

- **First create a group of n friends -> enter n (total numbers of friends)**
now take input of name of n friends and assign index to every friend.

n=4

1- Rajat

2- Sourabh

3- Aditya

4- Ramya

- **1 for Adding transaction**
- **2 for showing balances**
- **3 for showing cashflow**
- **-1 for exit from splitwise**

1->

enter paid amount,

enter 1 for equally distribution in all members

enter 2 for equally distribution in selective friends

enter 3 for unequally distribuion

1.1->

- distribution of amount equally

1.2->

- display index of each friend
- enter indexes of frined
- distibutte amount in input indexes

1.3->

- Enter amount for rajat
- enter amount for sourabh
- enter amount for aditya
- enter amount for ramya
- check total = paid amount

2->

Display balances ->

Enter 1 for all balances

Enter 2 for selective balances

3-> Showing cashflow from one friend to another friend.

```

#include<stdio.h>

struct Pair {
    double first;
    char* second;
};

int main()
{
    printf("Enter Number of friends in your group: ");
    int n;
    scanf("%d",&n);
    double exp[n];
    for(int i = 0;i<n;i++)
    {
        exp[i] = 0.0;
    }
    char names[n][100];
    printf("Enter %d names of your friend circle:\n", n);
    for(int i = 0;i<n;i++)
    {
        scanf("%s",names[i]);
    }
    while(1)
    {
        printf("Press 1 for adding a transaction\n");
        printf("Press 2 for showing expenses.\n");
        printf("Press 3 for showing cashflow.\n");
        printf("Press -1 for exit from Splitwise\n");
        int choice;
        scanf("%d", &choice);
        if(choice==-1)
        {
            printf("Thanks for using Splitwise!!!\n");
            break;
        }
        if(choice==1)
        {
            printf("Enter Total Amount paid: ");
            double amount;
            scanf("%lf", &amount);

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

```

```

    }

    int paid_name;
    printf("Who paid(Enter Index of your friend): ");
    scanf("%d", &paid_name);

    exp[paid_name-1]+=amount;
    while(1)
    {
        printf("Press 1 for Equal Distribution.\n");
        printf("Press 2 for Equal Selective Distribution.\n");
        printf("Press 3 for Unequal Distribution.\n");
        printf("Press -1 for exit.\n");

        int Dis_choice;
        scanf("%d", &Dis_choice);
        if(Dis_choice== -1)
        {
            break;
        }
        if(Dis_choice==1)
        {
            double bal = amount/((double)n);
            for(int i = 0;i<n;i++)
            {
                exp[i]-=bal;
            }
            break;
        }
        if(Dis_choice==2)
        {
            for (int i = 0; i < n; i++)
            {
                printf("%d - %s\n", i + 1, names[i]);
            }

            int sel_n;
            printf("Enter number of selective friends: ");
            scanf("%d", &sel_n);

            double bal = amount/((double)sel_n);
            printf("Enter names Indexes of selective friends (only enter
Index of your friend): ");
            for(int i = 0;i<sel_n;i++)
            {

```

```

        int x;
        scanf("%d", &x);
        exp[x-1]-=bal;
    }
    break;
}
if(Dis_choice==3)
{
    double total = 0;
    int v[n];
    double x;
    for(int i = 0;i<n;i++)
    {
        printf("Enter Expenses for %s: ", names[i]);
        scanf("%lf", &x);
        v[i] = x;
        total += x;
    }
    if(total==amount)
    {
        for(int i = 0;i<n;i++)
        {
            exp[i]-=v[i];
        }
        break;
    }
    else
    {
        exp[paid_name-1]-=amount;
        printf("Total distributed amount does not equal to the
paid amount, Enter again\n");
        continue;
    }
}
printf("Invalid Choice, choose again!!!\n");
}
continue;
}
if(choice==2)
{
    while(1)
    {
        printf("Press 1 for Showing All friends balances\n");
        printf("Press 2 for Showing Selective friends balances\n");
    }
}

```

```

        int display_choice;
        scanf("%d", &display_choice);

        if(display_choice==1)
        {
            for(int i = 0;i<n;i++)
            {
                printf("%s : %.2lf\n", names[i], exp[i]);
            }
            break;
        }
        if(display_choice==2)
        {
            for(int i = 0;i<n;i++)
            {
                printf("%d - %s\n", i + 1, names[i]);
            }
            printf("Enter index number of selective friends and press -1
for exit: ");

            while(1)
            {
                int x;
                scanf("%d", &x);
                if(x== -1)
                {
                    break;
                }
                if(x>n)
                {
                    printf("Wrong index, you do not have %d friends\n",
x);

                    continue;
                }
                printf("%s : %.2lf\n", names[x - 1], exp[x - 1]);
            }
            break;
        }
        printf("Invalid Choice, choose again!!!\n");
    }
    continue;
}
if(choice==3)
{
    int neg_cnt = 0, pos_cnt = 0;
    for(int i = 0;i<n;i++)

```

```

{
    if(exp[i]<0)
    {
        neg_cnt++;
    }
    if(exp[i]>0)
    {
        pos_cnt++;
    }
}
if(neg_cnt==0 || pos_cnt==0)
{
    printf("All Settled.\n");
    continue;
}
struct Pair neg[neg_cnt],pos[pos_cnt];
int j = 0,k = 0;
for(int i = 0;i<n;i++)
{
    if(exp[i]<0.0)
    {
        neg[j].first = exp[i];
        neg[j].second = names[i];
        j++;
    }
    if(exp[i]>0.0)
    {
        pos[k].first = exp[i];
        pos[k].second = names[i];
        k++;
    }
}
int i = 0;j = 0;
while (i<neg_cnt && j<pos_cnt)
{
    double v = -1*neg[i].first;
    if (v == pos[j].first) {
        printf("%s Owes %s %.2lf\n", neg[i].second, pos[j].second,
pos[j].first);
        neg[i].first = 0;
        pos[j].first = 0;
        i++;
        j++;
        continue;
    }
}

```

```

        if (v < pos[j].first) {
            printf("%s Owes %s %.2lf\n", neg[i].second, pos[j].second,
v);
            pos[j].first -= v;
            neg[i].first = 0;
            i++;
            continue;
        }

        if (v > pos[j].first) {
            printf("%s Owes %s %.2lf\n", neg[i].second, pos[j].second,
pos[j].first);
            neg[i].first += pos[j].first;
            pos[j].first = 0;
            j++;
            continue;
        }
    }
    continue;
}
printf("Invalid Choice, Choose again!!\n");
}
}

```