```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct {
    double amount;
    char category[100];
} transaction;

typedef struct {
    transaction transactions[100];
    int count;
    double income;
    double expenses;
} budgettra;

budgettra* budegetvar() {
    budgettra* tracker = (budgettra*)malloc(sizeof(budgettra));
    if (tracker) {
        tracker->count = 0;
        tracker->income = 0.0;
        tracker->expenses = 0.0;
    }
    return tracker;
}

void income(budgettra* tracker, double amt) {
    tracker->income += amt;
    tracker->transactions[tracker->count].amount = amt;
    strcpy(tracker->transactions[tracker->count].category, "Income");
    tracker->count++;
}

void expense(budgettra* tracker, double amt, const char* cat) {
    tracker->expenses += amt;
    tracker->transactions[tracker->count].amount = amt;
    strcpy(tracker->transactions[tracker->count].category, cat);
    tracker->count++;
}

double expenses(const budgettra* tracker) {
    return tracker->expenses;
}

double balance(const budgettra* tracker) {
    return tracker->income - tracker->expenses;
}
```

```
void gettrans(const budgettra* tracker) {
    printf("**** Transactions ****\n");
    for (int i = 0; i < tracker->count; i++) {
        if (strcmp(tracker->transactions[i].category, "Income") != 0) {
            printf("Category: %s\n", tracker->transactions[i].category);
            printf("Amount: Rs.%.2f\n", tracker->transactions[i].amount);
            printf("\n");
        }
    }
    printf("_________________\n");
}

void getcat(const budgettra* tracker) {
    printf("**** Category ****\n");

    double bal = balance(tracker);
    if (bal >= 0) {
        printf("Category: Balance, Amount: Rs.%.2f\n", bal);
    } else {
        printf("Category: Limit Exceeded\n");
    }
    printf("_________________\n");

    for (int i = 0; i < tracker->count; i++) {
        if (strcmp(tracker->transactions[i].category, "Income") != 0 &&
strcmp(tracker->transactions[i].category, "Balance") != 0) {
            double totalAmount = 0.0;
            int processed = 0;
            for (int j = 0; j < i; j++) {
                if (strcmp(tracker->transactions[i].category, tracker-
>transactions[j].category) == 0) {
                    processed = 1;
                    break;
                }
            }
            if (processed)
            continue;

            for (int j = 0; j < tracker->count; j++) {
                if (strcmp(tracker->transactions[i].category, tracker-
>transactions[j].category) == 0) {
                    totalAmount += tracker->transactions[j].amount;
                }
            }
            printf("Category: %s, Total Amount: Rs.%.2f\n", tracker-
>transactions[i].category, totalAmount);
        }
    }
```

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

void freebud(budgettra* tracker) {
    free(tracker);
}

int main() {
    budgettra* tracker = budegetvar();
    if (!tracker) {

        return 1;
    }

    double amount;
    char category[100];

    printf("Enter Income Amount: Rs.");
    scanf("%lf", &amount);
    income(tracker, amount);

    int choice;
    do {
        printf("\n **** Menu ****\n");
        printf("1. Add Expense\n");
        printf("2. View Transactions\n");
        printf("3. View Total Expenses\n");
        printf("4. View Category Summary\n");
        printf("5. View Balance\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter Expense Amount: Rs.");
                scanf("%lf", &amount);
                printf("Enter Expense Category: ");
                scanf("%s", category);
                expense(tracker, amount, category);
                break;
            case 2:
                gettrans(tracker);
                break;
            case 3:
                if (balance(tracker) < 0) {
                    printf("Limit Exceeded\n");
```

```
                } else {
                    printf("Total Expenses: Rs.%.2f\n", expenses(tracker));
                }
                break;
            case 4:
                getcat(tracker);
                break;
            case 5:
                if (balance(tracker) < 0) {
                    printf("Limit Exceeded\n");
                } else {
                    printf("Balance: Rs.%.2f\n", balance(tracker));
                }
                break;
            case 6:
                printf("Exiting\n");
                break;
            default:
                printf("Invalid choice.\n");
        }
    } while (choice != 6);

    freebud(tracker);

    return 0;
}
```