

DSA Semester Project

This semester I used lists to create a mini puzzle game. The premise of the game is that you have several books that you need to arrange or delete (depending on the level) so that you get a sequence or are left with specific books in the library.

In level one we need to arrange books so that we get them in alphabetical order in terms of their authors. We give the player flexibility in swapping books places, and we also got two hints of them.

```
Level 1
Book: To_Kill_A_Mockingbird by Harper_Lee ,colour Yellow ,year 1960
Book: War_And_Peace by Leo_Tolstoy ,colour Blue ,year 1869
Book: Crime_And_Punishment by Fyodor_Dostoevsky ,colour Red ,year 1866
Book: 1984 by George_Orwell ,colour Green ,year 1949
Try ordering the books by pressing 1 or if you want a hint you can press 2
```

After they arrange the books in that order, they will be sent to level two where we will add two more books, and we want to remain with books that only have the cover colour a primary colour.

```
You did it! Let me make it a bit harder
Level 2
Book: Crime_And_Punishment by Fyodor_Dostoevsky ,colour Red ,year 1866
Book: 1984 by George_Orwell ,colour Green ,year 1949
Book: To_Kill_A_Mockingbird by Harper_Lee ,colour Yellow ,year 1960
Book: War_And_Peace by Leo_Tolstoy ,colour Blue ,year 1869
Book: Les_Miserables by Victor_Hugo ,colour Orange ,year 1862
Book: The_Master_and_Margarita by Mikhail_Bulgakov ,colour Red ,year 1967
Type 1 to remove a book or press 2 to get a hint
```

If they try to remove books that should not be removed a message will come on screen and the action will be invalidated.

```
Choose the name of the book you want to remove
Crime_And_Punishment
WHAT WAS THAT
Book: Crime_And_Punishment by Fyodor_Dostoevsky ,colour Red ,year 1866
Book: 1984 by George_Orwell ,colour Green ,year 1949
Book: To_Kill_A_Mockingbird by Harper_Lee ,colour Yellow ,year 1960
Book: War_And_Peace by Leo_Tolstoy ,colour Blue ,year 1869
Book: Les_Miserables by Victor_Hugo ,colour Orange ,year 1862
Book: The_Master_and_Margarita by Mikhail_Bulgakov ,colour Red ,year 1967
Type 1 to remove a book or press 2 to get a hint
```

Once they remove the books, they will get to level three. Level three lets you add a book to the list

```
Level 3
This time i will ask you to add a book of your own and i will set the colour
What is the name of the book you want to add
Story
What is your name
Fabian
You might need the internet for this
Book: Crime_And_Punishment by Fyodor_Dostoevsky ,colour Red ,year 1866
Book: To_Kill_A_Mockingbird by Harper_Lee ,colour Yellow ,year 1960
Book: War_And_Peace by Leo_Tolstoy ,colour Blue ,year 1869
Book: The_Master_and_Margarita by Mikhail_Bulgakov ,colour Red ,year 1967
Book: Lolita by Vladimir_Nabokov ,colour Orange ,year 1955
Book: Wuthering_Heights by Emily_Bronte ,colour Blue ,year 1847
Book: Story by Fabian ,colour Indigo ,year 2025
Press 1 to remove a book or press 2 to get a hint
```

Also, with two other books. This level has the same premise as the last but here we need to remove any books that don't have Russian authors except our book.

If we try to remove our new book we will get a message:

```
Your book is too special to deserve this
Book: Crime_And_Punishment by Fyodor_Dostoevsky ,colour Red ,year 1866
Book: To_Kill_A_Mockingbird by Harper_Lee ,colour Yellow ,year 1960
Book: War_And_Peace by Leo_Tolstoy ,colour Blue ,year 1869
Book: The_Master_and_Margarita by Mikhail_Bulgakov ,colour Red ,year 1967
Book: Lolita by Vladimir_Nabokov ,colour Orange ,year 1955
Book: Wuthering_Heights by Emily_Bronte ,colour Blue ,year 1847
Book: Story by Fabian ,colour Indigo ,year 2025
Press 1 to remove a book or press 2 to get a hint
```

One they fulfil the requirements of this level we will get to level four, the last level, in which we need to remove books until there are no two books that share the same specs.

```
Level 4
Book: Crime_And_Punishment by Fyodor_Dostoevsky ,colour Red ,year 1866
Book: War_And_Peace by Leo_Tolstoy ,colour Blue ,year 1869
Book: The_Master_and_Margarita by Mikhail_Bulgakov ,colour Red ,year 1967
Book: Lolita by Vladimir_Nabokov ,colour Orange ,year 1955
Book: Story by Fabian ,colour Indigo ,year 2025
Book: The_Brothers_Karamazov by Fyodor_Dostoevsky ,colour Brown ,year 1879
Book: The_man_whoLaughs by Victor_Hugo ,colour Green ,year 1869
Press 1 to remove a book or press 2 to get a hint
```

So here we have many choices to remove but if we try to remove something that does not have specs that repeat, we will get a message:

```
Choose the name of the book you want to remove
Lolita
AGAIN
Book: Crime_And_Punishment by Fyodor_Dostoevsky ,colour Red ,year 1866
Book: War_And_Peace by Leo_Tolstoy ,colour Blue ,year 1869
Book: The_Master_and_Margarita by Mikhail_Bulgakov ,colour Red ,year 1967
Book: Lolita by Vladimir_Nabokov ,colour Orange ,year 1955
Book: Story by Fabian ,colour Indigo ,year 2025
Book: The_Brothers_Karamazov by Fyodor_Dostoevsky ,colour Brown ,year 1879
Book: The_man_whoLaughs by Victor_Hugo ,colour Green ,year 1869
Press 1 to remove a book or press 2 to get a hint
```

After we remove the last book that shares specs with other books the game will end, and we will get a game over message

```
Choose the name of the book you want to remove
War_And_Peace
Nicee
Game Over...
For now :)...
Process finished with exit code 0
```

CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
//creating the list
typedef struct JOC {
    char name[30];
    char autor[30];
    int year;
    char colour[20];
    struct JOC *next;
}carti;
//set first and last to NULL
carti *first,*last=NULL;
//adding the books for level one
void Adaugare_Unu() {
    carti *temp,*temp2;
    first=(carti *)malloc(sizeof(carti));
    strcpy(first->name, "To_Kill_A_Mockingbird");
    strcpy(first->autor, "Harper_Lee");
```

```

first->year = 1960;
strcpy(first->colour,"Yellow");
temp2=(carti *)malloc(sizeof(carti));
strcpy(temp2->name, "Crime_And_Punishment");
strcpy(temp2->autor, "Fyodor_Dostoevsky");
temp2->year = 1866;
strcpy(temp2->colour,"Red");
temp=(carti *)malloc(sizeof(carti));
strcpy(temp->name, "War_And_Peace");
strcpy(temp->autor, "Leo_Tolstoy");
temp->year = 1869;
strcpy(temp->colour,"Blue");
last=(carti *)malloc(sizeof(carti));
strcpy(last->name, "1984");
strcpy(last->autor, "George_Orwell");
last->year = 1949;
strcpy(last->colour,"Green");
first->next=temp;
temp->next=temp2;
temp2->next=last;
last->next=NULL;
}

//adding the books for level two
void Adaugare_Doi() {
    carti *temp,*temp2;
    temp=(carti *)malloc(sizeof(carti));
    strcpy(temp->name, "Les_Miserables");
    strcpy(temp->autor, "Victor_Hugo");
    temp->year = 1862;
    strcpy(temp->colour,"Orange");
    temp2=(carti *)malloc(sizeof(carti));
    strcpy(temp2->name, "The_Master_and_Margarita");
    strcpy(temp2->autor, "Mikhail_Bulgakov");
    temp2->year = 1967;
    strcpy(temp2->colour,"Red");
    last->next=temp;
    temp->next=temp2;
    temp2->next=NULL;
    last=temp2;
}

```

```

//adding the books for level three
void Adaugare_Trei() {
    carti *temp,*temp2;
    temp=(carti *)malloc(sizeof(carti));
    strcpy(temp->name, "Lolita");
    strcpy(temp->autor, "Vladimir_Nabokov");
    temp->year = 1955;
    strcpy(temp->colour,"Orange");
    temp2=(carti *)malloc(sizeof(carti));
    strcpy(temp2->name, "Wuthering_Heights");
    strcpy(temp2->autor, "Emily_Bronte");
    temp2->year = 1847;
    strcpy(temp2->colour,"Blue");
    last->next=temp;
    temp->next=temp2;
    temp2->next=NULL;
    last=temp2;
}

//adding the books for level four
void Adaugare_Patru() {
    carti *temp,*temp2;
    temp=(carti *)malloc(sizeof(carti));
    strcpy(temp->name, "The_Brothers_Karamazov");
    strcpy(temp->autor, "Fyodor_Dostoevsky");
    temp->year = 1879;
    strcpy(temp->colour,"Brown");
    temp2=(carti *)malloc(sizeof(carti));
    strcpy(temp2->name, "The_man_whoLaughs");
    strcpy(temp2->autor, "Victor_Hugo");
    temp2->year = 1869;
    strcpy(temp2->colour,"Green");
    last->next=temp;
    temp->next=temp2;
    temp2->next=NULL;
    last=temp2;
}

//the function that we use at level one to swap the place of a book
void Schimbare(char Cartea[],int Locul,int NumarCarti) {
    carti *temp,*temp2;
    temp=first;
    if(strcmp(temp->name, Cartea)==0) {

```

```

    temp2=temp;
    first=temp->next;
} else {
    while(strcmp(temp->next->name,Cartea)!=0) {
        temp=temp->next;
    }
    if(temp->next==last) {
        temp2=temp->next;
        temp->next=NULL;
        last=temp;
    } else {
        temp2=temp->next;
        temp->next=temp->next->next;
    }
}
temp=first;
if(Locul!=1 && Locul!=NumarCarti) {
    for(int i=1;i<Locul-1;i++) {
        temp=temp->next;
    }
    temp2->next=temp->next;
    temp->next=temp2;
}
if(Locul==1) {
    temp2->next=first;
    first=temp2;
}
if(Locul==NumarCarti) {
    temp2->next=NULL;
    last->next=temp2;
    last=temp2;
}
}

//the condition for level one where i verify if the books are order alphabetically based on
the authors
int Fals_Unu(int NumarCarti) {
    carti *temp=first;
    for(int i=1;i<NumarCarti;i++) {
        if(strcmp(temp->autor,temp->next->autor)>0)
            return 0;
        temp=temp->next;
    }
}

```

```

    }
    return 1;
}
// level one where i choose the actions and move books around
void Nivelul_Unu(int NumarCarti) {
    printf("Level 1\n");
    int Action, Locul;
    char Cartea[30];
    NumarCarti=4;
    int Hint=0;
    Adaugare_Unu();
    while(Fals_Unu(NumarCarti)==0) {
        carti *current=first;
        while(current!=NULL) {
            printf("Book: %s by %s ,colour %s ,year %d \n",current->name,current->autor,current->colour,current->year);
            current=current->next;
        }
        printf("Try ordering the books by pressing 1 or if you want a hint you can press 2\n");
        scanf("%d",&Action);
        if(Action==1) {
            printf("Choose the name of the book you want to move \n");
            scanf("%s",&Cartea);
            printf("Choose the location where you want to move the book \n");
            scanf("%d",&Locul);
            Schimbare(Cartea,Locul,NumarCarti);
        }else if(Action==2) {
            if(Hint==0) {
                printf("Arrange the books as if the stars themselves aligned in a sequence, guided by the silent rhythm of their makers.\n");
                Hint++;
            }
            else if(Hint==1) {
                printf("Set the books in a line where the whispers of their creators fall in a quiet, ascending harmony of names.\n");
                Hint++;
            }
            else if(Hint==2) {
                printf("No more hints!");
            }
        }
    }
}

```

```

    }
    printf("You did it! Let me make it a bit harder\n");
}
// the condition of level two to see if there are green or orange books
int Fals_Doi() {
    carti *temp=first;
    int Numar=0;
    while(temp!=NULL) {
        if(strcmp(temp->colour,"Orange")==0 || strcmp(temp->colour,"Green")==0) {
            Numar++;
        }
        temp=temp->next;
    }
    if(Numar!=0) {
        return 0;
    }
    return 1;
}
// the delete function
void Stergere(char Cartea[]) {
    carti *temp=first;
    if(strcmp(temp->name, Cartea)==0) {
        first=temp->next;
        free(temp);
    }
    else {
        while(strcmp(temp->next->name, Cartea)!=0) {
            temp=temp->next;
        }
        if(temp->next->next==NULL) {
            free(temp->next);
            temp->next=NULL;
            last=temp;
        }else{
            free(temp->next);
            temp->next=temp->next->next;
        }
    }
}
}

```



```

// here i check if i need to delete a certain book or if a player did a mistake
int Check(char Cartea[]) {
    carti *temp=first;
    while(strcmp(temp->name,Cartea)!=0) {
        temp=temp->next;
    }
    if(strcmp(temp->colour,"Orange")==0 || strcmp(temp->colour,"Green")==0) {
        return 1;
    }
    return 0;
}

// level 2 where i try to get rid of the books that colours are not primary
void Nivelul_Doi() {
    int Action;
    int Hint=0;
    char Cartea[30];
    Adaugare_Doi();
    printf("Level 2\n");
    while(Fals_Doi()==0) {
        carti *current=first;
        while(current!=NULL) {
            printf("Book: %s by %s ,colour %s ,year %d \n",current->name,current-
>autor,current->colour,current->year);
            current=current->next;
        }
        printf("Type 1 to remove a book or press 2 to get a hint\n");
        scanf("%d",&Action);
        if(Action==1) {
            printf("Choose the name of the book you want to remove \n");
            scanf("%s",&Cartea);
            if(Check(Cartea)==1) {
                Stergere(Cartea);
            }
            else {
                printf("WHAT WAS THAT\n");
            }
        }
        if(Action==2) {
            if(Hint==0) {
                printf("Three stand pure, unmixed, unbroken.\nFire's glow, ocean's depth, and
sunlight's token.\nKeep these true, let the rest be forsaken.\n");
            }
        }
    }
}

```

```

        Hint++;
    }
    else if(Hint==1) {
        printf("Three colors stand, no mixes, no blends.\n");
        Hint++;
    }
    else if(Hint==2) {
        printf("Dont make it easy\n");
    }
}
printf("You learn fast!!NEEEXT\n");

}
//here i add the node created by the player at level three
void Adaugare_Nod() {
    carti *temp;
    char Cartea[30];
    char Name[30];
    temp=(carti *)malloc(sizeof(carti));
    printf("What is the name of the book you want to add \n");
    scanf("%s",&Cartea);
    printf("What is your name\n");
    scanf("%s",&Name);
    strcpy(temp->name, Cartea);
    strcpy(temp->autor, Name);
    temp->year=2025;
    strcpy(temp->colour, "Indigo");
    last->next=temp;
    temp->next=NULL;
    last=temp;
}
//here i check if the books have russian writers and need to be or not deleted
int Check_Trei(char Cartea[]) {
    carti *temp=first;
    while(strcmp(temp->name, Cartea)!=0) {
        temp=temp->next;
    }
    if(strcmp(temp->colour, "Indigo")==0) {
        return 3;
    }
}

```

```

    }
    else if(strcmp(temp->autor,"Mikhail_Bulgakov")==0 || strcmp(temp-
>autor,"Fyodor_Dostoevsky")==0 || strcmp(temp->autor,"Leo_Tolstoy")==0 ||
strcmp(temp->autor,"Vladimir_Nabokov")==0) {
        return 2;
    }else {
        return 1;
    }
}
//level tree where i need to remove all books that dont have russian writers exception
makes the book that the player creates
void Nivelul_Trei() {
    int Action;
    int Hint=0;
    int Numar=2;
    char Cartea[30];
    printf("Level 3\n");
    printf("This time i will ask you to add a book of your own and i will set the colour\n");
    Adaugare_Trei();
    Adaugare_Nod();
    printf("You might need the internet for this\n");
    while(Numar!=0) {
        carti *current=first;
        while(current!=NULL) {
            printf("Book: %s by %s ,colour %s ,year %d \n",current->name,current-
>autor,current->colour,current->year);
            current=current->next;
        }
        printf("Press 1 to remove a book or press 2 to get a hint\n");
        scanf("%d",&Action);
        if(Action==1) {

            printf("Choose the name of the book you want to remove \n");
            scanf("%s",&Cartea);
            if(Check_Trei(Cartea)==1) {
                Stergere(Cartea);
                Numar--;
            }
            else if(Check_Trei(Cartea)==2) {
                printf("What are you doing\n");
            }else {

```

```

        printf("Your book is too special to deserve this\n");
    }
}
if(Action==2) {
    if(Hint==0) {
        printf("Born beneath the northern sky,\nTheir words are deep,their thoughts are
high\nIf not from where the cold winds blow\nShould such voices fade,or grow\n");
        Hint++;
    }
    else if(Hint==1) {
        printf("From the land of snow,true wisdom flows\nBooks not from there,to the
fire they go\n");
        Hint++;
    }
    else if(Hint==2) {
        printf("You can do this!\n");
    }
}
}
}

```

//here we check if there are two books that share the same publication year

```

int Check_If_Year() {
    carti *temp=first;
    while(temp->next!=NULL) {
        int Repetari=0;
        carti *temp2=first;
        while(temp2->next!=NULL) {
            if(temp->year==temp2->year) {
                Repetari++;
            }
            temp2=temp2->next;
        }
        if(temp2->year==temp->year) {
            Repetari++;
        }
        if(Repetari==2) {
            return 0;
        }
        temp=temp->next;
    }
    return 1;
}

```

```
}  
//here i check if a certain book shares a publication year with another to see if i delete it  
or not
```

```
int Check_Year_Carte(char Cartea[]) {  
    carti *temp=first;  
    carti *temp2;  
    int Repetari=0;  
    while(temp->next!=NULL) {  
        if(strcmp(temp->name, Cartea)==0) {  
            temp2=temp;  
        }  
        temp=temp->next;  
    }  
    if (strcmp(temp->name, Cartea)==0) {  
        temp2=temp;  
    }  
    temp=first;  
    while(temp!=NULL) {  
        if(temp2->year==temp->year) {  
            Repetari++;  
        }  
        temp=temp->next;  
    }  
    if(Repetari==2) {  
        return 0;  
    }  
    return 1;  
}
```

```
//here we check if there are two books that share the same author
```

```
int Check_If_Autor() {  
    carti *temp=first;  
    while(temp->next!=NULL) {  
        int Repetari=0;  
        carti *temp2=first;  
        while(temp2->next!=NULL) {  
            if(strcmp(temp2->autor, temp->autor)==0) {  
                Repetari++;  
            }  
            temp2=temp2->next;  
        }  
        if(strcmp(temp2->autor, temp->autor)==0) {
```

```

        Repetari++;
    }
    if(Repetari==2) {
        return 0;
    }
    temp=temp->next;
}
return 1;
}

```

//here i check if a certain book shares an author with another to see if i delete it or not

```

int Check_Autor_Carte(char Cartea[]) {
    carti *temp=first;
    carti *temp2;
    int Repetari=0;
    while(temp->next!=NULL) {
        if(strcmp(temp->name,Cartea)==0) {
            temp2=temp;
        }
        temp=temp->next;
    }
    if (strcmp(temp->name,Cartea)==0) {
        temp2=temp;
    }
    temp=first;
    while(temp!=NULL) {
        if(strcmp(temp2->autor,temp->autor)==0) {
            Repetari++;
        }
        temp=temp->next;
    }
    if(Repetari==2) {
        return 0;
    }
    return 1;
}

```

//here we check if there are two books that share the same colour

```

int Check_If_Colour() {
    carti *temp=first;
    while(temp->next!=NULL) {
        int Repetari=0;
        carti *temp2=first;

```

```

while(temp2->next!=NULL) {
    if(strcmp(temp2->colour,temp->colour)==0) {
        Repetari++;
    }
    temp2=temp2->next;
}
if(strcmp(temp2->colour,temp->colour)==0) {
    Repetari++;
}
if(Repetari==2) {
    return 0;
}
temp=temp->next;
}
return 1;
}
//here I check if a certain book shares the colour with another to see if i delete it or not
int Check_Colour_Carte(char Cartea[]) {
    carti *temp=first;
    carti *temp2;
    int Repetari=0;
    while(temp->next!=NULL) {
        if(strcmp(temp->name,Cartea)==0) {
            temp2=temp;
        }
        temp=temp->next;
    }
    if (strcmp(temp->name,Cartea)==0) {
        temp2=temp;
    }
    temp=first;
    while(temp!=NULL) {
        if(strcmp(temp2->colour,temp->colour)==0) {
            Repetari++;
        }
        temp=temp->next;
    }
    if(Repetari==2) {
        return 0;
    }
    return 1;
}

```

```

}
//level four where I delete books untill there are no two books that share the same specs
void Nivelul_Patru() {
    int Action;
    int Hint=0;
    char Cartea[30];
    printf("Level 4\n");
    Adaugare_Patru();
    while(Check_If_Year()==0 || Check_If_Autor()==0 || Check_If_Colour()==0) {
        carti *current=first;
        while(current!=NULL) {
            printf("Book: %s by %s ,colour %s ,year %d \n",current->name,current->
>autor,current->colour,current->year);
            current=current->next;
        }
        printf("Press 1 to remove a book or press 2 to get a hint\n");
        scanf("%d",&Action);
        if(Action==1) {
            printf("Choose the name of the book you want to remove \n");
            scanf("%s",&Cartea);
            if(Check_Year_Carte(Cartea)==0 || Check_Autor_Carte(Cartea)==0 ||
Check_Colour_Carte(Cartea)==0) {
                Stergere(Cartea);
            }else {
                printf("AGAIN\n");
            }
        }
        if(Action==2) {
            if(Hint==0) {
                printf("No two may share their shade, their sire, or their start.\nBreak the bonds,
and set them apart.\n");
                Hint++;
            }
            else if(Hint==1) {
                printf("On the shelf, no pair may dwell—\nIn hue, hand, or time they tell.\nSplit
the echoes, break the chain,\nLet no twin or trace remain.\n");
                Hint++;
            }
            else if(Hint==2) {
                printf("How did you get this far?\n");
            }
        }
    }
}

```



```

    }
}
printf("Nicee\n");
}
//here I have a for which indicates the lvl I am at
int main()
{
    int NumarNivele=5,NumarCarti;
    for(int i=1;i<=NumarNivele;i++){
        if(i==1){
            Nivelul_Unu(NumarCarti);
        }
        else if(i==2){
            Nivelul_Doi();
        }
        else if(i==3){
            Nivelul_Trei();
        }
        else if(i==4){
            Nivelul_Patru();
        }
        else if(i==5){
            printf("Game Over...\nFor now :)...");
        }
    }
}
return 0;
}

```