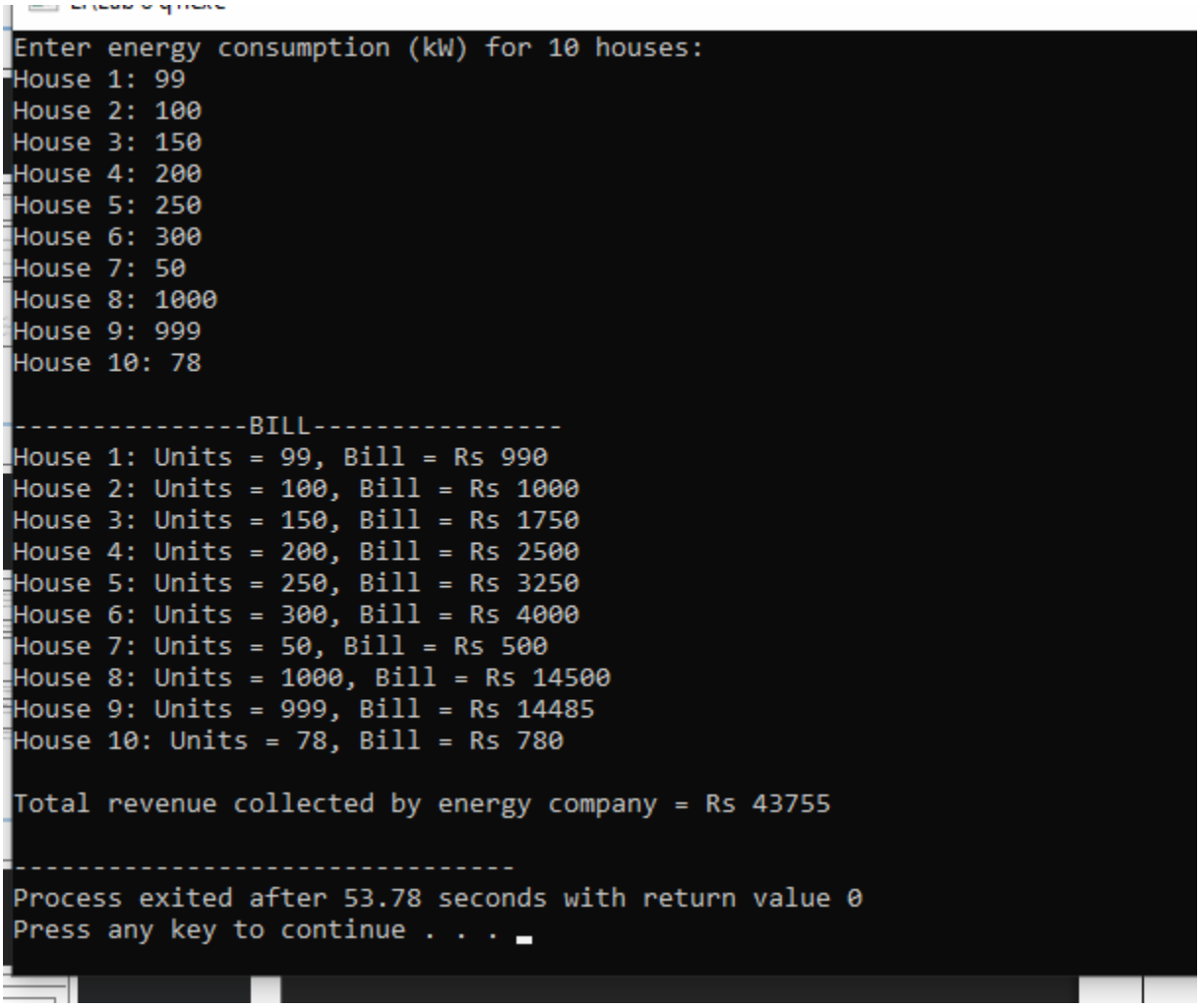
TASK 1:

```c
#include <stdio.h>

int bill_for_units(int units) {
    int bill = 0;
    if (units <= 100) {
        bill = units * 10;
    } else {
        bill = 100 * 10 + (units - 100) * 15;
    }
    return bill;
}

int main() {
    int i;
    int house = 10;
    int consumption[10];
    int *p = consumption;
    int totalRevenue = 0;

    printf("Enter energy consumption (kW) for 10 houses:\n");
    for (i = 0; i < house; i++) {
        printf("House %d: ", i+1);
        scanf("%d", p + i);

    }

    printf("\n---------------BILL----------------\n");
    for (i = 0; i < house; i++) {
        int units = *(p + i);
        int bill = bill_for_units(units);
        printf("House %d: Units = %d, Bill = Rs %d\n", i+1, units, bill);
        totalRevenue += bill;
    }

    printf("\nTotal revenue collected by energy company = Rs %d\n", totalRevenue);
    return 0;
}
```

OUTPUT 1:


Enter energy consumption (kW) for 10 houses:
House 1: 99
House 2: 100
House 3: 150
House 4: 200
House 5: 250
House 6: 300
House 7: 50
House 8: 1000
House 9: 999
House 10: 78
----------BILL----------
House 1: Units = 99, Bill = Rs 990
House 2: Units = 100, Bill = Rs 1000
House 3: Units = 150, Bill = Rs 1750
House 4: Units = 200, Bill = Rs 2500
House 5: Units = 250, Bill = Rs 3250
House 6: Units = 300, Bill = Rs 4000
House 7: Units = 50, Bill = Rs 500
House 8: Units = 1000, Bill = Rs 14500
House 9: Units = 999, Bill = Rs 14485
House 10: Units = 78, Bill = Rs 780
Total revenue collected by energy company = Rs 43755
--------------------------------
Process exited after 53.78 seconds with return value 0
Press any key to continue . . .

TASK 2:

```c
#include <stdio.h>

float maxtemp(float temps[], int n) {
    int i;
    float max = temps[0];
    for (i = 1; i < n; i++) {
        if (temps[i] > max) max = temps[i];
    }
    return max;
}

int main() {
    int patients = 5, read = 4;
    float temps[patients][read];
    int i, j;
    float *p = &temps[0][0];

    printf("Enter %d readings for each of %d patients (in Fahrenheit):\n", read, patients);
    for (i = 0; i < patients; i++) {
        for (j = 0; j < read; j++) {
            printf("Patient %d, reading %d: ", i+1, j+1);
            scanf("%f", (p + i*read + j));
        }
    }

    printf("\nHighest temperature for each patient:\n");
    for (i = 0; i < patients; i++) {
        float highest = maxtemp(&temps[i][0], read);
        printf("Patient %d: Highest = %.2f F", i+1, highest);
        if (highest > 101.0) {
            printf(" ALERT: above 101 F");
        }
        printf("\n");
    }

    return 0;
}
```

Output 2:

```
Patient 1, reading 2: 99
Patient 1, reading 3: 98
Patient 1, reading 4: 90
Patient 2, reading 1: 101
Patient 2, reading 2: 101
Patient 2, reading 3: 122
Patient 2, reading 4: 96
Patient 3, reading 1: 100
Patient 3, reading 2: 97
Patient 3, reading 3: 98
Patient 3, reading 4: 99
Patient 4, reading 1: 78
Patient 4, reading 2: 100
Patient 4, reading 3: 89
Patient 4, reading 4: 88
Patient 5, reading 1: 98
Patient 5, reading 2: 97
Patient 5, reading 3: 96
Patient 5, reading 4: 95

Highest temperature for each patient:
Patient 1: Highest = 99.00 F
Patient 2: Highest = 122.00 F <-- ALERT: above 101 F
Patient 3: Highest = 100.00 F
Patient 4: Highest = 100.00 F
Patient 5: Highest = 98.00 F

--------------------------------
Process exited after 60.43 seconds with return value 0
Press any key to continue . . .
```

Task 3:

```c
#include <stdio.h>

int STUDENTS = 5, SUBJECTS = 3;

float average_marks(int *marks_row, int subjects) {
    int i;
    int sum = 0;
    for (i = 0; i < subjects; i++) {
        sum += *(marks_row + i);
    }
    return (float)sum / subjects;
}
char grade_from_avg(float avg) {
    if (avg >= 80) return 'A';
    if (avg >= 70) return 'B';
    if (avg >= 60) return 'C';
    if (avg >= 50) return 'D';
    return 'F';
}

int topper_index(int marks[STUDENTS][SUBJECTS]) {
    int i, j;
    float bestAvg = -1.0;
    int bestIdx = 0;
    for (i = 0; i < STUDENTS; i++) {
        float avg = average_marks(marks[i], SUBJECTS);
        if (avg > bestAvg) {
            bestAvg = avg;
            bestIdx = i;
        }
    }
    return bestIdx;
}
D
int main() {
    int STUDENTS = 5, SUBJECTS = 3;
    int marks[STUDENTS][SUBJECTS];
```

```c
int main() {
    int STUDENTS = 5, SUBJECTS = 3;
    int marks[STUDENTS][SUBJECTS];
    int i, j;
    printf("Enter marks (out of 100) for %d students and %d subjects:\n", STUDENTS, SUBJECTS);
    for (i = 0; i < STUDENTS; i++) {
        printf("Student %d:\n", i+1);
        for (j = 0; j < SUBJECTS; j++) {
            printf(" Subject %d: ", j+1);
            scanf("%d", &marks[i][j]);
        }
    }

    printf("\nStudent averages and grades:\n");
    for (i = 0; i < STUDENTS; i++) {
        float avg = average_marks(marks[i], SUBJECTS);
        char g = grade_from_avg(avg);
        printf("Student %d: Average = %.2f, Grade = %c\n", i+1, avg, g);
    }

    int (*get_topper)(int [STUDENTS][SUBJECTS]) = topper_index;
    int top = get_topper(marks);
    printf("\nTopper is Student %d\n", top + 1);

    return 0;
}
```

Output 3:

```
Enter marks (out of 100) for 5 students and 3 subjects:
Student 1:
 Subject 1: 77
 Subject 2: 88
 Subject 3: 99
Student 2:
 Subject 1: 99
 Subject 2: 88
 Subject 3: 77
Student 3:
 Subject 1: 44
 Subject 2: 55
 Subject 3: 66
Student 4:
 Subject 1: 65
 Subject 2: 89
 Subject 3: 85
Student 5:
 Subject 1: 39
 Subject 2: 49
 Subject 3: 50

Student averages and grades:
Student 1: Average = 88.00, Grade = A
Student 2: Average = 88.00, Grade = A
Student 3: Average = 55.00, Grade = D
Student 4: Average = 79.67, Grade = B
Student 5: Average = 46.00, Grade = F

Topper is Student 1

--------------------------------
Process exited after 32.75 seconds with return value 0
Press any key to continue . . .
```