

SANJAY GHODAWAT UNIVERSITY

# Kolhapur

Established under section 2(f) of UGC Act 1956

Sanjay Ghodawat University Act XL of 2017 of Govt. Maharashtra Approved by PCI, COA & AICTE

**PROJECT SYNOPSIS**

**ON**

**“PHONE DIRECTORY”**

A synopsis submitted in partial fulfillment of the requirements for the

School of Computer Science and Engineering By

**MOHAMMAD SAYEED KAZI 22SC114281005**

**NIHAL NAIK 22SC114281016**

**RASHI INGALE 22SC114281025**

**SURAJ KARANDE 22SC114281026**

Program: Bachelor of Computer Science and Engineering Class: FY BTech

Under Supervision of

## Mrs. Veena Mali

Assistant Professor

School of Computer Science and Engineering

A.Y 2022-23



SANJAY GHODAWAT UNIVERSITY

# Kolhapur

Established under section 2(f) of UGC Act 1956

Sanjay Ghodawat University Act XL of 2017 of Govt. Maharashtra Approved by PCI, COA & AICTE

**School of Computer Science and Engineering**

**CERTIFICATE**

**PROJECT SYNOPSIS**

**ON**

**“PHONE DIRECTORY”**

Submitted By

**MOHAMMAD SAYEED KAZI 22SC114281005**

**NIHAL NAIK 22SC114281016**

**RASHI INGALE 22SC114281025**

**SURAJ KARANDE 22SC114281026**

Program: Master of Computer Science and Engineering Class: FY BTech

Is work done by him/her and submitted during academic year 2022-23,

in partial fulfillment of the Project Synopsis.

**Sanjay Ghodawat University, Kolhapur**

**Mrs. Veena Mali Dr. B. Suresh Kumar**

**Course coordinator HOS, SoCSE**

# INTRODUCTION

**“The best way to learn a new programming language is by writing programs in it"** - **DENNIS RITCHIE**

## The Phone Directory project based on DSA using C language aims to develop a program that enables users to store, search, update, and delete contact information such as phone numbers, names, and addresses. The project requires the use of Data Structures and Algorithms (DSA) concepts such as arrays, linked lists, searching, and sorting algorithm.

## The project is divided into several modules, each implementing a specific operation. The modules include adding a new contact, searching for a contact, updating a contact, deleting a contact, and displaying all contacts. The project uses an array of structures to store the contact information and a linked list to implement the search and delete operations efficiently.

## The program allows users to enter and store contact information such as name, phone number, and address. Users can search for a contact using either linear or binary search algorithm, update the contact information, delete a contact, and display all the contacts in the phone directory.

The Phone Directory project based on DSA using C language provides an efficient and user-friendly way to manage contacts. It helps users to organize their contacts and quickly retrieve the desired information.

# OBJECTIVES

The objectives of our project, ‘Phone Directory’ are as follows:

1. The objective of a phone directory program using C language is to allow users to store, manage, and retrieve contact information such as names, phone numbers, and addresses.
2. The program should be able to perform basic operations such as adding a new contact, deleting an existing contact, updating an existing contact, and searching for a contact by name or phone number.
3. The phone directory program can be implemented using data structures such as arrays, linked lists, or hash tables to store the contact information. The program should also provide a user-friendly interface that allows users to interact with the directory easily.

Overall, the main objective of a phone directory program is to provide a convenient way to store and manage contact information, allowing users to easily keep track of their contacts and quickly retrieve their information when needed.

# EXISTING SYSTEM

A traditional phone directory system typically includes the following components:

**Printed directories**: Phone directories used to be printed in books or booklets, organized alphabetically by the name of individuals, businesses, and organizations. They included contact information such as phone numbers, addresses, and occasionally brief descriptions.

**Phone company directories**: Phone company directories were also commonly used, especially for landline phone numbers. These directories were provided by phone companies and included the phone numbers and names of their customers.

A traditional phone directory system relied heavily on manual data entry and maintenance, which made them susceptible to errors and inaccuracies. They also required regular updates and distribution to ensure that the information provided was up-to-date and reliable.

# LIMITATIONS

There are several limitations of a phone directory implemented using C language, including:

* **Scalability**: C language is not ideal for building large-scale applications, and as the number of contacts in a phone directory grows, it can become difficult to manage and maintain the code.
* **Search efficiency**: Searching through a phone directory can become time-consuming if the directory is not sorted properly. C language does not have built-in support for data structures like hash tables or binary search trees, which can help improve search efficiency.
* **User interface**: C language is not the best choice for building user interfaces, and a phone directory implemented using C may have a limited or basic interface.
* **Limited data validation**: C language does not have built-in support for data validation, which can make it challenging to ensure that the data entered into the phone directory is valid and consistent.
* **Limited functionality**: A phone directory implemented using C language may lack some of the more advanced features found in modern phone directoryapplications, such as integration with social media, the ability to import/export contacts, and voice recognition.
* **Data security:** C language does not have built-in support for encryption and other security features, which can make it challenging to protect sensitive data in a phone directory**.**

# PROPOSED SYSTEM

## A phone directory system is an application that allows users to store and retrieve contact information. The system typically consists of a database or array that stores contact information such as name, phone number, and possibly other details such as email address, address, and notes.

## In the C language, the phone directory system can be implemented using data structures such as arrays and structures. The array can be used to store a list of contacts, while the structure can be used to define the contact information fields such as name and phone number.

## The phone directory system can provide various functionalities such as adding a new contact, searching for a contact by name, displaying all contacts, updating contact information, and deleting a contact. The system can be implemented using functions that perform these operations on the contact list.

# TECHNOLOGY USED

## SOFTWARE REQUIREMENTS

* + Turbo C++ / Dev C++ / Code: Blocks / VS code

## HARDWARE REQUIREMENTS

* + 4 GB RAM
  + 500 GB HDD
  + i3 processor

# METHODOLOGY

**Algorithm:**

**STEP** **1**: Start.

**STEP** **2**: Display the options on the screen.

**STEP** **3**: Print "WELCOME TO PHONE Directory " and "Menu" on the screen

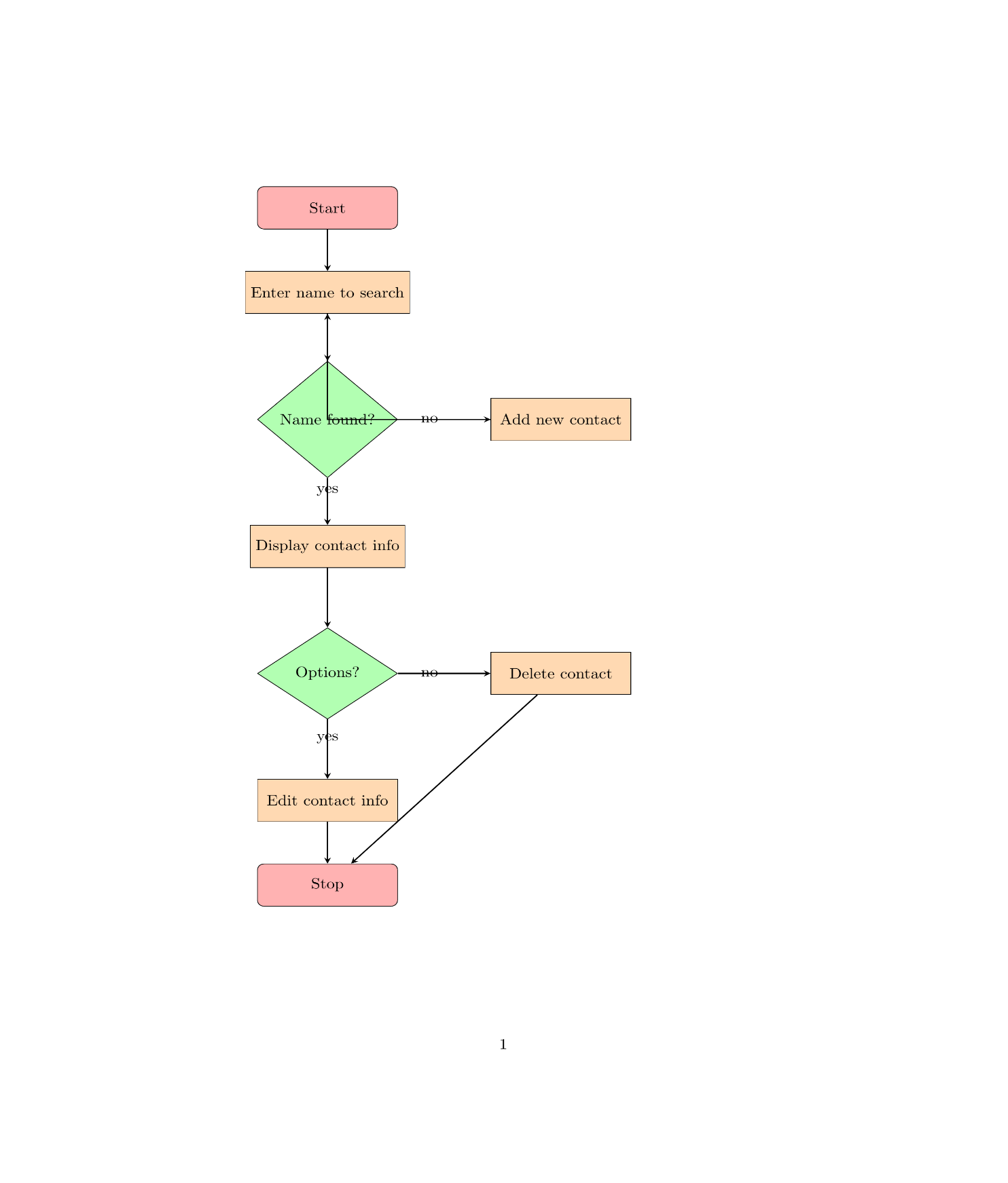
**STEP** **4**: Add contact/entry

**STEP** **5**: Search contact/entry

**STEP** **6**: Delete contact/ entry

**STEP** **7**: Display contact

**STEP** **8**: Close phone book

**Flowchart:** 

Name found?

Name found?

**EXPECTED OUTCOMES**

Expected outcomes of this program are, it will show a title of Phone Directory with five options. These five options include:

**1.Add entry**

**2.Search for entry**

**3. Delete entry**

**4.Print directory**

**5.Exit**

It will also display a message to enter choice, as per the given options you have to choose the options and it will give you the further information. If 1st option is choose then you have to enter the name and number of particular person. For 2nd option you will be able to search a particular entry with phone number. using 3rd option you can easily delete the any entry which makes the directory easy to manage using 4th option you will be able to print the phone directory and using 5th option you can directly exit the output of this program

# REFERENCES

* Websites:

<https://stackoverflow.com/>

<https://www.w3schools.com/r/r_graph_scatterplot.asp>

<https://www.geeksforgeeks.org/>

* + - Publication References Books:

1. C: The Complete Reference.
2. Algorithms in a Nutshell
3. Data Structure and Program Design Using C