

DSA Project

Simplify Debts

About :

In this project, I have developed a program by which friends can split their expenses and give the final amount of money that a specific friend shall pay to another friend. I have provided the following features in the program:

1. Number of friends and amount of money they have paid
2. Splitting the expenses
3. Simplifying the total expenses

1. Add Number of Friends and expenses

```
void AddData(char names[][name_size], int n, float money[])
{
    printf("\nDo you want to add an expense? (Type '1' if Yes or press any key for No): ");
    int x;
    scanf("%d", &x);

    if (x == 1)
    {
        for (int i = 0; i < n; i++)
        {
            printf("\nType %d for %s ", (i + 1), names[i]);
        }

        printf("\n\nWho paid? : ");
        int p1;
        scanf("%d", &p1);

        printf("How much did he/she pay? : ");
        float amt;
        scanf("%f", &amt);
    }
}
```


2. Split the expenses equally or add it who owes

```
printf("\nDo you want to split this expense equally?(Type 1 for Yes or press any key for No): ");
int ch;
scanf("%d", &ch);

if (ch == 1)
{
    for (int i = 0; i < n; i++)
    {
        if (i == p1 - 1)
        {
            money[i] += amt - (amt / n);
            continue;
        }
        money[i] -= (amt / n);
    }
}
else
{
    printf("Who owes? : ");
    int p2;
    scanf("%d", &p2);

    for (int i = 0; i < n; i++)
    {
        if (i == p1 - 1)
            money[i] += amt;
        if (i == p2 - 1)
            money[i] -= amt;
    }
}
printf("Expense Successfully Added!\n\n");
AddData(names, n, money);
}
else
{
    return;
}
}
```

Outputs of 1. and 2

```
-----  
| WELCOME TO SIMPLIFY DEBTS PROGRAM |  
-----
```

```
Enter the number of people in the group: 3  
Enter the names of the people in the group (seperated with a space): Man Mihir Mulya  
Do you want to add an expense? (Type '1' if Yes or press any key for No): 1  
  
Type 1 for Man  
Type 2 for Mihir  
Type 3 for Mulya  
  
Who paid? : 1  
How much did he/she pay? : 1000  
  
Do you want to split this expense equally?(Type 1 for Yes or press any key for No): 1  
Expense Successfully Added!  
Do you want to add an expense? (Type '1' if Yes or press any key for No): 1  
  
Type 1 for Man  
Type 2 for Mihir  
Type 3 for Mulya  
  
Who paid? : 1  
How much did he/she pay? : 500  
  
Do you want to split this expense equally?(Type 1 for Yes or press any key for No): 0  
Who owes? : 3  
Expense Successfully Added!
```


3.Simplify the total expenses among all

```
void Simplify(char names[][name_size], int n, float money[])
{
    int min = minOf(money, n);
    int max = maxOf(money, n);

    if (money[min] == 0 && money[max] == 0)
        return;
    float small_amt = (0 - money[min]) < money[max] ? (0 - money[min]) : money[max];

    if (small_amt == 0.000000)
        return;
    money[min] += small_amt;
    money[max] -= small_amt;

    printf("--> %s pays %f to %s\n", names[min], small_amt, names[max]);
    Simplify(names, n, money);
}
```

4.Functions for simplifying the expenses

```
int minOf(float money[], int n)
{
    int min = 0;
    for (int i = 0; i < n; i++)
    {
        if (money[i] < money[min])
            min = i;
    }
    return min;
}
```

```
int maxOf(float money[], int n)
{
    int max = 0;
    for (int i = 0; i < n; i++)
    {
        if (money[i] > money[max])
            max = i;
    }
    return max;
}
```


Final Output

```
Do you want to add an expense? (Type '1' if Yes or press any key for No): 0
```

```
Type 1 to display final transaction list: 1
```

```
--> Man pays 500.000000 to Mulya
```

```
*****Program Terminated*****
```


Code

```
int name_size = 100;

void AddData(char names[][name_size], int n, float money[])
{
    printf("\nDo you want to add an expense? (Type '1' if Yes or press any key for No): ");
    int x;
    scanf("%d", &x);

    if (x == 1)
    {
        for (int i = 0; i < n; i++)
        {
            printf("\nType %d for %s ", (i + 1), names[i]);

            printf("\n\nWho paid? : ");
            int p1;
            scanf("%d", &p1);

            printf("How much did he/she pay? : ");
            float amt;
            scanf("%f", &amt);
```

```
printf("\nDo you want to split this expense equally?(Type 1 for Yes or press any key for No): ");
int ch;
scanf("%d", &ch);

if (ch == 1)
{
    for (int i = 0; i < n; i++)
    {
        if (i == p1 - 1)
        {
            money[i] += amt - (amt / n);
            continue;
        }
        money[i] -= (amt / n);
    }
}
else
{
    printf("Who owes? : ");
    int p2;
    scanf("%d", &p2);
}
```



```
        for (int i = 0; i < n; i++)
        {
            if (i == p1 - 1)
                money[i] += amt;
            if (i == p2 - 1)
                money[i] -= amt;
        }
    }
    printf("Expense Successfully Added!\n\n");
    AddData(names, n, money);
}
else
{
    return;
}
}
```

```
int minOf(float money[], int n)
{
    int min = 0;
    for (int i = 0; i < n; i++)
    {
```

```
        if (money[i] < money[min])
            min = i;
    }
    return min;
}
```

```
int maxOf(float money[], int n)
{
    int max = 0;
    for (int i = 0; i < n; i++)
    {
        if (money[i] > money[max])
            max = i;
    }
    return max;
}
```

```
void Simplify(char names[][name_size], int n, float money[])
{
    int min = minOf(money, n);
    int max = maxOf(money, n);

    if (money[min] == 0 && money[max] == 0)
```



```
float small_amt = (0 - money[min]) < money[max] ? (0 - money[min]) : money[max];
```

```
if (small_amt == 0.000000)
```

```
    return;
```

```
money[min] += small_amt;
```

```
money[max] -= small_amt;
```

```
printf("--> %s pays %f to %s\n", names[min], small_amt, names[max]);
```

```
Simplify(names, n, money);
```

```
}
```

```
int main()
```

```
{
```

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

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

```
printf("\t\t|                WELCOME TO SIMPLIFY DEBTS PROGRAM                |\n");
```

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

```
//printf("\n\t\t\t*****Welcome to Simplify Debts program*****\n\n");
```

```
printf("Enter the number of people in the group: ");
```

```
int n = 0;
```

```
scanf("%d", &n);
```

```
float money[n];
```

```
    float small_amt;
```

```
printf("\t\t-----\n");
printf("\t\t|          WELCOME TO SIMPLIFY DEBTS PROGRAM          |\n");
printf("\t\t-----\n\n");
//printf("\n\t\t\t*****Welcome to Simplify Debts program*****\n\n");
printf("Enter the number of people in the group: ");
int n = 0;
scanf("%d", &n);
float money[n];
char names[n][100];
printf("\nEnter the names of the people in the group (seperated with a space): ");
for (int i = 0; i < n; i++)
    scanf("%s", names[i]);
AddData(names, n, money);
printf("\nType 1 to display final transaction list: ");
int ch;
scanf("%d", &ch);
if (ch == 1)
{
    Simplify(names, n, money);
}
printf("\n\n*****Program Terminated*****");
return 0;
```


THANK YOU!