

MINI PROJECT

PROBLEM STATEMENT:

C program that calculates and keeps track of all the products purchased from a restaurant weekly. It should be able to tell-

How many customers visited in a week

How many customers visited per day

Mean of no. of customers that visited

Total revenue generated in a week

Total revenue generated per day

Mean of total revenue generated per day

On which day the most revenue was generated

Given that the menu is as follows, ['Dish' (price)]

1- Crispy Calamari Rings (400)

2- Prawn Pie (700)

3- Sticky Toffee Pudding (600)

4- Yorkshire Lamb Patties (900)

5- Roast Lamb Salad (800)

6- Stuffed Jacket Potatoes (300)

7- Paneer Steak (500)

CODE:

```
#include <stdio.h>

int main () {
    int menu_prices[] = {400, 700, 600, 900, 800, 300, 500};
    int num_customers[7] = {0};
    int revenue [7] = {0};
    int total_customers = 0;
    int total_revenue = 0;
    int max_revenue_day = 0;

    printf("*****MENU*****\n\n");
    printf("1 - Crispy Calamari Rings      ₹400\n");
    printf("2 - Prawn Pie                       ₹700\n");
    printf("3 - Sticky Toffee Pudding           ₹600\n");
    printf("4 - Yorkshire Lamb Patties         ₹900\n");
    printf("5 - Roast Lamb Salad                ₹800\n");
    printf("6 - Stuffed Jacket Potatoes        ₹300\n");
    printf("7 - Paneer Steak                   ₹500\n");
    printf("\n");

    for (int day = 1; day <= 7; day++) {
        printf("\nEnter the number of customers on day %d: ", day);
        scanf("%d", &num_customers[day-1]);

        for (int i = 0; i < num_customers[day-1]; i++) {
            printf("\nEnter the dish ordered by customer %d (1-7):", i+1);
            int dish;
            scanf("%d", &dish);
            revenue[day-1] += menu_prices[dish-1];
        }
        total_customers += num_customers[day-1];
        total_revenue += revenue[day-1];
        if (revenue[day-1] > revenue[max_revenue_day]) {
            max_revenue_day = day-1;
        }
    }
}
```

```

    }
}
printf("\nStatics:\n");
printf("Total number of customers in the week: %d\n\n", total_customers);
printf("Average number of customers per day:  %.2f\n\n", (float)
total_customers / 7);
printf("Total revenue generated in the week:  %d\n\n", total_revenue);
printf("Average revenue generated per day:  %.2f\n\n", (float)
total_revenue / 7);
printf("Day with the most revenue:           Day %d\n\n",
max_revenue_day+1);

return 0;
}

```

OUTPUT:

1. The menu appears along the price and corresponding digit for reference as program initiates.

Output

Clear

```

/tmp/mw7zpH00BX.o
*****MENU*****

1 - Crispy Calamari Rings      ₹400
2 - Prawn Pie                  ₹700
3 - Sticky Toffee Pudding      ₹600
4 - Yorkshire Lamb Patties     ₹900
5 - Roast Lamb Salad           ₹800
6 - Stuffed Jacket Potatoes    ₹300
7 - Paneer Steak               ₹500

Enter the number of customers on day 1:

```

2. Then the program asks to input total customers that visited on that day. Along with the dishes ordered by each.

Output Clear

```
/tmp/oQwnvaJQ0b.o
*****MENU*****

1 - Crispy Calamari Rings      ₹400
2 - Prawn Pie                  ₹700
3 - Sticky Toffee Pudding      ₹600
4 - Yorkshire Lamb Patties     ₹900
5 - Roast Lamb Salad           ₹800
6 - Stuffed Jacket Potatoes    ₹300
7 - Paneer Steak               ₹500

Enter the number of customers on day 1: 4
Enter the dish ordered by customer 1 (1-7):5
Enter the dish ordered by customer 2 (1-7):2
Enter the dish ordered by customer 3 (1-7):3
Enter the dish ordered by customer 4 (1-7):6
```

3. Same is repeated for all 7 days of the week.

Output Clear

```
Enter the number of customers on day 2: 6
Enter the dish ordered by customer 1 (1-7):4
Enter the dish ordered by customer 2 (1-7):1
Enter the dish ordered by customer 3 (1-7):2
Enter the dish ordered by customer 4 (1-7):5
Enter the dish ordered by customer 5 (1-7):6
Enter the dish ordered by customer 6 (1-7):3
Enter the number of customers on day 3: 5
Enter the dish ordered by customer 1 (1-7):7
Enter the dish ordered by customer 2 (1-7):4
Enter the dish ordered by customer 3 (1-7):1
Enter the dish ordered by customer 4 (1-7):2
Enter the dish ordered by customer 5 (1-7):3
Enter the number of customers on day 4: 6
Enter the dish ordered by customer 1 (1-7):5
Enter the dish ordered by customer 2 (1-7):2
Enter the dish ordered by customer 3 (1-7):1
Enter the dish ordered by customer 4 (1-7):4
Enter the dish ordered by customer 5 (1-7):5
Enter the dish ordered by customer 6 (1-7):2
Enter the number of customers on day 5: 4
Enter the dish ordered by customer 1 (1-7):7
6Enter the dish ordered by customer 2 (1-7):5
Enter the dish ordered by customer 3 (1-7):6
Enter the dish ordered by customer 4 (1-7):3
Enter the number of customers on day 6: 5
Enter the dish ordered by customer 1 (1-7):2
Enter the dish ordered by customer 2 (1-7):1
Enter the dish ordered by customer 3 (1-7):4
Enter the dish ordered by customer 4 (1-7):3
Enter the dish ordered by customer 5 (1-7):6
Enter the number of customers on day 7: 7
Enter the dish ordered by customer 1 (1-7):7
Enter the dish ordered by customer 2 (1-7):5
Enter the dish ordered by customer 3 (1-7):3
Enter the dish ordered by customer 4 (1-7):1
Enter the dish ordered by customer 5 (1-7):4
Enter the dish ordered by customer 6 (1-7):5
Enter the dish ordered by customer 7 (1-7):6
```

4. Then the final statistics are given by the code.

```
Output
Enter the number of customers on day 1: 7
Enter the dish ordered by customer 1 (1-7):7
Enter the dish ordered by customer 2 (1-7):5
Enter the dish ordered by customer 3 (1-7):3
Enter the dish ordered by customer 4 (1-7):1
Enter the dish ordered by customer 5 (1-7):4
Enter the dish ordered by customer 6 (1-7):5
Enter the dish ordered by customer 7 (1-7):6
Statics:
Total number of customers in the week: 37

Average number of customers per day:    5.29

Total revenue generated in the week:    22900

Average revenue generated per day:      3271.43

Day with the most revenue:              Day 4
```