Contents

[LIST OF FIGURES:- 3](#_Toc50924474)

[LIST OF TABLES: - 3](#_Toc50924475)

[1.) INTRODUCTION: - 4](#_Toc50924476)

[ASSUMPTIONS: - 5](#_Toc50924477)

[2.) OBJECTIVE: - 5](#_Toc50924478)

[3.) DESIGN OF THE PROGRAM: - 6](#_Toc50924479)

[FLOW CHART: - 6](#_Toc50924480)

[PSEUDO CODE: - 7](#_Toc50924481)

[4) C PROGRAMMING CONCEPT’s. 8](#_Toc50924482)

[Comments 8](#_Toc50924483)

[Header File 9](#_Toc50924484)

[Data Types 9](#_Toc50924485)

[Integer type 9](#_Toc50924486)

[Floating point type 10](#_Toc50924487)

[Character type 11](#_Toc50924488)

[Arrays 11](#_Toc50924489)

[Structure 11](#_Toc50924490)

[File Handling 12](#_Toc50924491)

[Function 14](#_Toc50924492)

[5)RESULTS. 16](#_Toc50924493)

[6.) Conclusion. 18](#_Toc50924494)

[REFRENCES: - 19](#_Toc50924495)

# LIST OF FIGURES:-

Figure 1:- Flow chart………………………………………………………………………………6

Figure 2:- Structure syntax……………………………………………………………………….12

Figure 3:- Output screen………………………………………………………………………….16

Figure 4:- Files before execution…………………………………………………………………16

Figure 5:- Files after execution…………………………………………………………………...17

Figure 6:- File content……………………………………………………………………………17

# LIST OF TABLES: -

[Table 1:-Integer type. 11](#_Toc50924194)

[Table 2:-Float and double 11](#_Toc50924195)

[Table 3:-Char type. 12](#_Toc50924196)

[Table 4:-File handling 14](#_Toc50924197)

# INTRODUCTION: -

In the beginning was Charles Babbage and his Analytical Engine, a machine he built in 1822 that could be programmed to carry out different computations.  Move forward more than 100 years, where the U.S. government in 1942 used concepts from Babbage’s engine to create the ENIAC, the first modern computer. The programming for Babbage was so hard that It needs to be manually done by replacing stacks of gears whereas in the case of ENIAC it had to be rewired by the hand.

C is a general-purpose high-level language that was originally developed by Dennis Ritchie for the Unix operating system. It was first implemented on the Digital Equipment Corporation PDP-11 computer in 1972. The Unix operating system and virtually all Unix applications are written in the C language. C is also termed as mother of all new languages because most of the new programming languages are made from C.

The requirement of C was due to rise in the system development work, C compiler runs nearly as fast as code written in assembly language that’s why C was adopted as a system development language .C is used in most of the field of computer science such as operating systems, Languages Compiler, Assemblers, Text Editors, Print spoolers, Network drivers, Modern Programs, Data bases, Languages Interpreters etc.

C program is widely used in the software related to management system. Car service management system software is vital for Car servicing workshop where keeping track of the respective customer is necessary. Since car servicing workshop had different type of service at different prices, car service system can easily provide a menu to the user for the selection of the service along with the type of service they need, and generating an automated invoice will also reduce time.

## ASSUMPTIONS: -

There were few assumptions made before developing the program: -

* The invoice number generated in the sequential order will start from 1.
* The menu will ask the customer to input service needed and type of service they need.
* The customer information will be recorded by the user which will be used for the invoice generation.
* Once the information will be gathered an invoice will be shown on the screen, an individual invoice will be saved as a txt file.

# OBJECTIVE: -

The objective of the assignment was to Develop a Fast track car service management system as per requirements stated below. The system was needed to develop using C programming language.

1. Display a menu of services provided by the shop with other details as shown in the above table.
2. To Record the customer and service information.
3. To autogenerate an invoice for each individual customer.

# DESIGN OF THE PROGRAM: -

## FLOW CHART: -

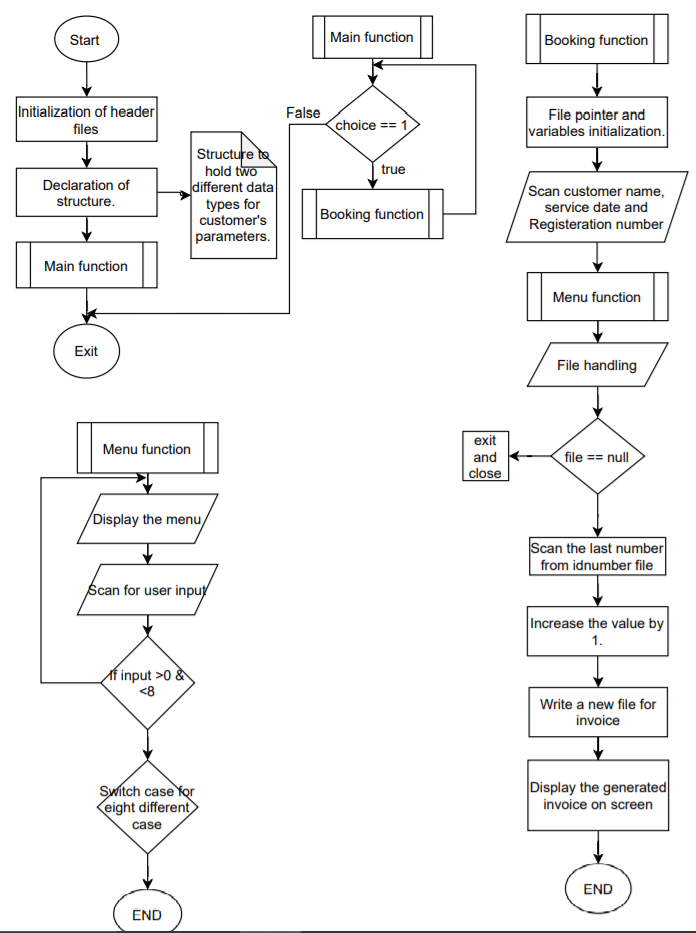


Figure 1:-Flow Chart.

## PSEUDO CODE: -

Pseudo code for the booking function is shown below.

PSEU DO CODE for new customer function ()

* START.
* Scan customer name.
* Scan service date.
* Scan registration number.
* Menu().
* Open file in read mode fopen("idnumber.txt","r+").
* if fp==0

print error.

* Fscanf value from the file.
* Close file.
* Increase value variable by one.
* Assign the value variable to record structure id member.
* Open file in write mode fopen("idnumber.txt","w").
* Fprintf value.
* Close file
* Get the customer name from the record structure.
* Strcpy customer name variable to new variable named filename.
* Strcpy filename content with string “.txt”.
* Open file in write mode fopen(filename, "w").
* Fprintf all the parameters with the unique id.
* Close file.
* END.

# 4) C PROGRAMMING CONCEPT’s.

## Comments

“In the C Programming Language, comments can be placed in the source code that are not executed as part of the program.

Comments provide clarity to the C source code by facilitating better understanding of what the code was intended to accomplish and greatly helping in debugging the code. Comments are especially important in large projects that contain hundreds or thousands of lines of code or in projects where many contributors are working on the source code.

A comment starts with a slash asterisk /\* and ends with a asterisk slash \*/ and can be placed anywhere in the program. Comments can span several lines within the C program. Comments are typically added directly above the related C source code.”

Adding source code comments to C source code is a highly recommended practice.

The syntax for a **comment** is:

/\* *comment goes here* \*/

//below are the header files that were used in the program

#include<stdio.h>

#include<ctype.h>

The above line shows the single line comment, used above the header file to explain the header files that were used in the program.

/\*

below function performs the file handling and generates a seperate invoice for each user

\*/

void booking()

{

FILE\* fp; char ch; int value;

The above line shows the multi-line comment that was used to describe the booking funcntion.

## Header File

* Header files contain function declaration and macro definition for C in-built library functions.
* All standard C library functions are declared in header files which are saved as file\_name.h.
* These header files are included in the C program using “#include <file\_name.h>” command to make use of the functions that are declared in the header files.
* When header files are included in the C program using “#include <filename.h>” command, all the C code of the header files are included in C program. Then, this C program is compiled by the compiler and then executed.

#include<stdio.h>

#include<ctype.h>

#include<conio.h>

#include<stdlib.h>

#include<Windows.h>

#include<string.h>

“Above lines shows some of the header files used in the program. Stdio.h known as standard input and output library contains all the functions required for the input and output operations. Conio.h is mainly used to access functions like clrscr() and getch () which is used widely all over the program. Stdlib.h is a general-purpose library often used in most of the programs some of the functions which were used in the programs are atoi() and itoa() which are string to integer and integer to string conversions.”

## Data Types

C language supports 2 different type of data types:

1. **Primary data types**:

The main data types in C are namely integer (int), floating point(float), character(char) and void.

1. **Derived data types**:

They are nothing but primary datatypes but a little altered or grouped together like **array**, **structure**, **union,** and **pointer**.

Integer type

Table :-Integer type.

|  |  |  |
| --- | --- | --- |
| **Type** | **Size(bytes)** | **Range** |
| int or signed int | 2 | -32,768 to 32767 |
| unsigned int | 2 | 0 to 65535 |
| short int or signed short int | 1 | -128 to 127 |
| unsigned short int | 1 | 0 to 255 |
| long int or signed long int | 4 | -2,147,483,648 to 2,147,483,647 |
| unsigned long int | 4 | 0 to 4,294,967,295 |

Floating point type

Floating types store real numbers.

Table :-Float and double

|  |  |  |
| --- | --- | --- |
| **Type** | **Size(bytes)** | **Range** |
| Float | 4 | 3.4E-38 to 3.4E+38 |
| double | 8 | 1.7E-308 to 1.7E+308 |
| long double | 10 | 3.4E-4932 to 1.1E+4932 |

Character type

Character types store characters value.

Table :-Char type.

|  |  |  |
| --- | --- | --- |
| **Type** | **Size(bytes)** | **Range** |
| char or signed char | 1 | -128 to 127 |
| unsigned char | 1 | 0 to 255 |

Arrays

“To declare any array in C, a programmer specifies the type of elements and the number of elements required by an array as follows –

type arrayName [ arraySize ];

This is called as a single-dimensional array. The arraySize has to be an integer constant greater than zero and the type can be any valid C data type. For example, to declare a 10-element array called balance of type double, use this statement −

double balance[10];”

Here balance is a variable array which is enough to hold up to 10 double numbers.

In the program developed integer, character and string type was used.

## Structure

Structure is a user-defined datatype in C language which allows the combination of data of different types together.

For example, struct keyword is used to define a structure. struct defines a new data type which is a combination of primary and derived datatypes.

**Syntax:**

As can be seen in the syntax above, the start is with the struct keyword, then it's optional to provide the structure a name, then inside the curly braces, it is necessary to mention all the member variables, which are nothing but normal C language variables of different types such as int, float, array etc.

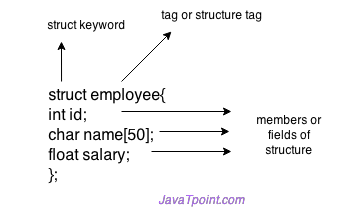


Figure 2:-structure syntax

struct record

{

char name[20];

char servicedate[20];

char registeration[20];

int ser;

int type;

int id;

}r;

Above line shows structure data type that was used in the program which has three member variables of char type and three member variables of integer type. The structure tag r was used in other functions and loops to access the variables of the structure.

## File Handling

In programming, some specific input data may be required to be generated several numbers of times. Sometimes, it is not sufficient to only display the data on the console. The data to be displayed may be quite large, and only a limited amount of data can be displayed on the console. Furthermore, since the memory is volatile, it is impossible to recover the programmatically generated data time after time. However, if need be, it may be stored onto the local file system which is volatile and can be accessed every time. Here comes the need of file handling.

File handling in C allows creating, updating, reading, and deleting the files stored on the local file system through the C program. The following operations can be performed on a file.

* Creation of a new file
* Opening an existing file
* Reading from the file
* Writing to the file
* Deleting the file

Table :-File handling



FILE\* fp; char ch; int value;

printf("\n Enter Customer name:-");

scanf("%s", &r.name);

printf("\n Enter Service date:-");

scanf("%s", &r.servicedate);

fflush(stdin);

printf("\nEnter registeration number:-");

scanf("%s", &r.registeration);

fflush(stdin);

menu();

fp = fopen("idnumber.txt", "r+"); // read mode

if (fp == NULL)//if there will be no file and no data inside the file it will show the error

{

perror("Error while opening the file.\n");

exit(EXIT\_FAILURE);

}

fscanf(fp, "%d", &value);//reading the id from the file

fclose(fp);

value++;

Above line shows the file handling technique which was used in the program for the booking function which hold the data in the structure name record. The file pointer fp uses the function fopen to open the file from the directory in read mode or r+ mode. Fscanf command was used to read the data from idnumber file and store in integer value.

## Function

The general form of a function definition in C programming is as follows −

return\_type function\_name( parameter list ) {

body of the function

}

“A function definition in C programming consists of ; *function header* and *function body*. Here are the parts of a function −

* **Return Type** − A function may return a value. **return\_type** is the data type of the value the function returns. Some functions perform desired operations without returning a value. In this case, return\_type is the keyword **void**.
* **Function Name** − This is the actual name of a function. The function name and the parameter list together outline the function signature.
* **Parameters** − A parameter is like a placeholder. When a function is executed, a value to the parameter is passed. This value is referred to as an actual parameter or argument. The parameter list refers to the type, order, and number of the different parameters of a function. Parameters are optional; that is, a function may contain no parameters at all.
* **Function Body** − This contains a set of statements that define what the function does.”

void menu()

{

int option;

int fee;

Above line shows the user defined function name menu used in the program. This function has no parameters so when the function will be called there will be no arguments thrown to the function. The function neither returns anything as the function return type is Void. This function was used in the program to get the user input for the selected service and the type of service.

# 5)RESULTS.

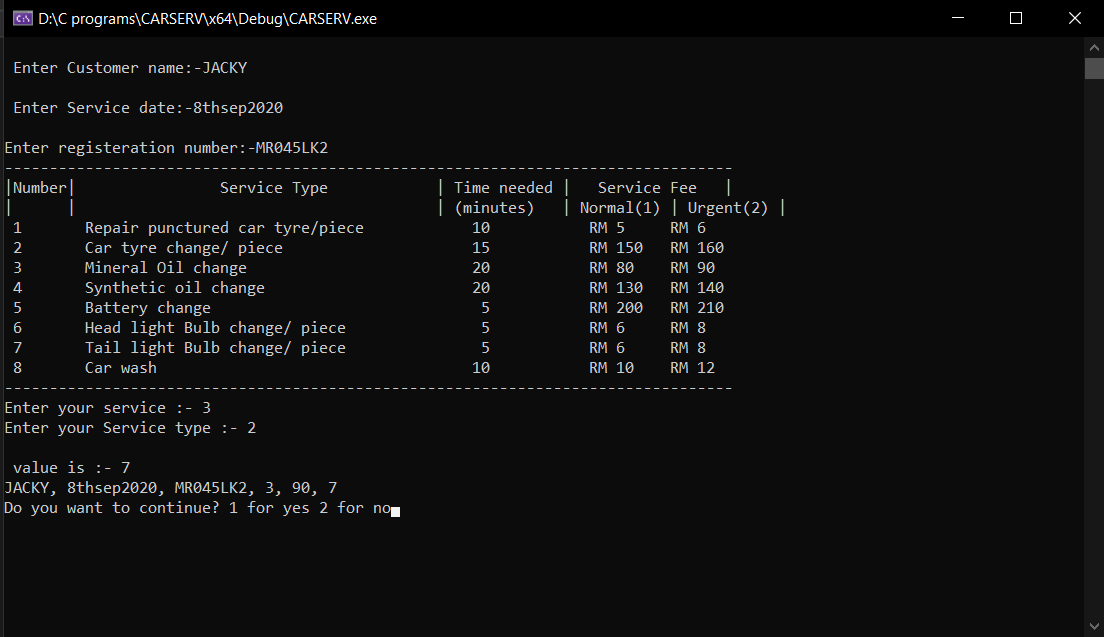


Figure 3:- Output Screen.

Figure 3 shows the output of the program where the customer name, Service date and registration number was input, then the menu was shown where the service and the service type was inputted after that a unique id was provided and the invoice is generated on the next line. Then a user continue for the next iteration was asked.

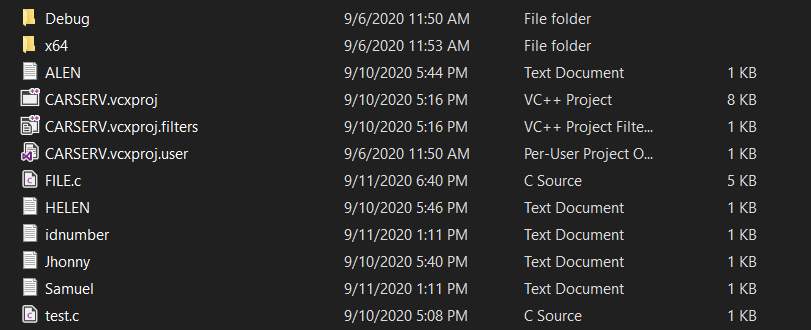


Figure 4:- Files before execution.

Figure 4 shows the files in the folder where there are 4 files generated for each individual customer. There

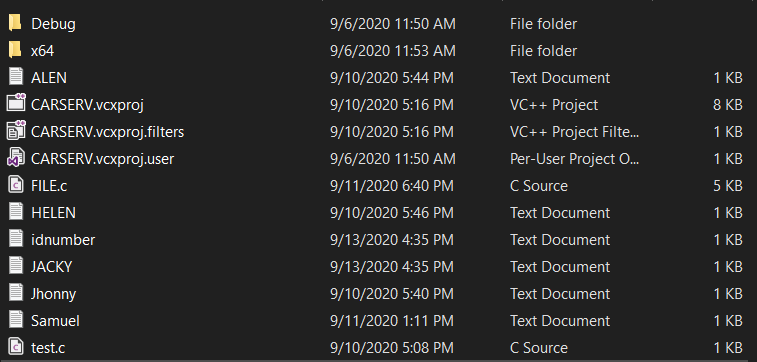


Figure 5:- Files after execution.

Figure 5 shows the file inside the folder where there is new text file generated in the name of Jacky and Figure 6 shows the data of the user along with his id and all the necessary files.

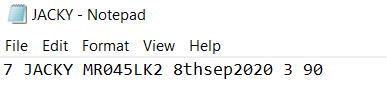


Figure 6:- File content

# 6.) Conclusion.

A menu was made including the details such as service type, time needed and service fee using printf function and the overall function was inside a user defined function named as menu hence it can be concluded that the first objective was achieved.

Another user defined function was made named as booking where the information of user was recorded such as the customer name, service date, Car registration number, Service needed, type of service and service fee. The data of the parameters were stored inside a structure named record, to maintain a correct data log a case was provided where if the user entered any number less than 1 and greater than 8 the program will show an error and ask the user to select the data again, since there are eight types of service was mentioned inside the menu. Hence it can be said that the second objective was achieved.

After the parameters were recorded a unique id was generating using fread function from the file where the program was reading the previous id and increase the value by1 for sequential generation of the number. The invoice was then shown on the display and new individual files was generated for each Customer where the file name was the customer name. Hence it can be concluded that the third objective was achieved.

It can be concluded that the overall assignment objective was achieved to create a Fast track car service management system in C.

# REFRENCES: -

Liberty, J. and Cadenhead, R. (2011). *Sams teach yourself C in 24 hours*. Indianapolis, Ind.: Sams Pub.

Ackermann, E. (2013). *C Programming Language Essentials*. Research & Education Association.

Scribd. (2020). *C Programming for Beginners: Your Guide to Easily Learn C Programming In 7 Days*. [online] Available at: https://www.scribd.com/read/347247914/C-Programming-for-Beginners-Your-Guide-to-Easily-Learn-C-Programming-In-7-Days [Accessed 15 Aug. 2020].

Programiz.com. (2020). *C Structures (With Examples)*. [online] Available at: https://www.programiz.com/c-programming/c-structures [Accessed 28 Aug. 2020].

OverIQ.com. (2020). *fwrite() Function in C - OverIQ.com*. [online] Available at: https://overiq.com/c-programming-101/fwrite-function-in-c/ [Accessed 2 Sep. 2020].