LOVELY PROFESSIONAL UNIVERSITY

CSE 101 PROJECT

TOPIC – INCOME TAX CALCULATOR

SUBMITTED TO – DR.JYOTI GODARA

SUBMITTED BY: SECTION K22BS

NAME :-GANESH MADDALA

ROLL NO:-44

REG NO :-12213211

SECTION:-K22BS

Project Title

INCOME TAX CALCULATOR

Description

Tax calculation is an important aspect of financial management for individuals and organizations. In many cases, the calculation of tax can be complex, involving various tax brackets and deductions. To manage and calculate tax effectively, computer programs can be used to automate the process and provide accurate results.

In this context, file handling in C provides a simple and effective way to store and manipulate tax records. A struct can be used to represent each tax record, and file operations such as reading, writing, and searching can be used to manage the records. This allows users to add, edit, and delete tax records, as well as generate reports and perform other operations on the data.

By using C programming with file handling, users can manage tax records in a flexible and efficient manner, allowing them to focus on other aspects of their financial management.

MODULES

1.Add New Record:

This module allows the user to add a new record to the file. The user is prompted to enter the name and income of the taxpayer. The program then calculates the tax based on the income using the following tax brackets:

No tax for income up to 250,000.

5% tax for income between 250,001 and 500,000.

20% tax for income between 500,001 and 1,000,000.

30% tax for income above 1,000,000.

The calculated tax is stored in the tax field of the TaxPayer struct before the record is added to the file.

2.List all Tax Payer along with income tax to be paid:

This module allows the user to list all the tax records in the file along with the calculated tax. The program reads each record from the file and prints the name, income, and tax fields of the TaxPayer struct.

3.Search:

This module allows the user to search for a record in the file by name. The user is prompted to enter the name to search for. The program reads each record from the file and checks if the name matches. If a match is found, the program prints the name, income, and tax fields of the TaxPayer struct.

4.Edit:

This module allows the user to edit a record in the file by name. The user is prompted to enter the name to search for. The program reads each record from the file and checks if the name matches. If a match is found, the user is prompted to enter the new income. The program then calculates the new tax based on the new income and updates the tax field of the TaxPayer struct before writing it back to the file.

5.Delete Record:

This module allows the user to delete a record from the file by name. The user is prompted to enter the name to delete. The program reads each record from the file and writes all the records except the one to be deleted to a temporary file. The temporary file is then renamed to the original file name, effectively deleting the record. If the record is not found, the program informs the user that the record was not found.

6.Exit:

This module allows the user to exit the program. It simply prints a message and returns 0 to indicate successful completion of the program.

Overall, this program demonstrates the use of file handling in C to store and manipulate records of tax payers. The program uses a struct to represent each record and calculates the tax based on the income of the taxpayer. It also provides the ability to search, edit, and delete records, giving the user the ability to manage the tax records effectively.

One potential improvement for this program could be to add validation for user input, such as ensuring that the user enters a valid income or name. Another improvement could be to provide the ability to sort the records based on name or income for easier management.

CODES

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>



typedef struct {

    char name[50];

    int age;
```

```
    int income;
    float tax;
} TaxPayer;

void addRecord();
void listRecords();
void searchRecord();
void editRecord();
void deleteRecord();

void addRecord() {
    FILE *fp;
    TaxPayer tp;
    printf("Enter name: ");
    scanf("%s", tp.name);
    printf("Enter age: ");
    scanf("%d", &tp.age);
    printf("Enter income: ");
    scanf("%d", &tp.income);
    if (tp.income <= 250000) {
```

```
        tp.tax = 0;
    } else if (tp.income > 250000 && tp.income <= 500000) {
        tp.tax = 0.05 * (tp.income - 250000);
    } else if (tp.income > 500000 && tp.income <= 1000000) {
        tp.tax = 12500 + 0.2 * (tp.income - 500000);
    } else {
        tp.tax = 112500 + 0.3 * (tp.income - 1000000);
    }
    fp = fopen("tax_records.txt", "ab");
    fwrite(&tp, sizeof(TaxPayer), 1, fp);
    fclose(fp);
    printf("Record added successfully.\n");
}

void listRecords() {
    FILE *fp;
    TaxPayer tp;
    fp = fopen("tax_records.txt", "rb");
    printf("%-20s %-10s %-10s %-10s\n", "Name", "Age", "Income", "Tax");
    while (fread(&tp, sizeof(TaxPayer), 1, fp) == 1) {
```

```
        printf("%-20s %-10d %-10d %-10.2f\n", tp.name, tp.age, tp.income, tp.tax);
    }
    fclose(fp);
}

void searchRecord() {
    FILE *fp;
    TaxPayer tp;
    char name[50];
    int found = 0;
    printf("Enter name to search: ");
    scanf("%s", &name);
    fp = fopen("tax_records.txt", "rb");
    while (fread(&tp, sizeof(TaxPayer), 1, fp) == 1) {
        if (strcmp(name, tp.name) == 0) {
            printf("%-20s %-10d %-10d %-10.2f\n", tp.name, tp.age, tp.income, tp.tax);
            found = 1;
            break;
        }
    }
```

```
    if (found == 0) {
        printf("Record not found.\n");
    }
    fclose(fp);
}

void editRecord() {
    FILE *fp;
    TaxPayer tp;
    char name[50];
    int found = 0;
    printf("Enter name to edit: ");
    scanf("%s", &name);
    fp = fopen("tax_records.txt", "r+b");
    while (fread(&tp, sizeof(TaxPayer), 1, fp) == 1) {
        if (strcmp(name, tp.name) == 0) {
            printf("Enter new name: ");
            scanf("%s", &tp.name);
            printf("Enter new age: ");
            scanf("%d", &tp.age);
```

```
            printf("Enter new income: ");
            scanf("%d", &tp.income);
            if (tp.income <= 250000){
tp.tax = 0;
} else if (tp.income > 250000 && tp.income <= 500000) {
tp.tax = 0.05 * (tp.income - 250000);
} else if (tp.income > 500000 && tp.income <= 1000000) {
tp.tax = 12500 + 0.2 * (tp.income - 500000);
} else {
tp.tax = 112500 + 0.3 * (tp.income - 1000000);
}
fseek(fp, -sizeof(TaxPayer), SEEK_CUR);
fwrite(&tp, sizeof(TaxPayer), 1, fp);
found = 1;
break;
}
}
if (found == 0) {
printf("Record not found.\n");
} else {
```

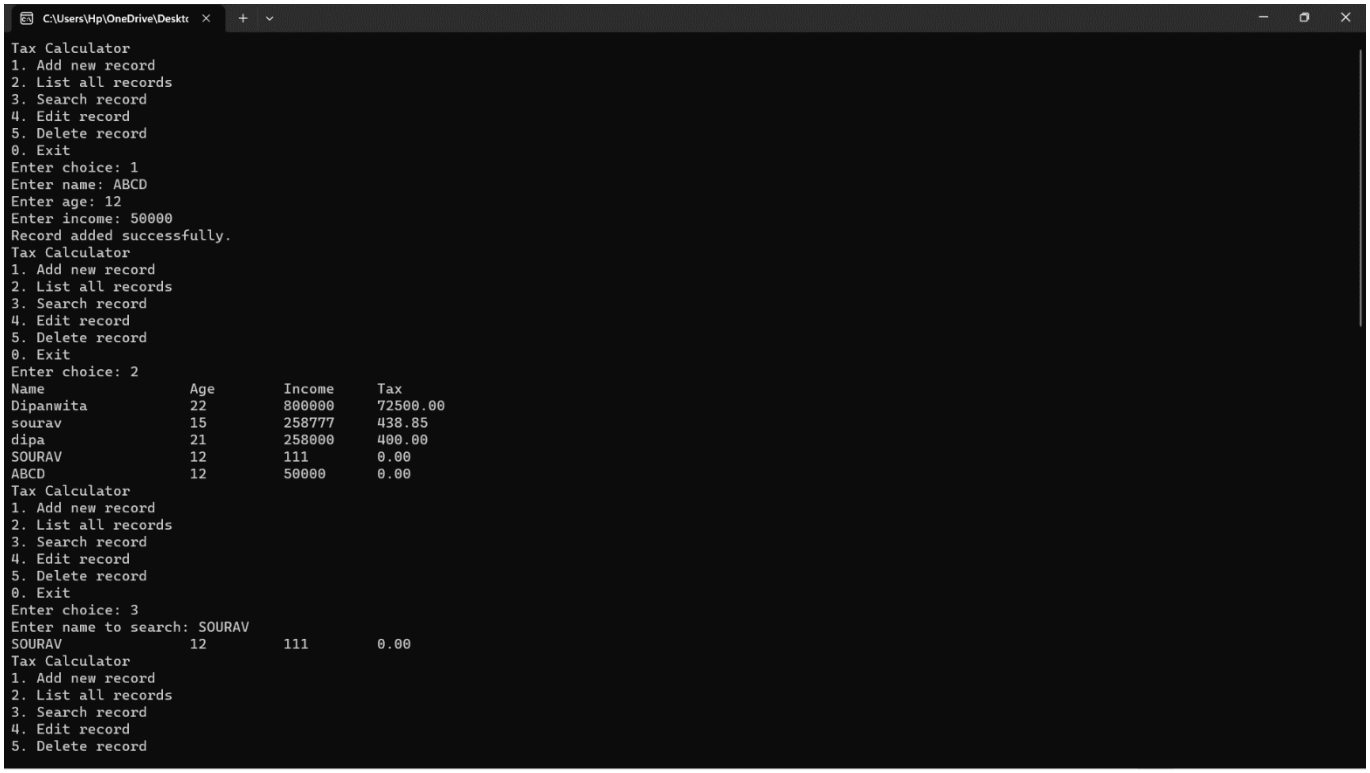
```
printf("Record updated successfully.\n");

}

fclose(fp);

}



void deleteRecord() {

FILE *fp, *tempfp;

TaxPayer tp;

char name[50];

int found = 0;

printf("Enter name to delete: ");

scanf("%s", &name);

fp = fopen("tax_records.txt", "rb");

tempfp = fopen("temp.txt", "wb");

while (fread(&tp, sizeof(TaxPayer), 1, fp) == 1) {

if (strcmp(name, tp.name) != 0) {

fwrite(&tp, sizeof(TaxPayer), 1, tempfp);

} else {

found = 1;

}
```

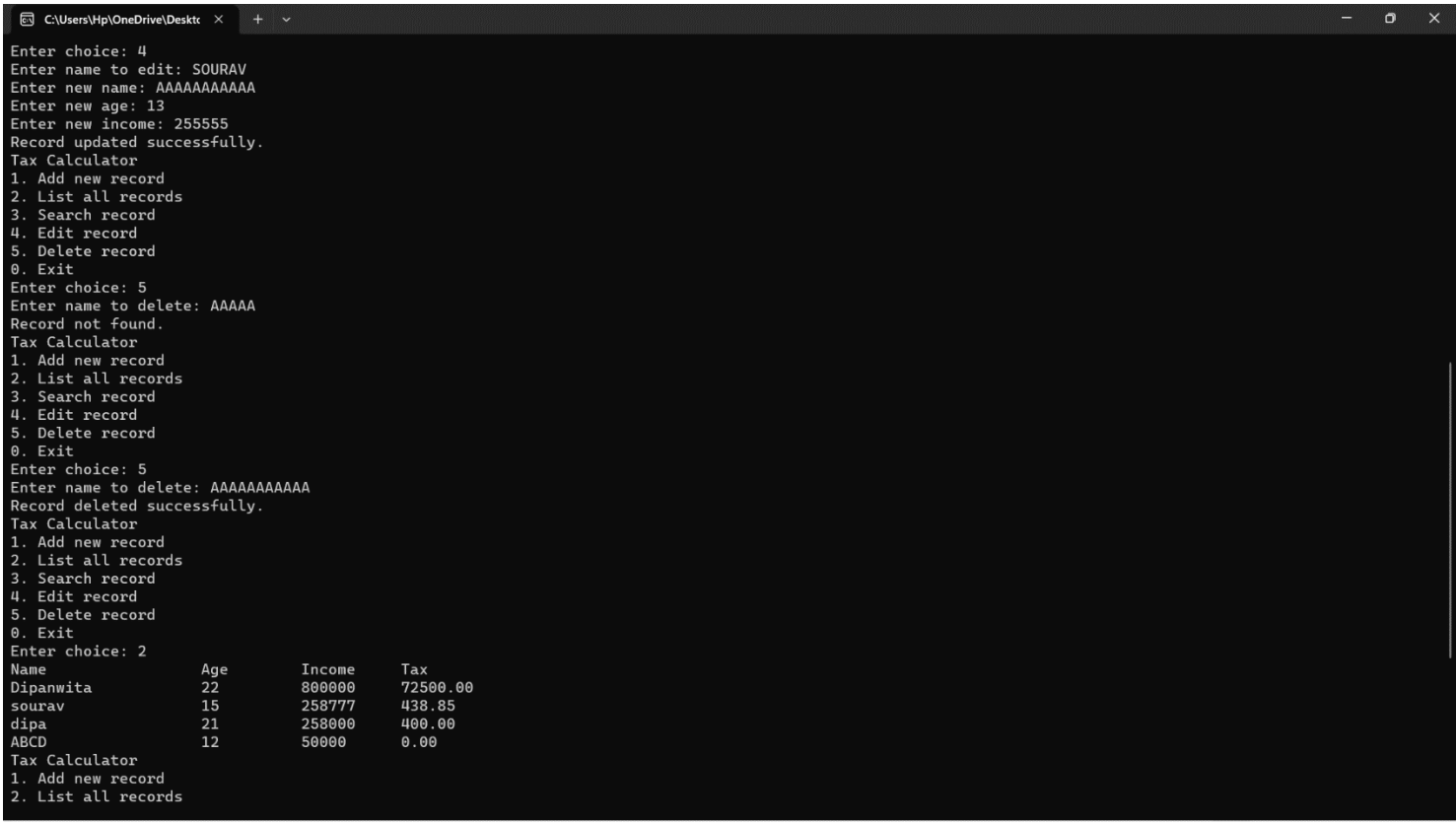
```
}

fclose(fp);

fclose(tempfp);

remove("tax_records.txt");

rename("temp.txt", "tax_records.txt");

if (found == 0) {

printf("Record not found.\n");

} else {

printf("Record deleted successfully.\n");

}

}



int main() {

int choice;

do {

printf("Tax Calculator\n");

printf("1. Add new record\n");

printf("2. List all records\n");

printf("3. Search record\n");

printf("4. Edit record\n");
```

```
printf("5. Delete record\n");

printf("0. Exit\n");

printf("Enter choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

addRecord();

break;

case 2:

listRecords();

break;

case 3:

searchRecord();

break;

case 4:

editRecord();

break;

case 5:

deleteRecord();

break;
```

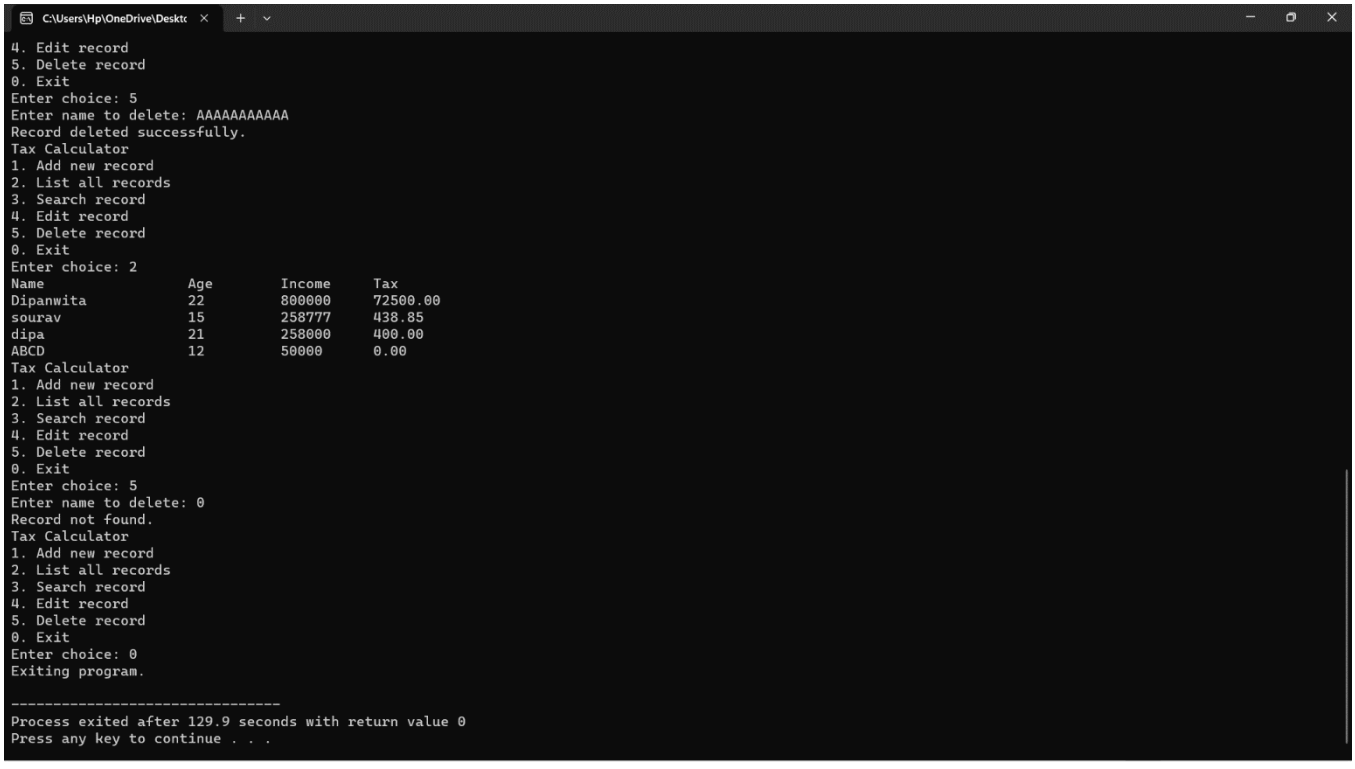
```
case 0:

printf("Exiting program.\n");

break;

default:

printf("Invalid choice. Please try again.\n");

break;

}

} while (choice != 0);

return 0;

}
```

OUTPUTS


C:\Users\Hp\OneDrive\Deskt
Tax Calculator
1. Add new record
2. List all records
3. Search record
4. Edit record
5. Delete record
0. Exit
Enter choice: 1
Enter name: ABCD
Enter age: 12
Enter income: 50000
Record added successfully.
Tax Calculator
1. Add new record
2. List all records
3. Search record
4. Edit record
5. Delete record
0. Exit
Enter choice: 2
Name Age Income Tax
Dipanwita 22 800000 72500.00
sourav 15 258777 438.85
dipa 21 258000 400.00
SOURAV 12 111 0.00
ABCD 12 50000 0.00
Tax Calculator
1. Add new record
2. List all records
3. Search record
4. Edit record
5. Delete record
0. Exit
Enter choice: 3
Enter name to search: SOURAV
SOURAV 12 111 0.00
Tax Calculator
1. Add new record
2. List all records
3. Search record
4. Edit record
5. Delete record

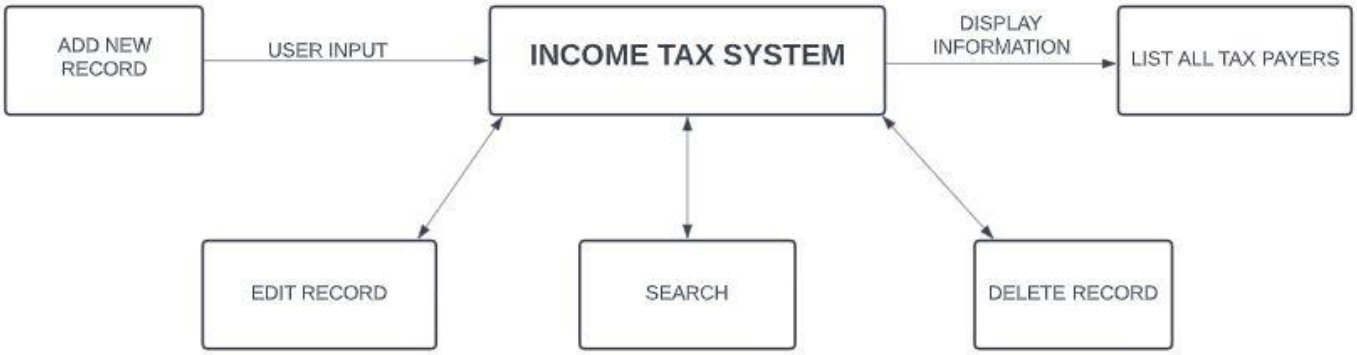


```
C:\Users\Hp\OneDrive\Desktc
Enter choice: 4
Enter name to edit: SOURAV
Enter new name: AAAAAAAAAA
Enter new age: 13
Enter new income: 255555
Record updated successfully.
Tax Calculator
1. Add new record
2. List all records
3. Search record
4. Edit record
5. Delete record
0. Exit
Enter choice: 5
Enter name to delete: AAAAA
Record not found.
Tax Calculator
1. Add new record
2. List all records
3. Search record
4. Edit record
5. Delete record
0. Exit
Enter choice: 5
Enter name to delete: AAAAAAAAAA
Record deleted successfully.
Tax Calculator
1. Add new record
2. List all records
3. Search record
4. Edit record
5. Delete record
0. Exit
Enter choice: 2
Name                 Age         Income      Tax
Dipanwita            22          800000      72500.00
sourav               15          258777      438.85
dipa                 21          258000      400.00
ABCD                 12          50000       0.00
Tax Calculator
1. Add new record
2. List all records
```



```
C:\Users\Hp\OneDrive\Desktc
4. Edit record
5. Delete record
0. Exit
Enter choice: 5
Enter name to delete: AAAAAAAAAA
Record deleted successfully.
Tax Calculator
1. Add new record
2. List all records
3. Search record
4. Edit record
5. Delete record
0. Exit
Enter choice: 2
Name                 Age         Income      Tax
Dipanwita            22          800000      72500.00
sourav               15          258777      438.85
dipa                 21          258000      400.00
ABCD                 12          50000       0.00
Tax Calculator
1. Add new record
2. List all records
3. Search record
4. Edit record
5. Delete record
0. Exit
Enter choice: 5
Enter name to delete: 0
Record not found.
Tax Calculator
1. Add new record
2. List all records
3. Search record
4. Edit record
5. Delete record
0. Exit
Enter choice: 0
Exiting program.

--------------------------------
Process exited after 129.9 seconds with return value 0
Press any key to continue . . .
```

LEVEL 0 DFD


ADD NEW RECORD
USER INPUT
INCOME TAX SYSTEM
DISPLAY INFORMATION
LIST ALL TAX PAYERS
EDIT RECORD
SEARCH
DELETE RECORD