CHAPTER 1

INTRODUCTION

IBall for your eyeballs is a project which aims to provide a one stop destination for all your contact lens needs using the concept of data structures in real time and analysing and implementing the same using C language. The data structure used in this project is linked list. Many user defined functions are declared and defined which are called based on the needs of the user. This project is built after analysing and understanding the technological needs to develop an application.

1.1 PROBLEM STATEMENT

Developing an application that will prove to be a one stop destination for all your contact lens needs using the concept of data structures in real time and analysing and implementing the same using C language.

1.2 OBJECTIVES

Analyse and understand the technological needs to support tools and develop an application which will provide a platform for the societal needs of a hassle-free experience to shop for their contact lens needs. This project is designed to provide a large variety of products for the benefit of customers. The main objective of this project is to provide high-quality eyewear to millions of users at affordable prices from the comfort of their homes.

1.3 EXPECTED OUTCOMES

The user will be able to choose and buy from a variety of products. The user can also add, delete and modify the contents of the cart. The total amount to be paid by user will be calculated after the user is done modifying the contents of the cart and displayed on the checkout screen. The user can then give their right and left eye powers and their base curve to the software. The area of delivery is also stored based on which the estimated time of delivery is displayed.

1.4 METHODOLOGY TO BE FOLLOWED

The concept of data structures is used to implement the concept of shopping cart. Each node of the linked list has four parts – the name of the item purchased, quantity, price and pointer to the next node. The concept of linked lists makes the processing and accessing of data easy. The user is first given two major categories to choose from and them systematically broken down into sub categories. The user can first add all the items he/she wishes for. The software then takes the user to review their purchases where they can delete any item, they have changed their mind about. The last process is the checkout process where the total amount to be paid by the user is displayed along with the option to enter their eye powers and the estimated time of delivery is displayed based on the area of delivery.

1.4 REQUIREMENT SPECIFICATIONS

1.4.1 Software Requirement

Any C compiler (preferably Turbo C++).

1.4.2 Hardware Requirements

- 1 GB RAM (2 GB+ recommended)
- 9-58 GB of free hard disk space,
- Basic GPU
- Intel Pentium or compatible processor 1.6 GHz minimum (2GHz+ recommended)
- 1024x768 or higher resolution monitor.

CHAPTER 2

DATA STRUCTURES

In computer science, a data structure is a data organization, management, and storage format that enables efficient access and modification. More precisely, a data structure is a collection of data values, the relationships among them, and the functions or operations that can be applied to the data.

The characteristics of data structures are as follows.

1. Linear or non-linear: This characteristic describes whether the data items are arranged in chronological sequence, such as with an array, or in an unordered sequence, such as with a graph.

2. Homogeneous or non-homogeneous: This characteristic describes whether all data items in a given repository are of the same type or of various types.

3. Static or dynamic: This characteristic describes how the data structures are compiled. Static data structures have fixed sizes, structures and memory locations at compile time. Dynamic data structures have sizes, structures and memory locations that can shrink or expand depending on the use. Classification of data structures is shown in figure 2.1.

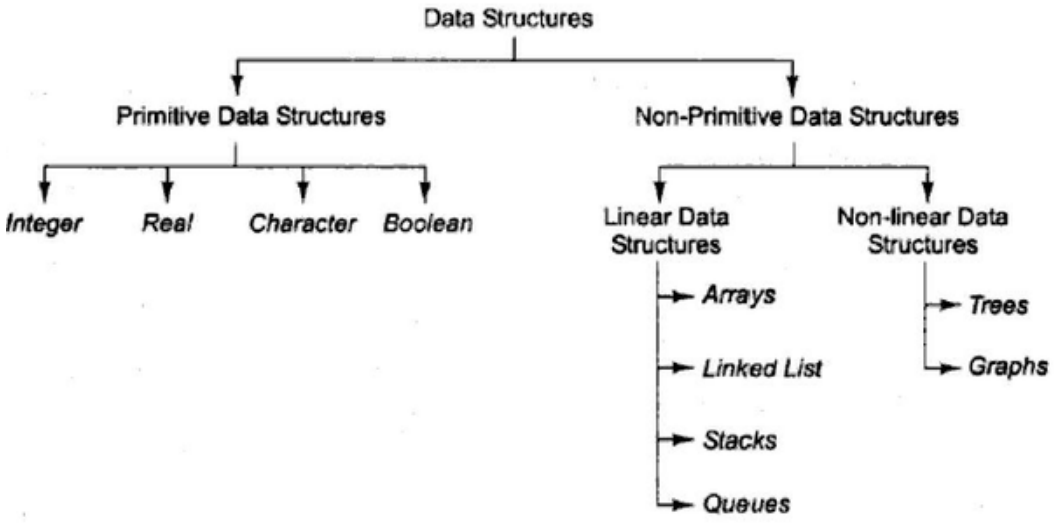

Fig 2.1: Data Structure Classification

2.1 STACKS

The stack works on the principle of last in first out. This implies the component entered in the stack last can be shown or erased first. It can be utilized to store the information in a specific way. The stack can be constrained by one variable or we can say that we can erase or enter a component utilizing one variable. Generally, the control variable is declared globally rather than passing it from function to function.

The two operations that can be performed on a Stack are: Push operation which inserts an element into the stack. Pop operation which removes the last element that was added into the stack as It follows Last in First Out (LIFO) Order as shown in figure 2.2. Insertion and deletion in stacks take place only from one end of the list. In stacks we maintain only one pointer to access the list which always points to the last element present in the list.

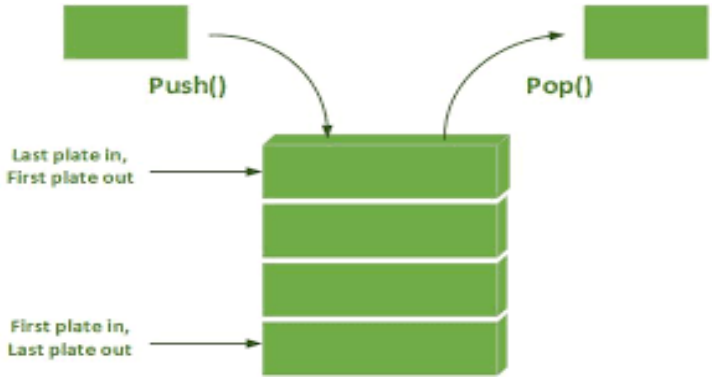

Fig 2.2: Representation of stack

2.1.1 Applications of Stack

1. Expression Evaluation - Stack is used to evaluate prefix, postfix and infix expressions.
2. Expression Conversion - An expression can be represented in prefix, postfix or infix notation. Stack can be used to convert one form of expression to another.
3. Syntax Parsing - Many compilers use a stack for parsing the syntax of expressions, program blocks etc. before translating into low level code.

2.2 QUEUE

Queues are based on the first in first out principle. The element inserted at the first will come out of the list first. The operation which is used to insert the data is called insert operation. The operation which is used to delete the data is called delete operation. The Insertion of element and deletion of element in queues can take place from the opposite ends. The insertion happens at the rear end and the deletion happens at the front end of the list as shown in figure 2.3.

In queues we maintain two pointers to access list. The front pointer always points to the first element inserted in the list and is still present, and the rear pointer always points to the last inserted element. There is one problem with the simple queue, after inserting the element if we delete that particular element, the memory which was occupied can't be used again to insert the elements, since we can only add linearly.

Fig 2.3: Representation of queue.

2.2.1 Applications of Queues

1. Queue is useful in CPU scheduling, Disk Scheduling. When multiple processes require CPU at the same time, various CPU scheduling algorithms are used which are implemented using Queue data structure.
2. When data is transferred asynchronously between two processes. Queue is used for synchronization. Examples: IO Buffers, pipes, file IO, etc.

2.3 TREES

Tree is non-linear data structure which stores information naturally in the form of hierarchy style. It represents nodes connected by edges. We know that a linked list is a chain of nodes which connect through next pointers, similar is tree but with a slight difference as in tree each node can be connected to multiple nodes. Binary tree representation is shown in figure 2.4.

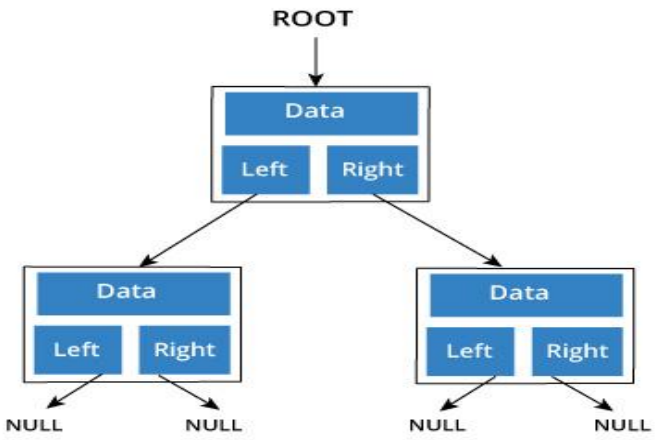

Fig 2.4: Binary Tree Representation.

There is a special pointer called root which points to the node that is the parent of all the other nodes. The nodes that don't have any children have their right and left pointers point to NULL.

2.3.1 Applications of Trees

1. Binary Search Trees (BSTs) are used to quickly check whether an element is present in a set or not. They are used for easy traversals.
2. Heap is a kind of tree that is used for heap sort.
3. A modified version of tree called Tries is used in modern routers to store routing information.
4. Most popular databases use B-Trees and T-Trees, which are variants of the tree structure we learned above to store their data.
5. Compilers use a syntax tree to validate the syntax of every program you write.

2.4 GRAPHS

A graph is a non-linear data structure which consists of vertices and edges as shown in figure Also represented as G=(V,E) where V is a set of vertices and E is set of edges. Graphs are usec represent networks. There are two ways in which a graph can be represented: -

- Sequential or Array representation
- Linked Representation

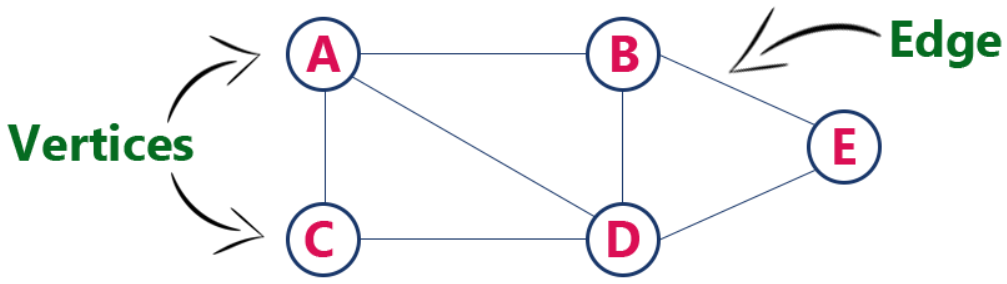

Fig 2.5: Representation of graphs

2.4.1 Application of graphs

1. Computer Science: In computer science, graph is used to represent networks of communication, data organization, computational devices etc.
2. Physics and Chemistry: Graph theory is also used to study molecules in chemistry and physics.
3. Social Science: Graph theory is also widely used in sociology.
4. Mathematics: In this, graphs are useful in geometry and certain parts of topology such as knot theory.
5. Biology: Graph theory is useful in biology and conservation efforts.

2.5 LINKED LIST

A linked list is a linear data structure where each element is a separate object. Linked list elements are not stored at contiguous location; the elements are linked using pointers as shown in figure 2.6. Each node of a list is made up of two items - the data and a reference to the next node. The types of Linked Lists are as follows.

1. Singly linked list.
2. Doubly linked list.
3. Circular linked list.

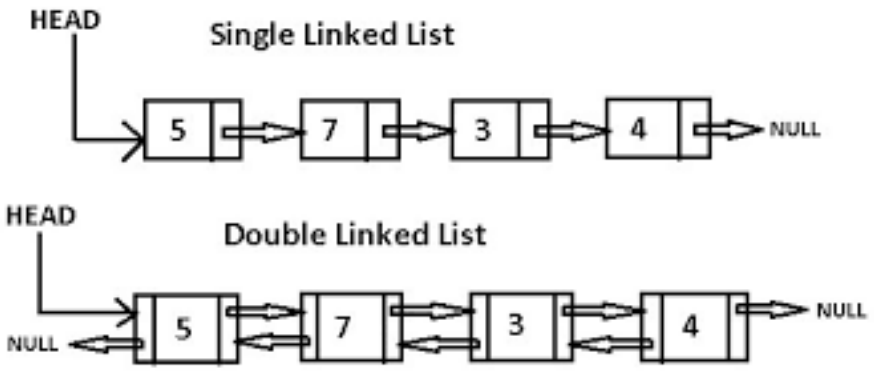

Fig 2.6: Pictorial representation of linked lists.

The principal benefit of a linked list over a conventional array is that the list elements can be easily inserted or removed without reallocation or reorganization of the entire structure because the data items need not be stored contiguously in memory or on disk, while restructuring an array at run-time is a much more expensive operation. Linked lists allow insertion and removal of nodes at any point in the list, and allow doing so with a constant number of operations by keeping the link previous to the link being added or removed in memory during list traversal.

The major disadvantages of using linked lists is that memory is wasted as pointers require extra memory for storage. No element can be accessed randomly; it has to access each node sequentially. Reverse Traversing is difficult in linked list.

2.5.1 Applications of linked list

1. Linked Lists can be used to implement Stacks, Queues.
2. Linked Lists can also be used to implement Graphs. (Adjacency list representation of Graph).
3. Implementing Hash Tables: Each Bucket of the hash table can itself be a linked list. (Open chain hashing).
4. Undo functionality in Photoshop or Word. Linked list of states.
5. A polynomial can be represented in an array or in a linked list by simply storing the coefficient and exponent of each term.
6. However, for any polynomial operation, such as addition or multiplication of polynomials, linked list representation is easier to deal with.
7. Linked lists are useful for dynamic memory allocation.
8. The real-life application where the circular linked list is used is our Personal Computers, where multiple applications are running.
9. All the running applications are kept in a circular linked list and the OS gives a fixed time slot to all for running. The Operating System keeps on iterating over the linked list until all the applications are completed.

CHAPTER 3

DESIGN AND ALGORITHM

An algorithm is a well-defined procedure that allows a computer to solve a problem. Another way to describe an algorithm is a sequence of unambiguous instructions. The use of the term 'unambiguous' indicates that there is no room for subjective interpretation. Every time you ask your computer to carry out the same algorithm, it will do it in exactly the same manner with the exact same result.

3.1 DESIGN

This project is designed using the concept of data structures (linked list) in C language. Some of the operations the functions in this program are designed to do are:

- Accepting the user details such as Name, phone number etc.
- Display the three main categories of products to choose from, from where the user will be directed towards sub categories and can recursively choose from a variety of products.
- Adding the desired products to the cart.
- Reviewing the products added to the cart from which the user can delete items if he wishes to do so.
- Calculating the total amount to be paid by the user. It displays an appropriate error message if the cart is empty stating the same.
- Accepting the delivery address and displaying the estimated time of delivery of the product.
- Accepting the eye powers of the user along with their base curve value.
- Displaying some precautions, the user must know before they start using the product purchased.

3.2 ALGORITHM

STEP 1: Start the program.

STEP 2: Include all the header files that will be required in the program.

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<string.h>
#include<ctype.h>
```

STEP 3: Declare the structure having four parts to accept the item name, its price, quantity and a pointer to the next node in the linked list. Also declare a global variable first that is initialized to NULL. Also declare all the modules that the user wishes to be implemented in the program.

```
struct node
{
        char *item;
        int q;
        int price;
        struct node *link;
};
struct node *first = NULL;
```

STEP 4: Accept the users name and phone number and calculate the length of the phone number. If the phone number exceeds 10 characters throw an error message and accept the phone number till a valid phone number is entered .

STEP 5: Display the two main categories of products i.e. Contact lens, Lens solution.

STEP 6: Accept the users input and direct the user towards the sub categories if any else display the detailed variety of products. Repeat this step if there are sub categories.

STEP 7: Add the desired products to the cart and ask the user if he wants to continue shopping. If yes, Go to step 5. If no, Go to Step 8.

STEP 8: Display the final shopping cart for reviewing.

STEP 9: Check if the cart is empty. If yes, go to step 17 and display an error message stating the same. If no, Go to step 9.

STEP 10: Ask the user if he wants to delete any item.

STEP 11: If yes, delete the desired item and go to step 10. If no, Go to step 13.

STEP 12: Ask the user if he wants to delete more items. If yes, go to step 10. If no, go to step 13.

STEP 13: Display the final amount to be paid by the user along with the final shopping cart.

STEP 14: Accept the user delivery address.

STEP 15: Ask the user to enter the right eye power, left eye power and base curve.

STEP 16: Display the important precautionary information that is required before using the product and display the estimated date of delivery.

STEP 17: Exit from the program.

CHAPTER 4

IMPLEMENTATION

4.1 MODULE 1 IMPLEMENTATION

Before entering the main function, all the header files that will be required during the development should be included. After including all the header files, the linked list which is used to store all the cart information should be defined. The linked used to store he purchase information should have four parts – item name, item quantity, price of the item, pointer to the next item in the linked list. The linked list used to develop this program is a singly linked list. This means that the last node in the linked list will always have its pointer part initialized to NULL. After declaring the linked list, we should declare a global variable – first. The global variable first should be initialized to NULL. Once the items are added to the cart, the first variable will point to the first item in the shopping cart. After this, all the modules that the developer wishes to be implemented are declared. Module 1 accepts the user information and displays the two main categories for the user to choose from. The option to choose from the categories is recursively printed for the user to add multiple items to the cart. The option to add items keeps displaying until the user specifically denies it. This is implemented using the do while loop. The user information that is accepted is stored in two variables – name, phone number. If the user enters an invalid phone number an error message is displayed. Only when the user enters a 10-digit phone number can the program proceed further else, the option to enter a valid phone number keeps displaying. This is implemented using awhile loop and calculating the length of the accepted phone number at the end of each iteration. Only when the length of the accepted is exactly equal to 10 will the further information about products be displayed. Once the user chooses a option the programs goes to that specific function and process the choices of the user.

4.2 MODULE 2 IMPLEMENTATION

If the user chooses the contact lens option from the main menu, he/she is directed towards the sub categories present in this menu i.e. Colored lenses and colorless lenses. If the user chooses colored lenses, the colors available are displayed. The user can choose any option they wish for and the sub options for that if any are displayed. The user is then asked to enter the number of boxes they require after the cost of each box is displayed. For the colorless option the cost of each box is displayed along with the option to enter the number of boxes required. If the user chooses lens solution option from the main menu, he/she is displayed option of varying quantities along with their prices for the user to choose from. After the user adds an item to the cart, an option to add more items is displayed. If the user wishes to continue, he can choose to do so else the user moves on the review function. If the user choses to continue, they are taken to the main category of options and the process is repeated again. Once a user adds a item to the cart, the item is added to the rear end of the linked list along with the item name, its quantity and the cost of the item. The pointer part of the linked list is used to point to the other items in the cart. The global variable ‘first’ points to the newly added item in the cart every time. The last item in the linked list will always have its pointer part declared as NULL. This is used to identify the end of the linked list. If the shopping cart is empty, that is there are no nodes in the linked list the global variable ‘first’ has NULL value. While adding a new item in cart, there are a few cases that need to be considered. If there is no item in the cart the new item is added and the pointer of that item is declared as NULL. If there are already items present in the cart, the new item is added and the pointer is made to point to the other items. This process is done every time the user chooses to add a new item to the list. Checking if the linked list is empty can be done easily by checking if the global variable ‘first’ is NULL. After the user is done adding all the items, the user is directed towards the review process.

4.3 MODULE 3 IMPLEMENTATION

Once all the items user wishes are added to the cart, he/she can delete any item if want to. The review cart option displays all the contents of the cart with their item name, quantity and cost of each item. The user can delete multiple items. If the shopping cart is empty an appropriate error message is displayed stating the same. To know if the shopping cart is empty means to check if there are any nodes existing in the linked list and this can be done by checking if the global variable 'first' is NULL. If the user wishes to delete any item from the linked list there are conditions that need to be considered. If there is only one item in the linked list and the user wishes to delete that item itself, the node is deleted and the global variable 'first' is initialized to NULL. If there are multiple nodes, then the entire linked list is traversed until the node that the user wishes to delete is found and when found the node is deleted. To traverse the entire linked list the temporary pointer should start from the node the first node till the node whose pointer is NULL. The node before the node to be deleted is made to point to the node after it. If the first node is to be deleted when there are multiple nodes in the linked list, then the global variable 'first' is also made to change its value and points to the next item in the linked list. If the user enters a value that does not exist in the linked list an error message is displayed stating that the data is not found. This process is repeated every time the user chooses to delete an item. This is implemented by recursively giving the user an option to choose whether he/she wants to delete any item. This is done using do-while loop. Once the user is done reviewing the cart and has finalized the list, the user proceeds to the checkout section of the program.

4.4 MODULE 4 IMPLEMENTATION

After the user reviews the cart, the checkout and the final stage of the process is started. Here, the final shopping cart is displayed along with the total amount to be paid by the user. This can be implemented by making a temporary variable traverse the linked list and adding the cost of each is added and stored in a variable. It is in this stage that the user can enter their eye powers and their base curve value. After entering their eye details the user enters the locality in which the delivery is to be made. Based on this the estimated time of delivery is displayed. A few precautions that the user must follow and displayed in the end before the user exits from the program.

CHAPTER 5

RESULTS

The below screenshots are the working results of the program.

1. Accepting user information.

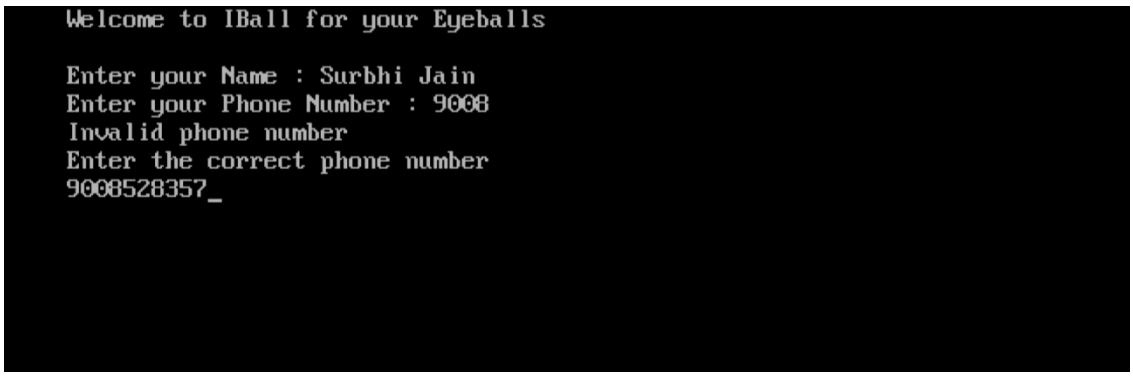

Fig 5.1: Result Screenshot-1

2. Displaying the main categories.

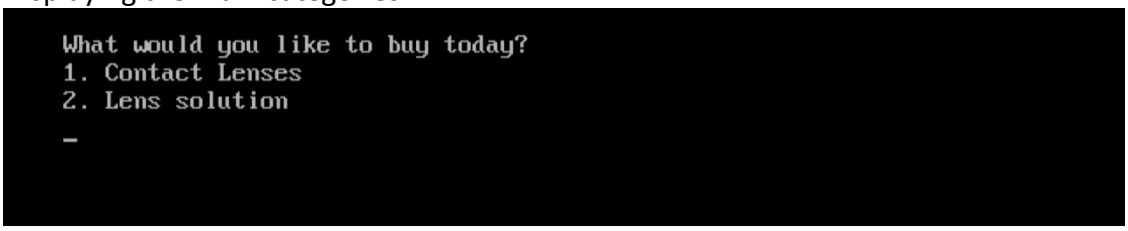

Fig 5.2: Result Screenshot-2

3. Sub options in contact lens category.

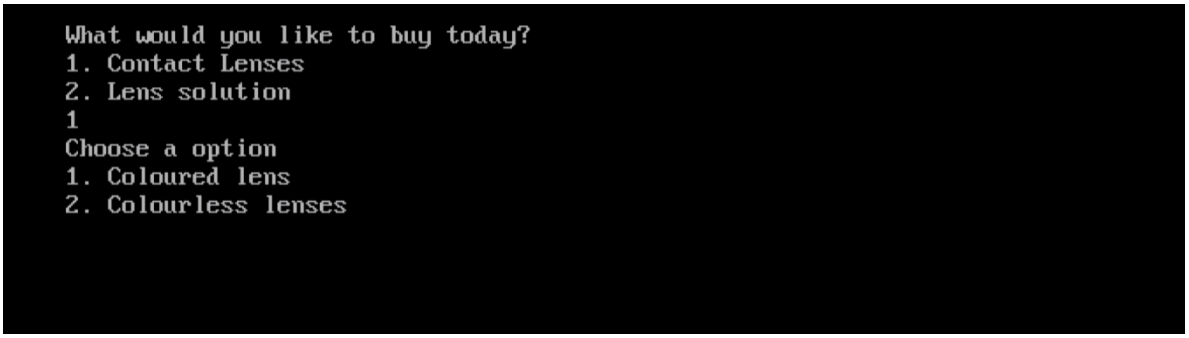

Fig 5.3: Result Screenshot-3

4. Sub options in colored lens category.

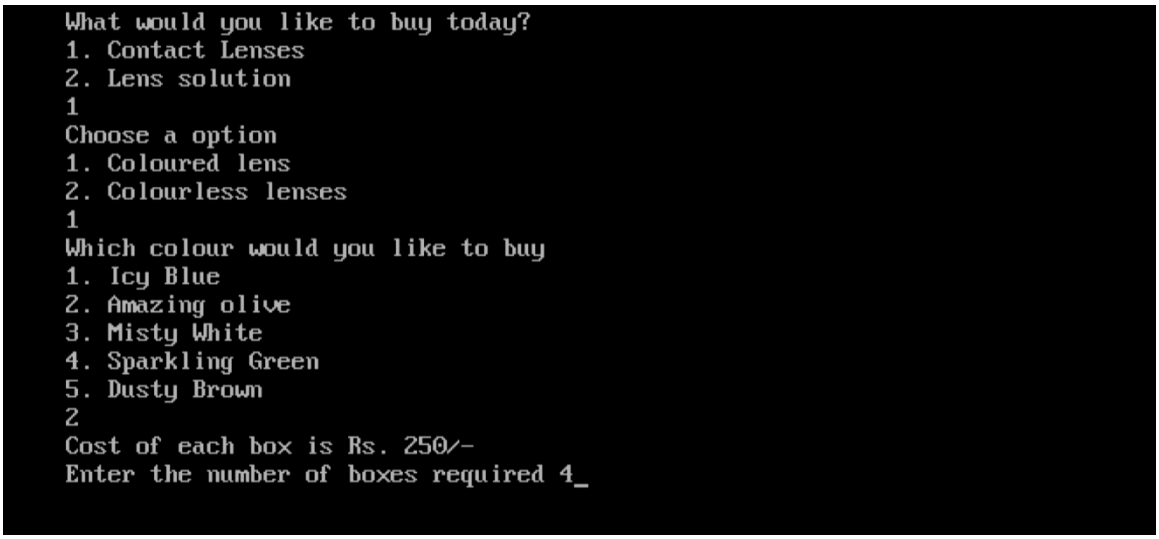

Fig 5.4: Result Screenshot-4

5. Option to continue shopping. (Recursive option).

Fig 5.5: Result Screenshot-5

6. Colorless lens category after choosing to continue shopping and choosing to buy colorless lenses from the lens category.

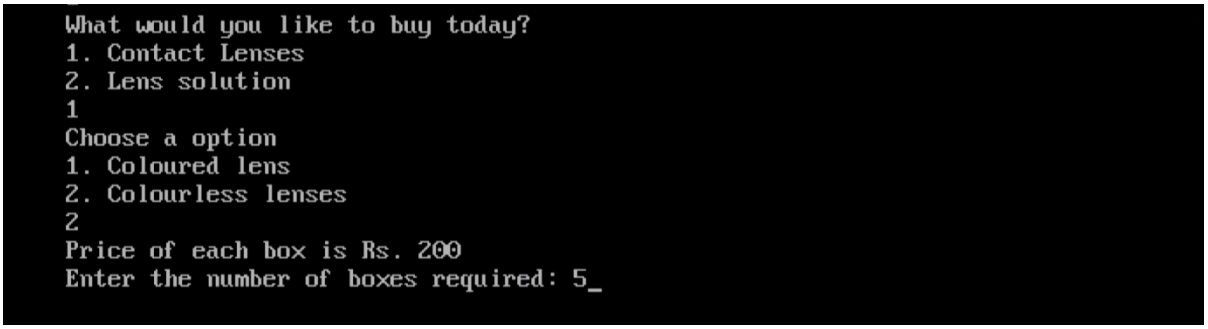

Fig 5.6: Result Screenshot-6

7. Lens solution category after choosing to continue shopping and choosing to buy lens solution.

```
What would you like to buy today?
1. Contact Lenses
2. Lens solution
2
Choose a item
1. 60 ml Rs. 70
2. 120ml Rs. 130
3. 180ml Rs. 190
4. 360ml Rs. 350
5. 500ml Rs. 470
3
Enter the number of boxes required: 2
```

Fig 5.7: Result Screenshot-7

8. Review cart screen.

```
SHOPPING CART

The contents of cart are
Sl.no. Item             Quantity       Cost
1.     Colorless lens      5           1000
2.     Amazing olive       4           1000
3.     Solution 180ml      2           380
```

Fig 5.8: Result Screenshot-8

9. Option to ask the user if he/she wishes to delete any items.

```
SHOPPING CART

The contents of cart are
Sl.no. Item             Quantity       Cost
1.     Colorless lens      5           1000
2.     Amazing olive       4           1000
3.     Solution 180ml      2           380
Do you wish to delete any item from cart
1.YES 2.NO
1
Enter the number you want to delete 1
1 number item is deleted
```

Fig 5.9: Result Screenshot-9

10. Option to ask if the user wishes to delete more items once a item has been deleted. (Recursive deletion option)

```
Do you wish to delete more items
1.Yes   2.No
2
```

Fig 5.10: Result Screenshot-10

11. Checkout screen with total amount to be paid.

```
CHECKOUT SCREEN

Thank you Surbhi Jain for shopping with us (IBall for your Eyeballs!!)

Your final shoping list is:
Sl.no. Item             Quantity       Cost
1.     Amazing olive       4           1000
2.     Solution 180ml      2           380

The amount to be paid is: Rs1380
```

Fig 5.11: Result Screenshot-11

12. Accepting users eye powers and base curve. Also accepting the area of delivery. Displaying of the precautions to be taken care of before using the product.

```
The amount to be paid is: Rs1380
Enter your right eye power: -2.5
Enter your left eye power: -2.75
Enter your base curve: 8.4

Enter area of delivery: Wilson Garden

Your order will be delivered to you within 3 working days.
Our delivery partner will contact you on 9008528357 for delivery instructions.

PRECAUTIONS

The lenses purchased are monthly disposable and the lens solution should be used
 within 3 months from the date opening.
Please do not use products after they expire or if their seal is broken
For further information send your queries to Email - arohi0812@gmail.com
```

Fig 5.12: Result Screenshot-12

13. Screenshot of the review screen and the checkout screen if the shopping cart is empty. An error message is displayed stating the same.

Fig 5.13: Result Screenshot-13

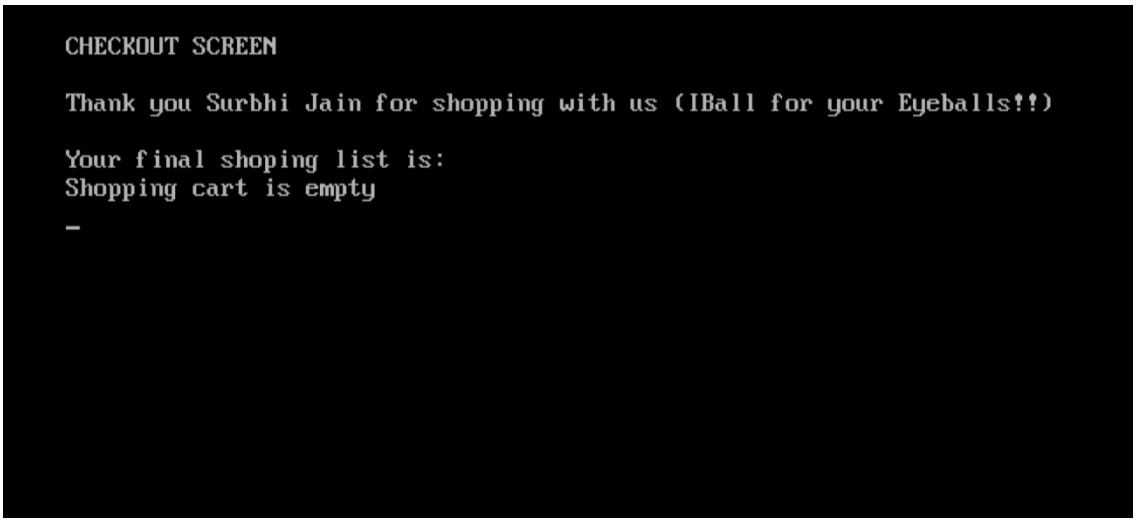

Fig 5.14: Result Screenshot-14

CHAPTER 6

CONCLUSION

In today's fast paced world, we tend to ignore our basic necessity and our daily requirements. This application was developed to provide all its users with the ease and the ability to order their contact lenses from the comfort of their homes. This is a very user – friendly application that can help you place your desired order in just a few steps without any hassles. The core reason for technology to be trusted more than the manual work is exactly that, reduction of time and increase in efficiency. This program aims to contribute to this very cause. The time consumed by a person in manually ordering a pair of eye-lens and the hassle caused due to the miscommunication between the consumer and supplier could be avoided thanks to this project. On the whole, this would give a comparatively faster and more efficient approach towards providing a smooth shopping experience. Hence, it helps the customer and eventually the society by making people's life more comfortable and thus saving the precious time that would otherwise be spent on insignificant tasks. Developing this project has also led us to have a better understanding of the data structure topics. Implementing the concepts in real life has made us more confident in the topic and we can henceforth be able to develop real time applications with much more clarity in mind. This has also led us to have a better grasp on the C language. The main objective of understanding and analyzing the technological needs of the society has been achieved.

REFRENCES

[1] S.K Srivasta and Deepali Srivasta – "*Data Structures Through C in Depth*".

[2] "*Let Us C"* – Ashutosh Pandey, Yashavant Kanetkar.

[3] https://www.geeksforgeeks.org/storage-for-strings-in-c/

[4] https://en.wikipedia.org/wiki/Data_structure

[5] https://github.com/topics/data-structures

[6] https://www.edureka.co/blog/c-data-structures/

[7] https://www.thecrazyprogrammer.com/2016/04/applications/