",&c);

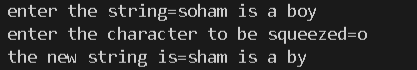
    ptr=squeeze(str,c);

    printf("the new string is=");

    puts(ptr);

    return 0;

}

Output: 

10.Write the function strend(s,t), which returns 1 if the string t occurs at the end of the string s, and zero otherwise.

Take both strings from the user then check each and every element from the end of the string 1 till the end of length of second string.

Program:

#include<stdio.h>

#include<string.h>

int stend(char s[50],char t[30])

{

    int l=strlen(s);

    int k=strlen(t);

    int c=0;

    for(int i=l-1,j=k-1;i>=0,j>=0;i--,j--)

    {

        if(s[i]!=t[j])

        {

            c=1;

            break;

        }

    }

    if(c==0)

    return 1;

    else

    return 0;

}

int main()

{

    char s[50];

    char t[30];

    printf("enter the string=");

    gets(s);

    fflush(stdin);

    printf("enter the string to be checked=");

    gets(t);

    if(stend(s,t)==1)

    printf("the string %s occurs at the end of string %s",t,s);

    else

    printf("the string %s does not occur at the end of string %s",t,s);

    return 0;

}

Output:

