**KARNATAK LAW SOCIETY’S**

**GOGTE INSTITUTE OF TECHNOLOGY**

**UDYAMBAG, BELAGAVI – 590008**

**(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)**

**(Approved By AICTE, New Delhi)**

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

**DATA STRUCTURE IN C**

RACHANA KAMPLI 2GI18IS032

LAXMI NYAMAGOUD 2GI18IS020

HEMANTH I T 2GI18IS015

ROHAN K

Under the Guidance of:  **Prof N.V Karekar**

Index

# 1.Title

# 2.Problem statement

# 3. Objectives of Defined Problem Statement

# 4.Methedology(pseudocode)

# 5.Implementation details

# 6.Working of final solution

# 7.Conclusion

1. **Title**: - Sorting of Names
2. **Problem statement: -**

A C-program to arrange the names alphabetically using linked list

1. **Objectives of Defined Problem Statement: -**

To Sort the Names Alphabetically using Linked list

1. **Methodology (pseudo code): -**

Step 1: Declaration of functions to create, display and sort the linked list.

Step 2: Declaring a structure to create a node.

Step 3: Enter the names to be sorted.

Step 4: Display the contents of Linked list before sorting using **disp ()** function.

Step 5: Call **Sort ()** function to sort the names.

Step 6: Display the contents of Linked list after sorting using **disp ()** function

**5.Implementation details: -**

/\*This program takes a set of strings as input and sorts them alphabetically before displaying \*/

#include<stdio.h>

#include<conio.h>

#include <stdlib.h>

#include <string.h>

void create(void);

void disp(void);

void sort(void);

struct name

{

char info [20];

struct name \*link;

} \*ptr,\*start,\*node;

typedef struct name list;

int main()

{

// clrscr();

printf("Program to enter the names and sort them using linked list.\n");

create (); //fun to enter the names

printf (" The contents of the list were \n: ");

disp (); //fun to display the data

sort (); //fun to sort the data

printf ("\nAfter sorting :\n");

printf ("The contents of the list are : \n");

disp ();

getch();

}

void create(void)

{

char ch='y';

node=(list \*)malloc(sizeof(list));

start=node;

while(ch=='y')

{

printf("Enter the name :\n");

gets(node->info);

ptr=node;

node=(list \*)malloc(sizeof(list));

ptr->link=node;

printf("Want to continue?(y/n)");

fflush(stdin);

scanf("%c",&ch);

if(ch=='n')

break;

ch='y';

}

ptr->link=NULL;

}

//-------------------------------------------

void disp(void)

{

ptr=start;

while(ptr!=NULL)

{

printf("\t%s",ptr->info);

ptr=ptr->link;

}

}

//-------------------------------------------

void sort(void)

{

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

int i;

char temp[20],s1[20],s2[20];

list \*ptr2;

for(ptr=start;ptr!=NULL;ptr=ptr->link)

{

for(ptr2=ptr->link;ptr2!=NULL;ptr2=ptr2->link)

{

strcpy(s1,ptr->info);

strcpy(s2,ptr2->info);

i=comp(s1,s2);

if(i==1)

{

strcpy(temp,ptr->info);

strcpy(ptr->info,ptr2->info);

strcpy(ptr2->info,temp);

}

}

}

}

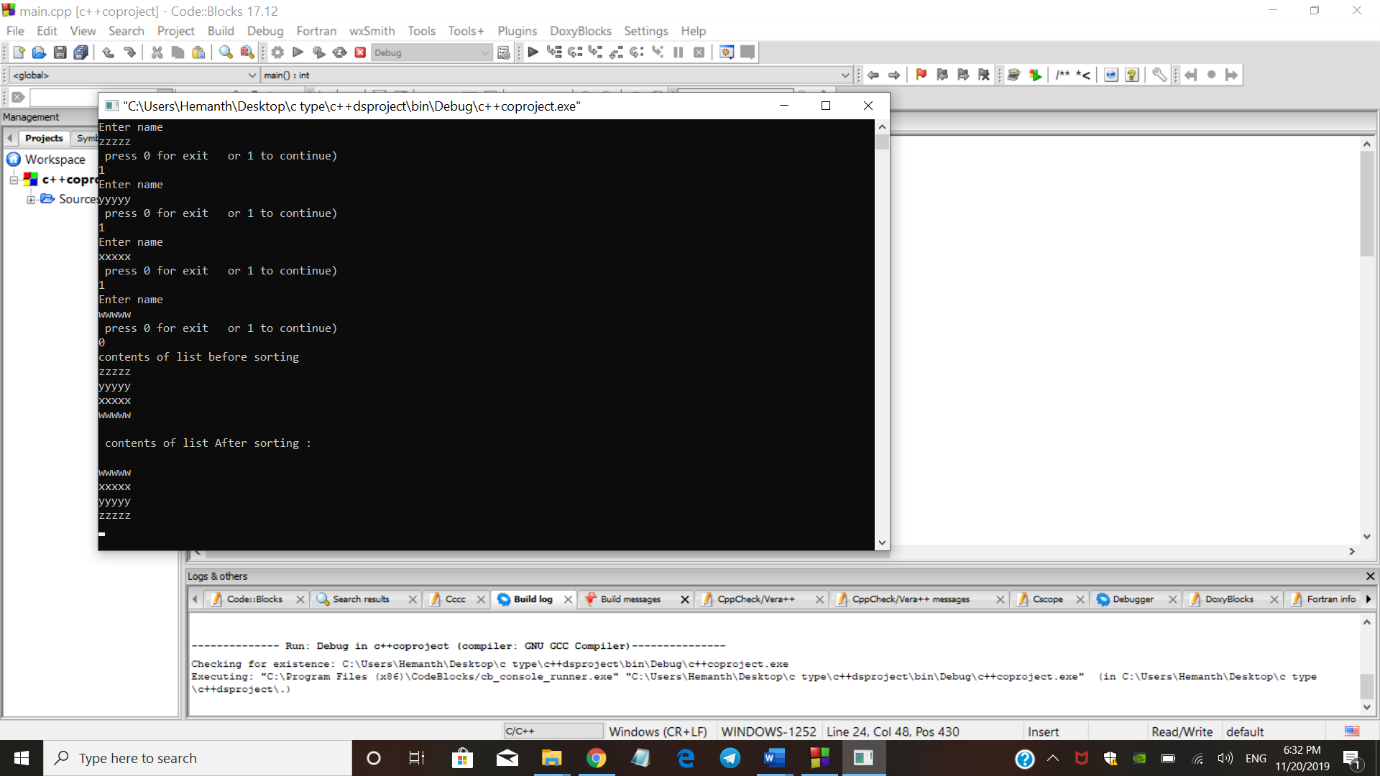
//-------------------------------------------

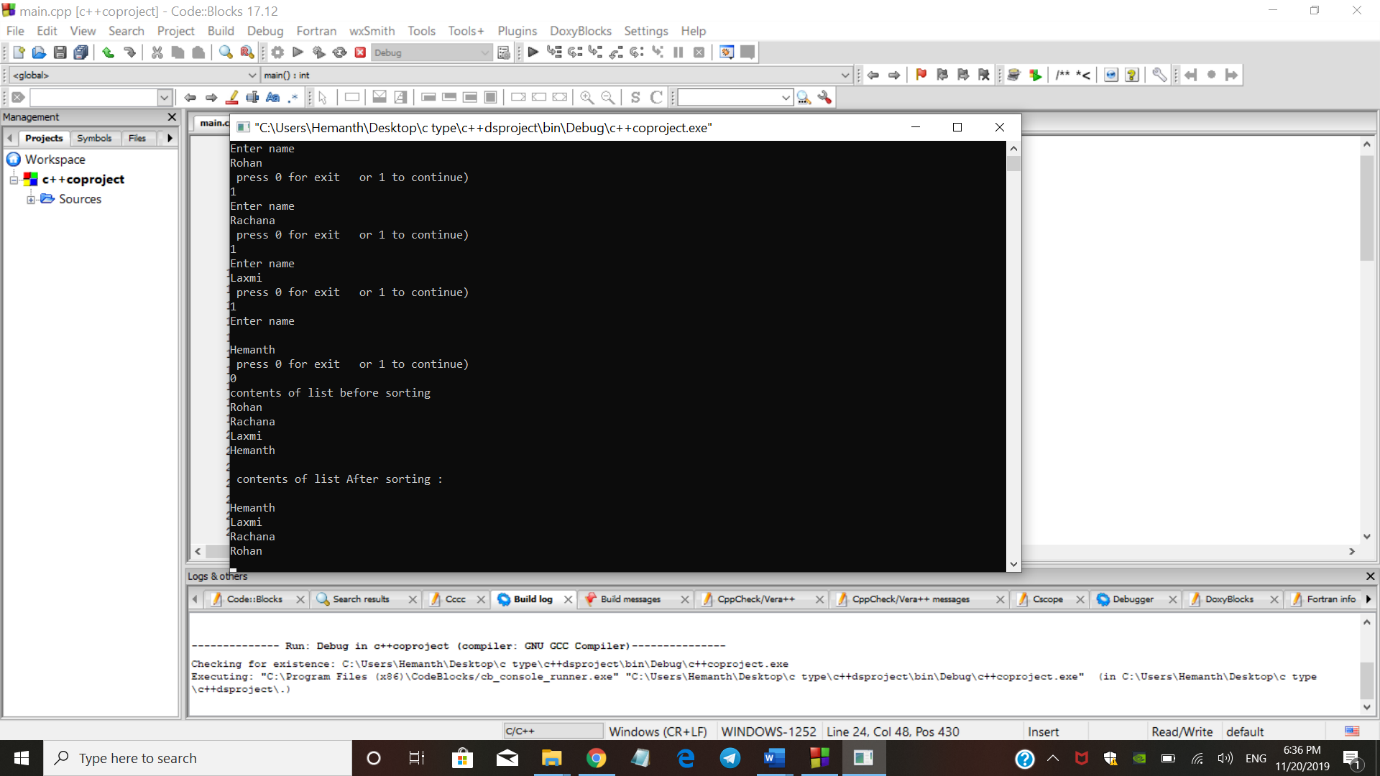
int comp (char s1[], char s2[]) {

return (strcmp (s1, s2));

}

**6.Working model of the final solution**: -

****

****

**Conclusion:** we are successfully implemented singly linked list for sorting of string. The above source code has given a brief idea for sorting the strings for an given pattern(alphabetically)