A Mini Project Report

on

**TO-DO LIST**

**DATA STRUCTURES LAB**

II B.E CSE-B, III SEMESTER

By:

P.Vaishnavi Reddy

1602-19-733-114

&

M.Tejaswini

1602-19-733-113



**Department of Computer Science & Engineering**

**Vasavi College of Engineering (Autonomous)**

**Ibrahimbagh, Hyderabad-31**

**2020-2021**

**ACKNOWLEDGEMENT**

We respect and thank Dr.V.Sireesha mam for constant support and motivation which helped us immensely in completing our project successfully.

**TABLE OF CONTENTS:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CONTENT** | **PAGE NO.** |
| 1 | Abstract | 3 |
| 2 | Introduction | 4 |
| 3 | Design-strategy-algorithm- modules | 5 |
| 4 | Code/Implementation | 6-10 |
| 5 | Outputs | 11-14 |
| 6 | Conclusion | 15 |
| 7 | Reference | 16 |
|  |  |  |

**1.ABSTRACT**

**OBJECTIVE:**

The main objective of the project is to have a to do list in which we write our works to perform

Existing system:

It is similar to the all to do apps

Working:

In this project admin is a user who wants to create a to do list ,so that he won’t forget the things which he want to do. The user will create a to do list and he can add or delete the works he want to do and he can see the list of works which he scheduled.

Concepts used:

Linked list(insertion and deletion).

Data Structure: linked list

Language:C

**2.INTRODUCTION**

A person who wants to make a note of all works in a day can use this app and it is vey useful for the user and the user can view all the works which he want to do.

Objective:

The main objective of this project is to have a note of all works which we want to do and not to have an incomplete work at the end.

Description:

* Firstly, our program is a user friendly program
* The user has to create a to do list
* While creating a list there is a option whether to add elements or not it displays yes or no?
* If you want to add the user have to enter y and else then n
* Then you can view your to do list by entering the respective key shown in the program
* If the user wants to delete one of the element then he can enter the specific no.of the element and can delete that element
* And the user can update his to do list by entering the respective key
* And you can view the updated to do list by entering the key which is assigned to view the to do list option
* If the total process of adding deleting and updating is done as user wishes
* Then the user can enter the exit key.

**3.DESIGN**

**3.1** STRATEGY

The aim of this project is to store the works which the user wants to do using Linked list

This strategy is used so that it would be easy for the user to add,del and update the elements

**3.2** Algorithm

Step1: want to create a to do list enter your choice is 2

Step2: do you want to add; yes or no?

Step 3: if yes enter y, add

Step4: if you want to del enter your choice is 3

Step5: enter the no. of the element you want to del

Step6: to view the elements in the to do list

Step7: enter your choice is 1

Step8: to update your to do list

Step9: enter your choice is 4

Step10: it asks you to add; yes or no. if yes then you can add elements

Step11: if you don’t want to add enter n

Step12: to view the elements of to do list enter your choice is 1

Steo13: to exit enter your choice is 5.

**4.IMPLEMENTATION**

#include <stdio.h>

#include <stdlib.h>

typedef struct ToDo todo;

struct ToDo{

char data[100];

todo \*link;

int count;

};

todo \*start=NULL;

int main()

{

int choice;

welcomeUser();

while(1){

system("color 3F");

system("cls");

printf("\n1.See Your ToDo List");

printf("\n2.Create Your ToDos");

printf("\n3.Delete Your ToDos");

printf("\n4.Update Your ToDos");

printf("\n5.Exit");

printf("\n\nEnter your choice..");

scanf("%d",&choice);

switch(choice){

case 1:

seeToDo();

break;

case 2:

createToDo();

break;

case 3:

delToDo();

break;

case 4:

updateToDo();

break;

case 5:

exit(0);

}

}

}

void welcomeUser(){

system("color 4F");

printf("\n\n\n\n\n");

printf("\t------------------------------------------------------------------------------------------------------\n\n");

printf("\t#################################### YOUR TODO LIST APP ##############################################\n\n");

printf("\t------------------------------------------------------------------------------------------------------");

printf("\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*WELCOME\*\*\*\*\*\*\*\*\*\*\n\n\n\n\n\n\n\n\n\t");

system("pause");

}

void seeToDo(){

system("cls");

todo \*temp;

temp=start;

if(start==NULL){

printf("\nEmpty TODO\n\n");

}

while(temp!=NULL){

printf("%d)",temp->count);

puts(temp->data);

fflush(stdin);

temp=temp->link;

}

printf("\n\n\n");

system("pause");

}

void createToDo(){

char k;

todo \*t,\*temp;

system("cls");

while(1){

printf("\nWant to add?y/n");

fflush(stdin);

scanf("%c",&k);

if(k=='n')

break;

else{

if(start==NULL){

t=(todo \*)calloc(1,sizeof(todo));

start=t;

printf("\nADD it..\n");

fflush(stdin);

gets(t->data);

t->count=1;

start->link=NULL;

}

else{

temp=(todo \*)calloc(1,sizeof(todo));

printf("\nADD it..\n");

fflush(stdin);

gets(temp->data);

temp->link=NULL;

t->link=temp;

t=t->link;

}

fixcount();

}

}

}

void delToDo(){

system("cls");

int d;

todo \*temp1,\*temp;

printf("\nEnter the no of the todo you want to remove\n");

scanf("%d",&d);

temp1=start;

temp=start->link;

while(1){

if(temp1->count==d){

start=start->link;

free(temp1);

fixcount();

break;

}

if(temp->count==d){

temp1->link=temp->link;

free(temp);

fixcount();

break;

}

else{

temp1=temp;

temp=temp->link;

}

}

system("pause");

}

void fixcount(){

todo \*temp;

int i=1;

temp=start;

while(temp!=NULL){

temp->count=i;

i++;

temp=temp->link;

}

}

void updateToDo(){

system("cls");

todo \*temp,\*t;

char k;

while(1){

printf("\nWant to add?y/n");

fflush(stdin);

scanf("%c",&k);

if(k=='n')

break;

printf("\nADD it..\n");

temp=(todo \*)calloc(1,sizeof(todo));

fflush(stdin);

gets(temp->data);

temp->link=NULL;

t=start;

while(t->link!=NULL){

t=t->link;

}

t->link=temp;

fixcount();

}

printf("\n\n");

system("pause");

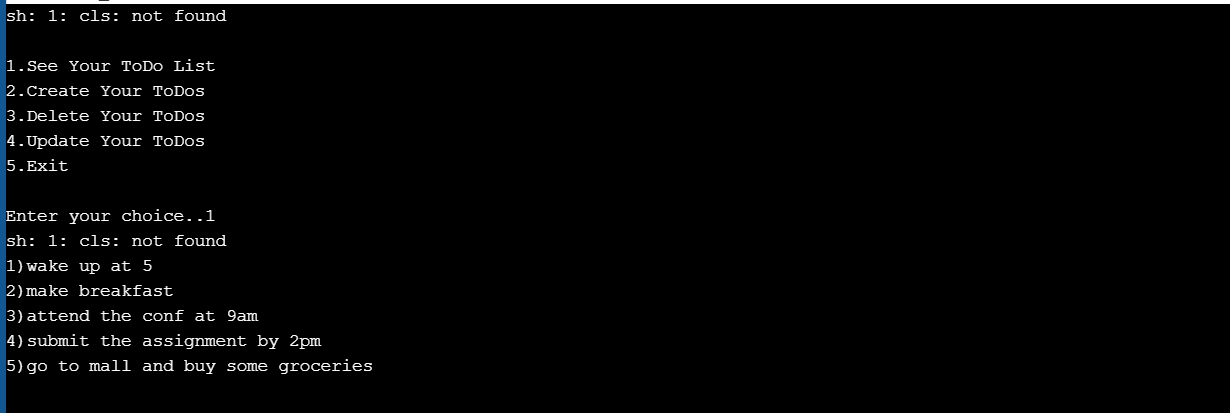
}

**5.OUTPUT/RESULT**

1.create a to do list



2. to view the created list



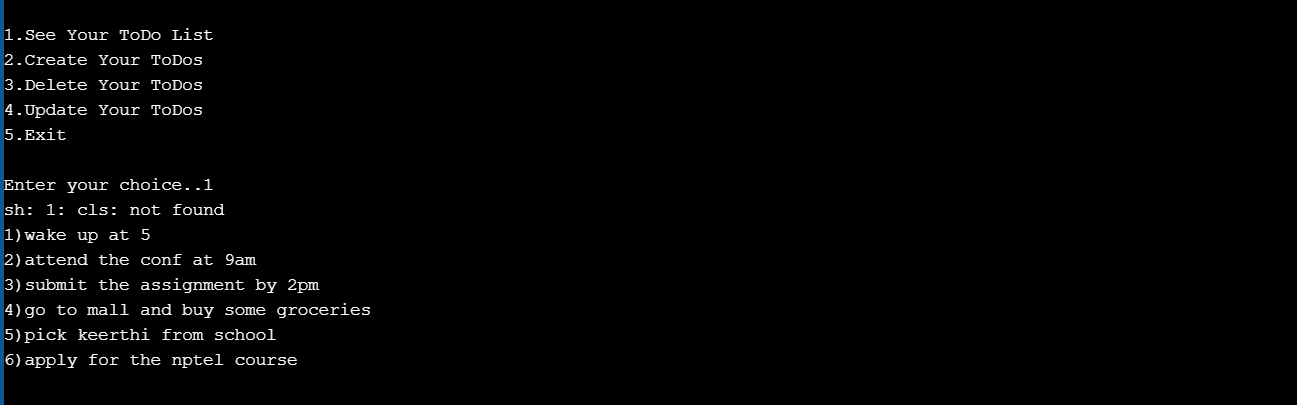
3. To do deletion one of the work in the created list that is to do list and to view after deletion



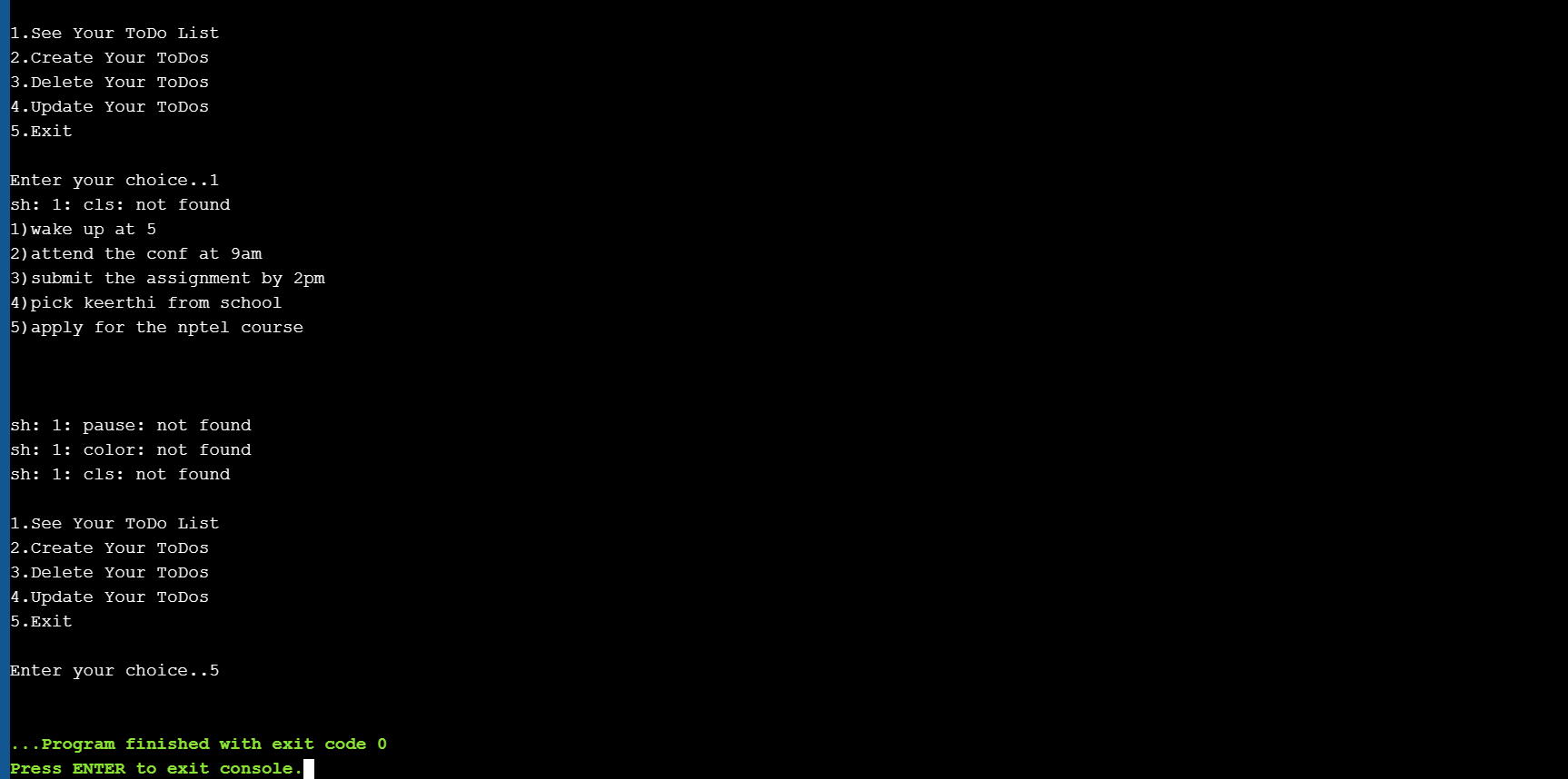
4.to update the to do list



To view it after update



5. exit



So the program is finished.

**6.CONCLUSION**

To conclude, for the people who want to make a list of the things to do in a day this app is very useful and We can del the works which we are not necessary and we can update the list. This provides the people to complete their work.

**7.REFERENCES**

1. Horowitz E, Sahni S and Susan Anderson-Freed, Fundamentals of Data structures in C, 2nd Edition (2008), Universities Press.
2. <http://www.nptel.ac.in/courses/106102064>