Updated IPO Chat

| Input | Process | Output |
| --- | --- | --- |
| password  id  snum  name  service | Control Structure:   * For loop - For (record = 1 to tot\_cust STEP 1) * Validation Checks-   + Pretest while statement: Denies access if password is incorrect.   + Pretest While statement: Fname and Lname are at least 3 char long.   + Pretest While Statement - Service not in [ 1, 2, 3, 4, 5,6 ]     - Nested CASE STRUCTURE: service = 1, 2, 3, 4, 5 or 6   + Pretest While Statement- snum < 1 or snum > 3   Sequential IF Statement - to determine if the station number selected is of full capacity.  Nested While Post Test Loop - to validate that station number selected is not the same as the one previously entered.   * CASE Structure: snum = 1, 2, or 3 * Nested IF Statement: To display service selected.   Calculation:  CASE- Total sales and customers per station:   * tot\_station1 = tot\_station1 + price * cNum1 = cNum1+ 1 * tot\_station2 = tot\_station2 + price * cNum2 = cNum2 + 1 * tot\_station3 = tot\_station3 + price * cNum3 = cNum3 + 1   Commission per station for the day:   * commision1 = tot\_station1 - (0.05 \* tot\_station1 + 500) * commision2 = tot\_station2 - (0.05 \* tot\_station2 + 500) * commision3 = tot\_station3 - (0.05 \* tot\_station3 + 500)   Total Sales & customers for the day:   * statEarn = tot\_station1 + tot\_station2 + tot\_station3 * totC = cNum1 + cNum2 + cNum3   Total Revenue   * totCommission = commission1 + commission2 + commission3 * rent = (0.05 \* statEarn) + (500 \* 3) * rev = totCommission+ rent   Accumulates minutes:  minutes = minutes + 30 | “Customer Receipt”  “Customer Name:”  “Customer ID:”  “Invalid Station”  “Station Number: ”  “Invalid Name”  “Enter customer name: ”  “Service not Offered”  “Service: ”  Service Selected:  “Men’s Haircut”  “Men’s Head Shave”  “Children’s Haircut”  “Children’s Head Shave”  “Beard/Mustache Lineup”  “Eyebrow Shave”  “Station is at full capacity. Please select a different station number:”  “Customers who selected 'Eyebrow Shave'”  “Station 1 Customers”  “Station 2 Customers”  “Station 3 Customers”  Total sales per station for the day:   * tot\_station1 * tot\_station2 * tot\_station3   Total customers per station for the day:   * cNum1 * cNum2 * cNum3   Commission per station for the day:   * commission1 * commission2 * commission3   Total day’s sales:   * statEarn   Total customers for the day:   * totC   Total revenue for the day:   * rev |

**Pseudocode**

START

1. DECLARE name[tot\_cust][50] as STRING
2. DECLARE snum[tot\_cust], service[tot\_cust], minutes, cNum1, cNum2, cNum3, totC, password[1], record, id[tot\_cust] as INTEGER
3. DECLARE tot\_station1, tot\_station2, tot\_station3, commission1, commission2, commission3, statEarn, rev, totCommission, rent, price as REAL
4. DECLARE tot\_cust =15 as CONSTANT INTEGER
5. cNum1 = 0
6. cNum2 = 0
7. cNum3 = 0
8. For (record = 0 to tot\_cust STEP 1)
   1. PRINT “Enter the password: ”
   2. READ password
   3. WHILE password != “14309” THEN
      1. PRINT “Invalid password. Please enter password:”
      2. READ password
   4. ENDWHILE
   5. PRINT “Please enter Customer ID: ”
   6. READ id[record]
   7. PRINT “Please enter name: ”
   8. READ name[record]
   9. WHILE length(name[record]) <3
      1. PRINT “Invalid Name”
      2. PRINT "Enter at least 3 letters or more. Renter name please: "
      3. READ name[record]
   10. ENDWHILE
   11. PRINT “1 = Men’s Haircut - $1000\n 2 = Men’s Head Shave - $1300\n 3 = Children’s Haircut - $600\n 4 = Children’s Head Shave - $900\n 5 = Beard/Mustache Lineup - $650\n 6 = Eyebrow Shave - $400”
   12. PRINT “Please enter the service (1,2,3,4,5,6): ”
   13. READ service[record]
   14. WHILE service[record] != 1 AND service[record] != 2 AND service[record] != 3 AND service[record] != 4 AND service[record] != 5 AND service[record] != 6
       1. PRINT “Service not Offered”
       2. PRINT “Please enter service: ”
       3. READ service[record]
   15. ENDWHILE
   16. DOCASE service[record]
       1. CASE 1
          1. price = 1000.00
       2. CASE 2
          1. price = 1300.00
       3. CASE 3
          1. price = 600.00
       4. CASE 4
          1. price = 900.00
       5. CASE 5
          1. price = 650.00
       6. CASE 6
          1. price = 400.00
       7. CASE OTHER
          1. PRINT “Service not offered”
   17. ENDCASE
   18. PRINT “Please enter the station number (1, 2, 3): ”
   19. READ snum[record]
   20. WHILE snum[record] < 1 OR snum[record] > 3
       1. PRINT “Invalid Station. Please enter station Number (1, 2, 3): ”
       2. READ snum[record]
   21. ENDWHILE
   22. IF snum[record] ==1 AND cNum1>=5 THEN
       1. DO
          1. PRINT "Station 1 is at full capacity. Please select a different station number: "
          2. READ snum[record]
       2. WHILE snum[record] ==1
   23. ENDIF
   24. IF snum[record] == 2 AND cNum2>=5 THEN
       1. DO
          1. PRINT "Station 2 is at full capacity. Please select a different station number: "
          2. READ snum[record]
       2. WHILE snum[record] ==2
   25. ENDIF
   26. IF snum[record] == 3 AND cNum3>=5 THEN
       1. DO
          1. PRINT "Station 3 is at full capacity. Please select a different station number: "
          2. READ snum[record]
       2. WHILE snum[record] ==3
   27. ENDIF
   28. DOCASE snum[record]
       1. CASE 1
          1. cNum1 = cNum1 + 1
          2. tot\_station1 = tot\_station1 + price

commission1 = tot\_station1 - (0.05 \* tot\_station1 + 500)

* + 1. CASE 2
       1. cNum2 = cNum2 + 1
       2. tot\_station2 = tot\_station2 + price

commission2 = tot\_station2 - (0.05 \* tot\_station2 + 500)

* + 1. CASE 3
       1. cNum3 = cNum3 + 1
       2. tot\_station3 = tot\_station3 + price

commission3 = tot\_station3 - (0.05 \* tot\_station3 + 500)

* 1. ENDCASE
  2. PRINT “Customer’s Receipt”
  3. PRINT “Customer’s Name: ”, name[record]
  4. PRINT “Customer ID:”, id[record]
  5. PRINT “Service Selected: ”, service[record]
  6. IF service[record] == ‘1’ THEN
     1. PRINT “Men’s Haircut”
  7. ELSE
     1. IF service[record] == ‘2’
        1. PRINT “Men’s Head Shave”
     2. ELSE
        1. IF service[record] == ‘3’
           1. PRINT “Children’s Haircut”
        2. ELSE
           1. IF service[record] == ‘4’

PRINT “Children’s Head Shave”

* + - * 1. ELSE

IF service[record] == ‘5’

PRINT “Beard/Mustache Lineup”

ELSE

IF service[record] == ‘6’

PRINT “Eyebrow Shave”

ENDIF

ENDIF

ENDIF

ENDIF

ENDIF

ENDIF

* 1. PRINT “Station Number: ”, snum[record]
  2. PRINT “Cost: $”, price

1. ENDFOR
2. statEarn = tot\_station1 + tot\_station2 + tot\_station3
3. totC = cNum1 + cNum2 + cNum3
4. totCommission = commission1 + commission2 + commission3
5. rent = (0.05 \* statEarn) + (500 \* 3)
6. rev = totCommission+ rent
7. PRINT “"Customers who selected 'Eyebrow Shave'"
8. PRINT “Customer ID Station Number Customer Name"
9. FOR record = 0 TO tot\_cust STEP 1
   1. IF service[record] == ‘6’ THEN
      1. PRINT “Customer ID: ”, id[record]
      2. PRINT “Customer Name: ”, name[record]
      3. PRINT “Station Number: ”, snum[record]
   2. ENDIF
10. PRINT “Station 1 Customers”
11. FOR record = 0 TO tot\_cust STEP 1
    1. IF snum[record] == 1 THEN
       1. PRINT “Customer Name: ”, name[record]
    2. ENDIF
12. ENDFOR
13. PRINT “ Station 2 Customers”
14. FOR record = 0 TO tot\_cust STEP 1
    1. IF snum[record] == 2 THEN
       1. PRINT “Customer Name: ”, name[record]
    2. ENDIF
15. ENDFOR
16. PRINT “ Station 3 Customers”
17. FOR record = 0 TO tot\_cust STEP 1
    1. IF snum[record] == 3 THEN
       1. PRINT “Customer Name: ”, name[record]
    2. ENDIF
18. ENDFOR
19. PRINT “Total sales per station for the day: ”
20. PRINT “Station 1 - $”, tot\_station1
21. PRINT “Station 2 - $”, tot\_station2
22. PRINT “Station 3 - $”, tot\_station3
23. PRINT “Total customers per station for the day: ”
24. PRINT “Station 1: ” cNum1
25. PRINT “Station 2: ” cNum2
26. PRINT “Station 3: ” cNum3
27. PRINT “Commission for station 1 for the day: $”,commission1
28. PRINT “Commission for station 2 for the day: $”,commission2
29. PRINT “Commission for station 3 for the day: $”,commission3
30. PRINT “Total Day’s Sales: $”, statEarn
31. PRINT “Total customers for the day: ”, totC
32. PRINT “Revenue for the day: $”, rev

STOP

**Chevannese Ellis and James Cameron C Code**

/\*Authors: Chevannese Ellis and James Cameron

Date: December 16, 2023

Description: A digital system to help manage daily operations at a barber shop.

\*/

#include <stdio.h>

#include <string.h>

#define tot\_cust 15

int main()

{

char name[tot\_cust][50];

char password[1];

int id[tot\_cust];

int service[tot\_cust];

int snum[tot\_cust],minutes,cNum1,cNum2,cNum3,totC,record;

double tot\_station1, tot\_station2, tot\_station3, commission1, commission2, commission3, statEarn, rev, rent, totCommission, price;

//Initializes the variables that are used within and outside of loop

cNum1 = 0;

cNum2 = 0;

cNum3 = 0;

tot\_station1= 0;

tot\_station2 = 0;

tot\_station3 = 0;

rev=0;

for (record = 0;record <tot\_cust;record++) //Stimulates the operation of a barbershop

{

printf("Enter the password: ");

scanf("%s",&password);

while (strcmp(password,"14309")!=0)

{

printf("Invalid password\nPlease enter password: ");

scanf("%s",&password);

}

printf("Please enter Customer ID: ");

scanf("%d",&id[record]);

printf("Please enter name: ");

scanf("%s",name[record]);

while (strlen(name[record]) <3) //Loop that validates if name is at least three characters long

{

printf("Enter at least 3 letters or more\nRenter name please: ");

scanf("%s",&name[record]);

}

//Displays menu of babershop services to user and prompts she or he to enter a code which reads the service inputed

printf("1 = Men's Haircut - $1000\n2 = Men's Head Shave - $1300\n3 = Children's Haircut - $600\n4 = Children's Head Shave - $900\n5 = Beard/Mustache Lineup - $650\n6 = Eyebrow Shave - $400");

printf("\nPlease enter the service (1, 2, 3, 4, 5, 6): ");

scanf("%d",&service[record]);

//Loop that validates that service inputed is based on the barbershop menu

while (service[record] != 1 && service[record] != 2 && service[record] != 3 && service[record] != 4 && service[record] != 5 && service[record] != 6)

{

printf("Please reread menu\nEnter the service (1, 2, 3, 4, 5, 6): ");

scanf("%d",&service[record]);

}

//Case Structure is used to assign a value to price based on service selected

switch (service[record])

{

case 1:

price = 1000.00;

break;

case 2:

price = 1300.00;

break;

case 3:

price = 600.00;

break;

case 4:

price = 900.00;

break;

case 5:

price = 650.00;

break;

case 6:

price = 400.00;

break;

default:

printf("Service NOT OFFERED\n");

}

printf("Please enter the station number (1,2,3): "); //Prompts user to enter a station number

scanf("%d",&snum[record]); //Reads the value of station number

while (snum[record] <1 || snum[record] >3) //Validates that station number is within the range of 1,2, or 3

{

printf("Invalid Station. Please enter station number: ");

scanf("%d",&snum[record]);

}

//Validation #1 check to see if Station 1 is at full capacity

if (snum[record] ==1 && cNum1 >=5)

{

do

{

printf("Station 1 is at full capacity. Please select a different station number: ");

scanf("%d",&snum[record]);

}while (snum[record]==1);

}

//Validation #1 check to see if Station 2 is at full capacity

if (snum[record] ==2 && cNum2 >=5)

{

do

{

printf("Station 2 is at full capacity. Please select a different station number: ");

scanf("%d",&snum[record]);

}while (snum[record]==2);

}

//Validation #1 check to see if Station 3 is at full capacity

if (snum[record] ==3 && cNum3 >=5)

{

do

{

printf("Station 3 is at full capacity. Please select a different station number: ");

scanf("%d",&snum[record]);

}while (snum[record]==3);

}

//Validation #2 check to see if Station 1 is at full capacity

if (snum[record] ==1 && cNum1 >=5)

{

do

{

printf("Station 1 is at full capacity. Please select a different station number: ");

scanf("%d",&snum[record]);

}while (snum[record]==1);

}

//Validation #2 check to see if Station 2 is at full capacity

if (snum[record] ==2 && cNum2 >=5)

{

do

{

printf("Station 2 is at full capacity. Please select a different station number: ");

scanf("%d",&snum[record]);

}while (snum[record]==2);

}

//Validation #2 check to see if Station 3 is at full capacity

if (snum[record] ==3 && cNum3 >=5)

{

do

{

printf("Station 3 is at full capacity. Please select a different station number: ");

scanf("%d",&snum[record]);

}while (snum[record]==3);

}

//Case Structure is used to count each station number selected and accumulate the total price of each station

switch (snum[record])

{

case 1:

tot\_station1 = tot\_station1 + price;

cNum1 = cNum1 + 1;

commission1 = tot\_station1 - (0.05 \* tot\_station1 + 500);

break;

case 2:

tot\_station2 = tot\_station2 + price;

cNum2 = cNum2 + 1;

commission2 = tot\_station2 - (0.05 \* tot\_station2 + 500);

break;

case 3:

tot\_station3 = tot\_station3 + price;

cNum3 = cNum3 + 1;

commission3 = tot\_station3 - (0.05 \* tot\_station3 + 500);

break;

default:

printf ("Invalid Station. Please enter station Number (1, 2, 3): ");

}

//Displays Customer Receipt

printf("\n");

printf("Customer Receipt");

printf("\nCustomer Name: %s",name[record]);

printf("\nCustomer ID: %d",id[record]);

printf("\nService Selected: %d,",service[record]);

//IF Statement is used to display the service selected based on code inputted by user

if (service[record] == 2)

{

printf(" Men's Head Shave\n");

}

else

{

if (service[record] == 3)

{

printf(" Children's Haircut\n");

}

else

{

if (service[record] == 4)

{

printf(" Children's Head Shave\n");

}

else

{

if (service[record] == 5)

{

printf(" Beard/Mustache Lineup\n");

}

else

{

if (service[record] == 1)

{

printf(" Men's Haircut\n");

}

else

{

if (service[record] == 6)

{

printf(" Eyebrow Shave\n");

}

}

}

}

}

}

printf("Cost: $%.f\n",price);

printf("Station Number: %d\n\n",snum[record]);

}

//Calculations are done after the For Loop is terminated

statEarn = tot\_station1 + tot\_station2 + tot\_station3;

totC = cNum1 + cNum2 + cNum3;

totCommission = commission1 + commission2 + commission3;

rent = (0.05 \* statEarn) + (500 \* 3);

rev = totCommission + rent;

//Displays the information once the system reaches 15 customers

printf("Customers who selected 'Eyebrow Shave'\n"); //Displays all customers that selected Eyebrow Shave

printf("Customer ID Station Number Customer Name\n");

for (record = 0;record <tot\_cust; record++)

{

if (service[record] == 6)

{

printf(" %d %d %s\n",id[record],snum[record],name[record]);

}

}

printf("\nStation 1 Customers\n"); //Displays all the names of customers that selected station 1

for (record = 0;record<tot\_cust; record++)

{

if (snum[record] == 1)

{

printf(" %s\n",name[record]);

}

}

printf("\nStation 2 Customers\n"); //Displays all the names of customers that selected station 2

for (record = 0;record<tot\_cust; record++)

{

if (snum[record] == 2)

{

printf(" %s\n",name[record]);

}

}

printf("\nStation 3 Customers\n"); //Displays all the names of customers that selected station 3

for (record = 0;record<tot\_cust; record++)

{

if (snum[record] == 3)

{

printf(" %s\n",name[record]);

}

}

//Displays the totals generated from the operation of the barbershop for the day

printf("\nTotal sales per station for the day\nStation 1 - $%.f\nStation 2 - $%.f\nStation 3 - $%.f\n",tot\_station1,tot\_station2,tot\_station3);

printf("Total customers per station for the day\nStation 1: %d\nStation 2: %d\nStation 3: %d\n",cNum1,cNum2,cNum3);

printf("Commissions for the day\nStation 1 Commission: $%.1f\nStation 2 Commission: $%.1f\nStation 3 Commission: $%.1f",commission1,commission2,commission3);

printf("\nTotal Day's Sales: $%.f",statEarn);

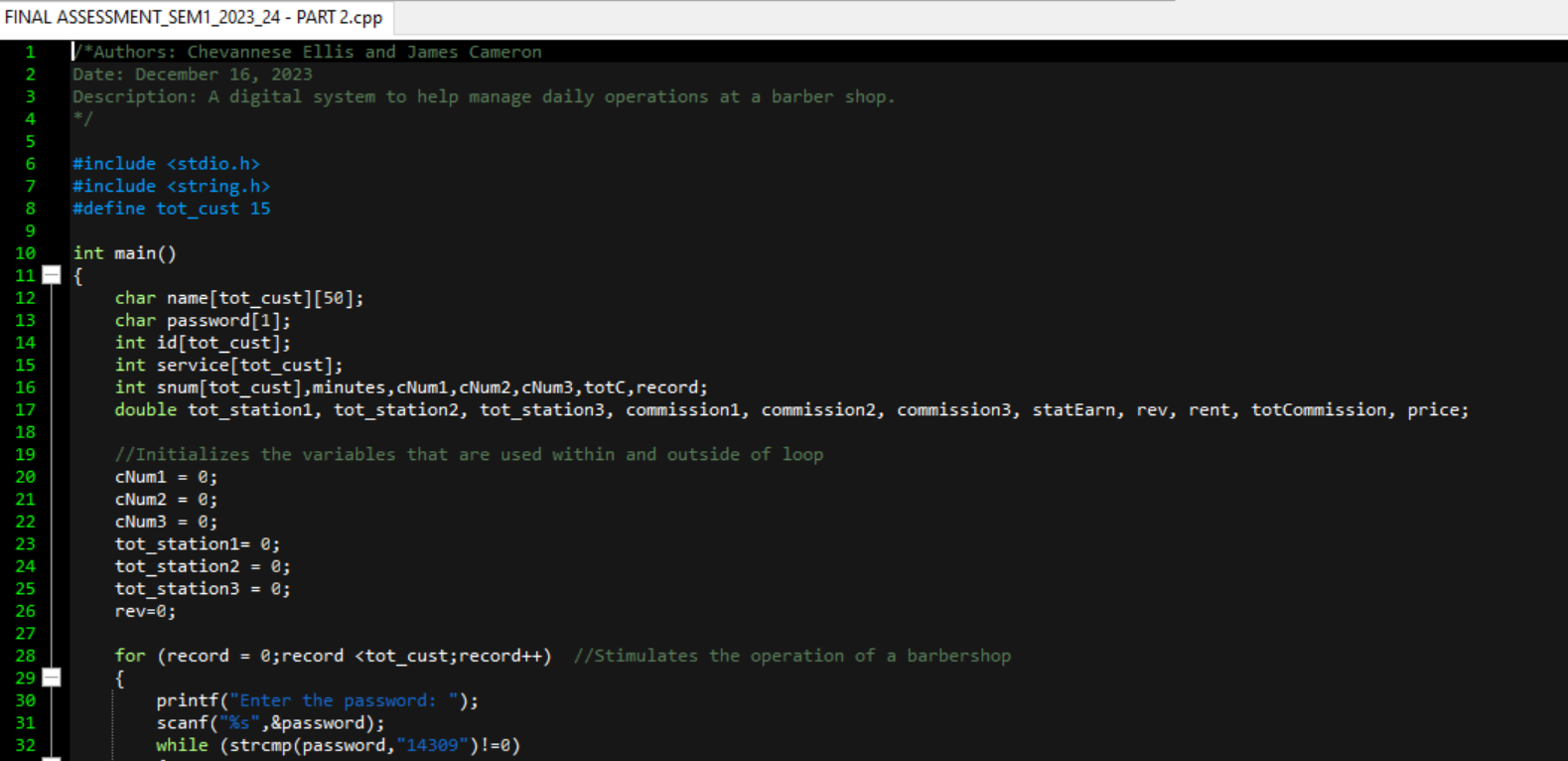
printf("\nTotal customers for the day: %d",totC);

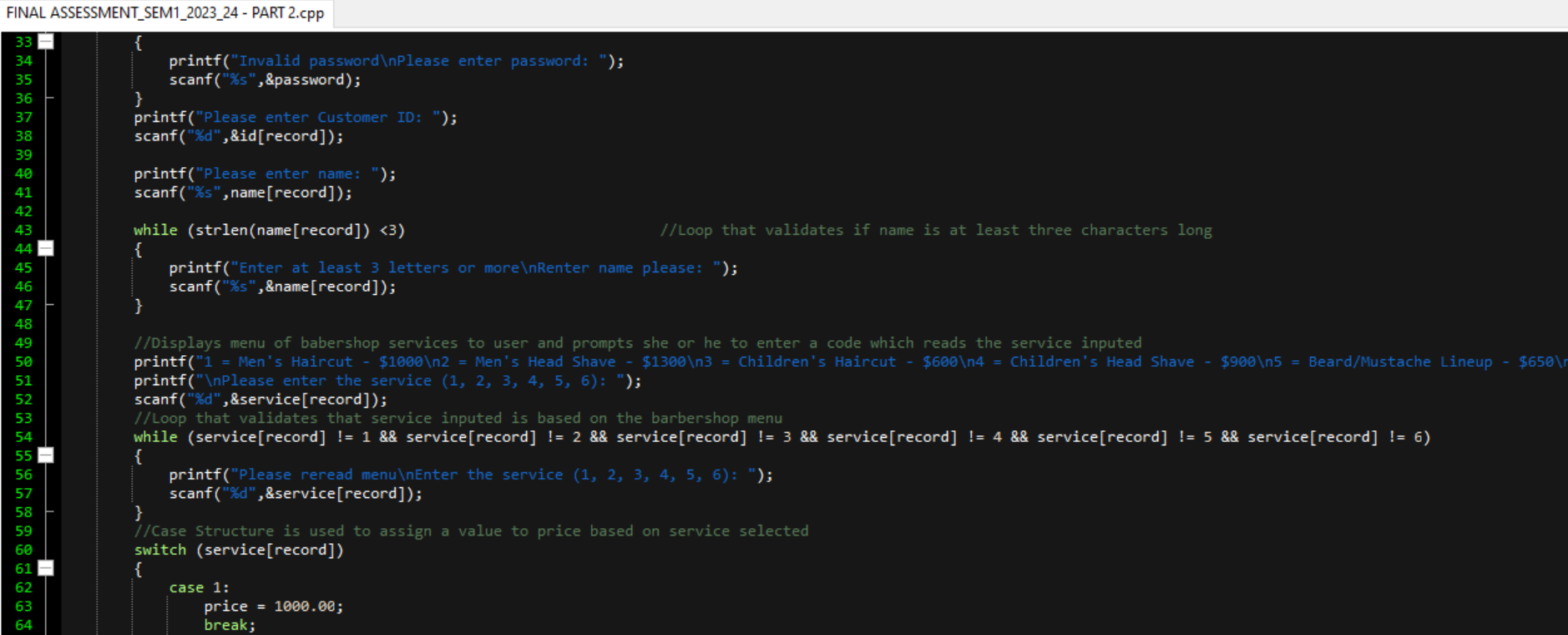
printf("\nRevenue for the day: $%.1f",rev);

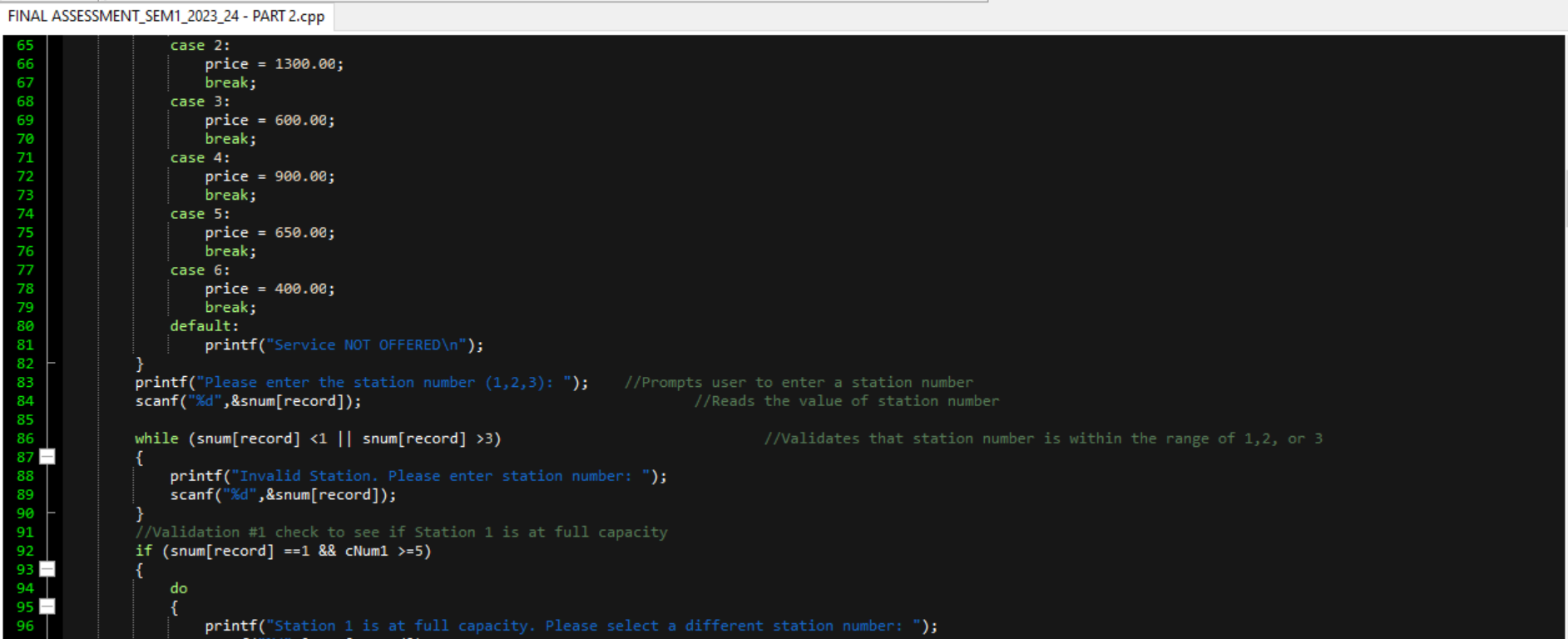
return 0;

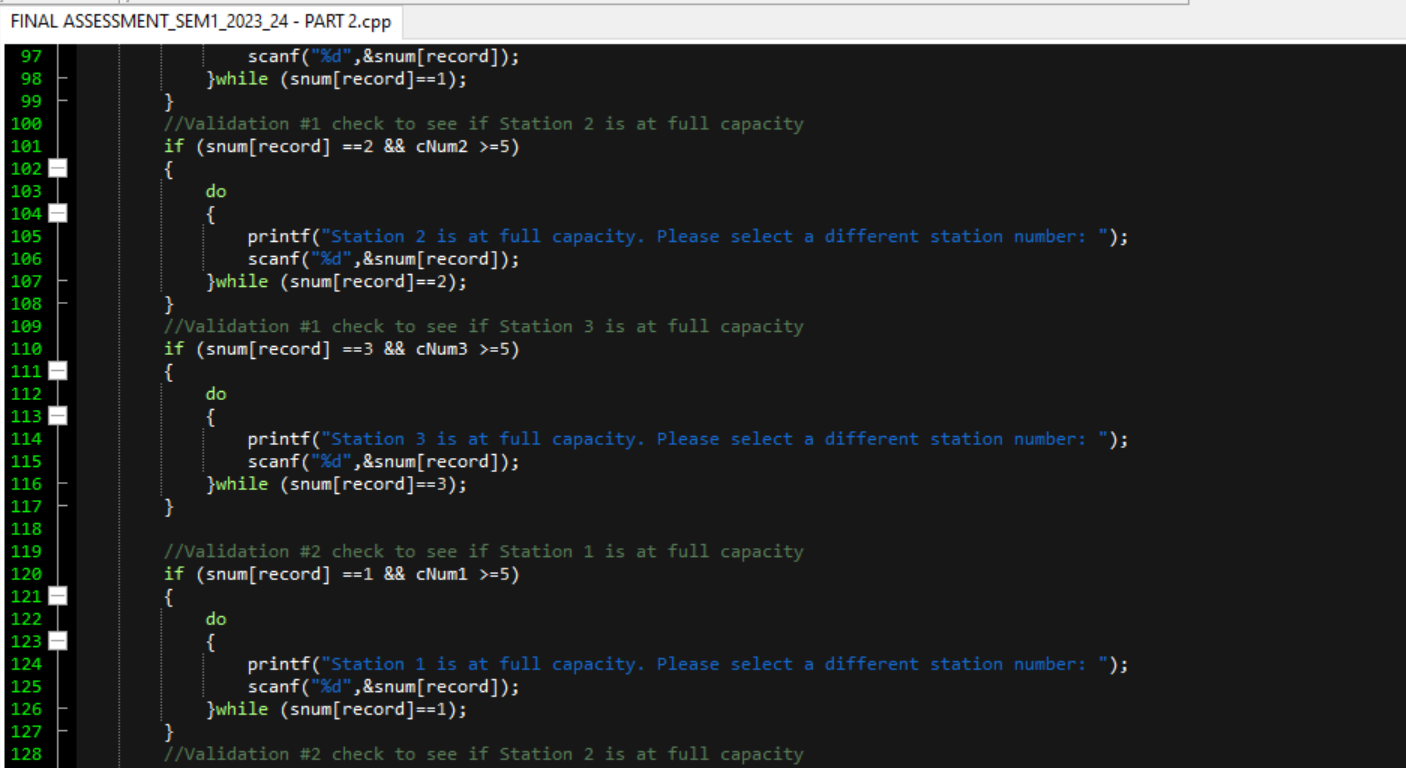
}

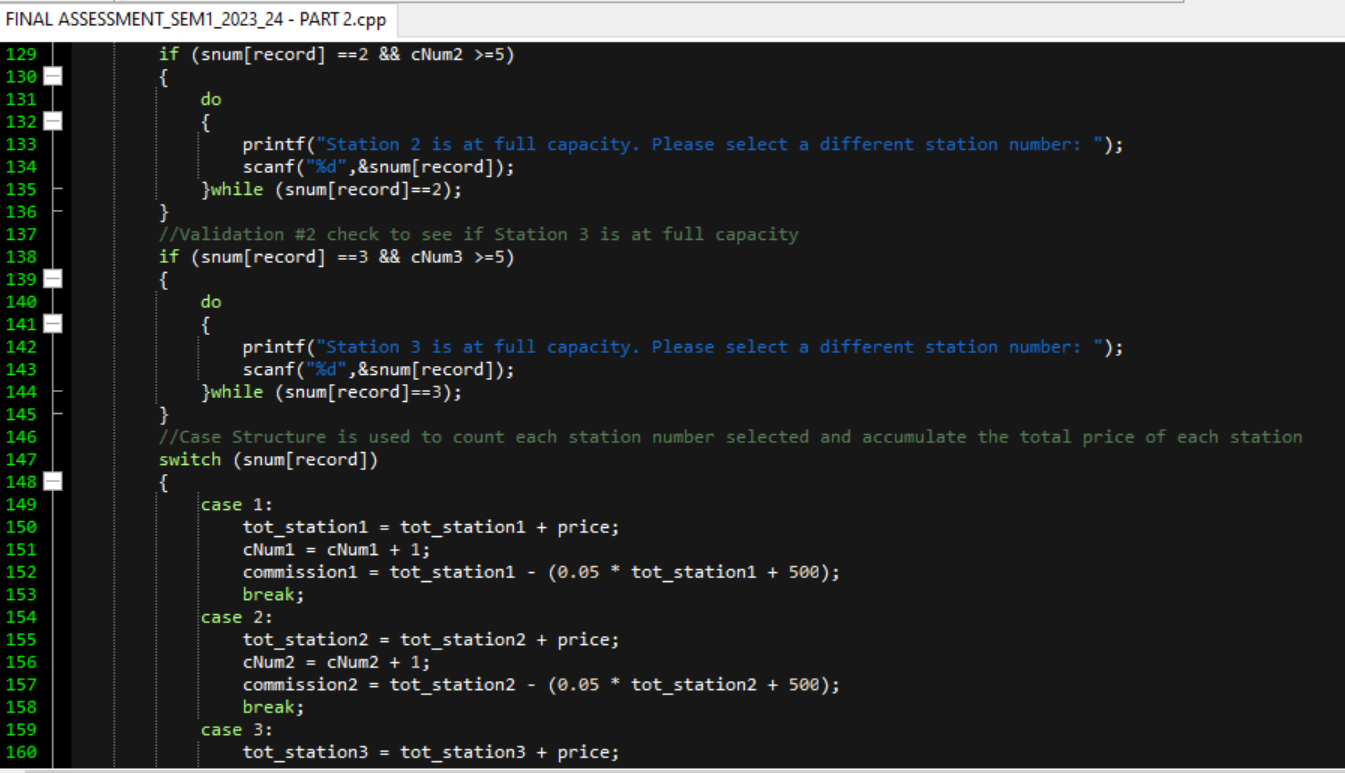
C Source Code Screenshot

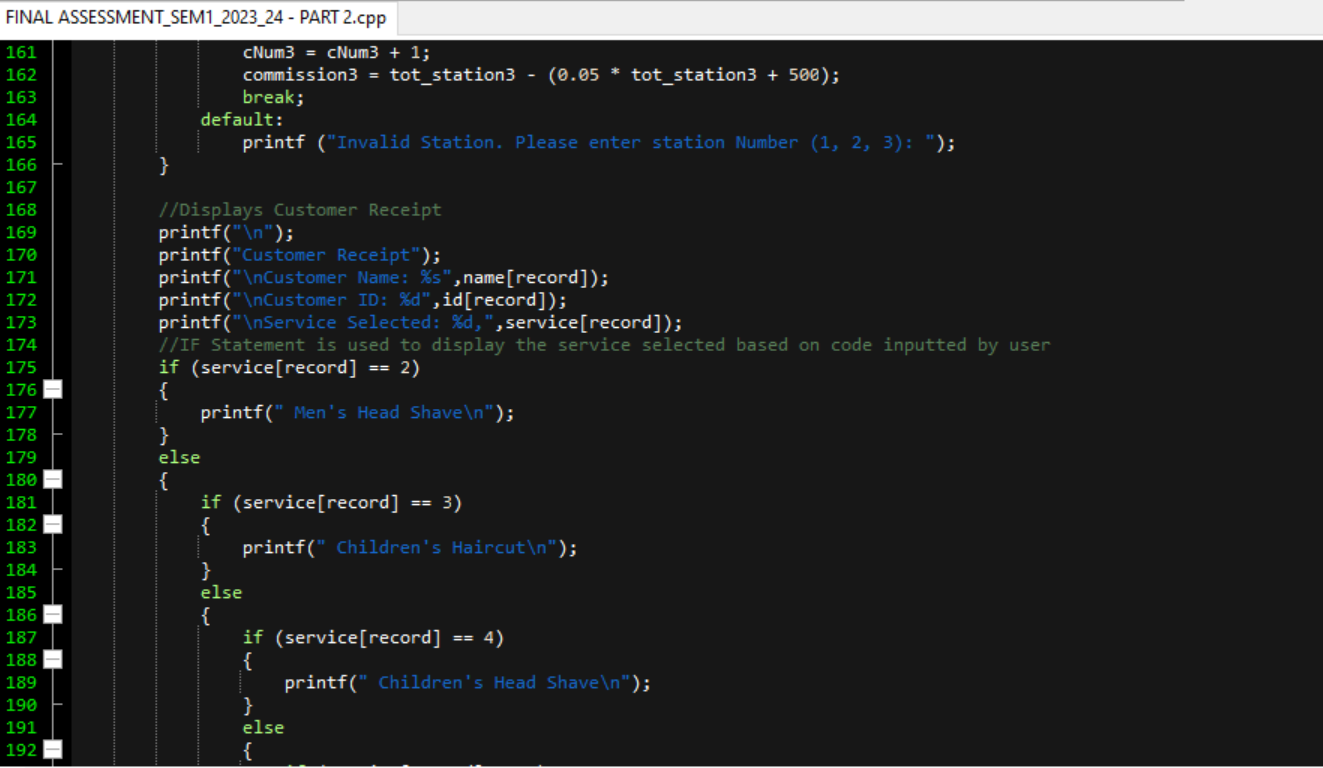




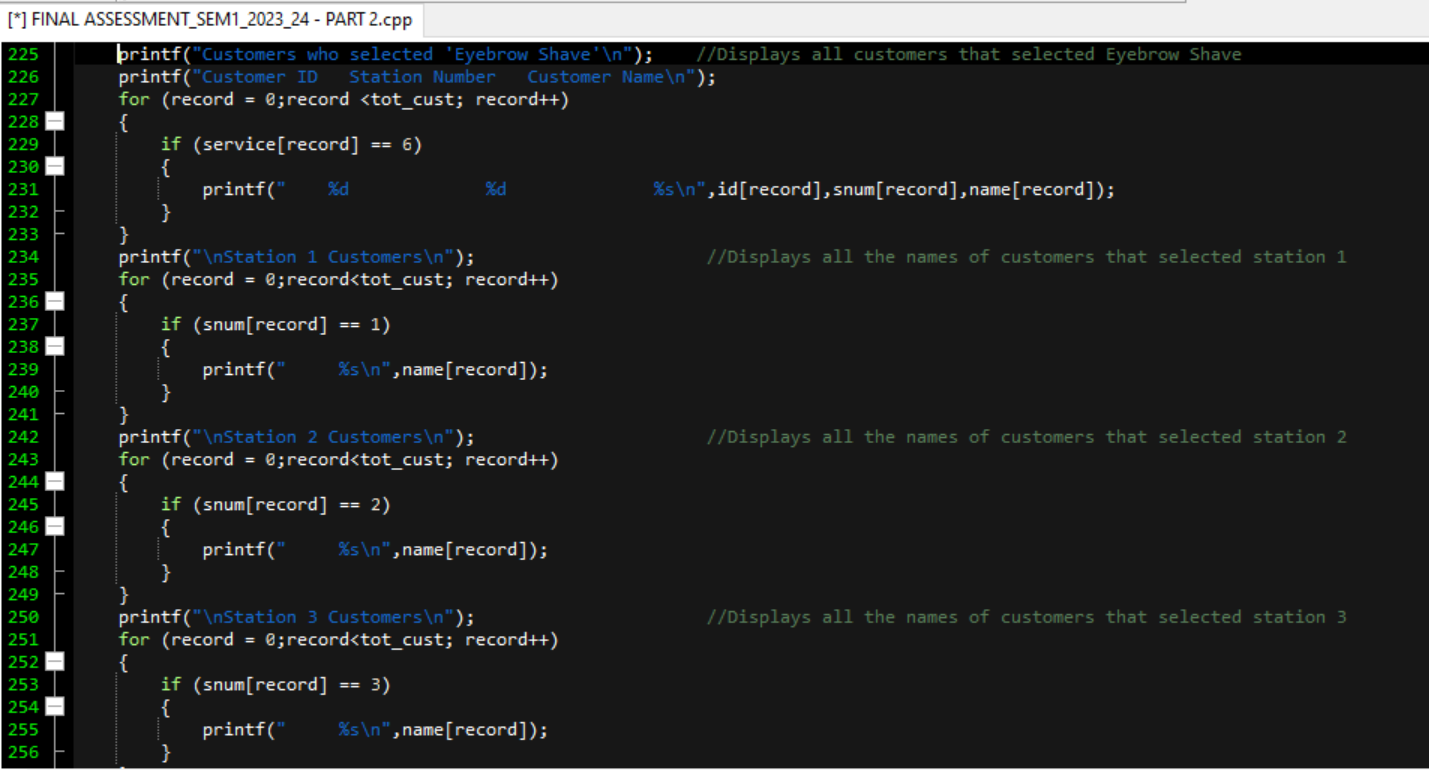


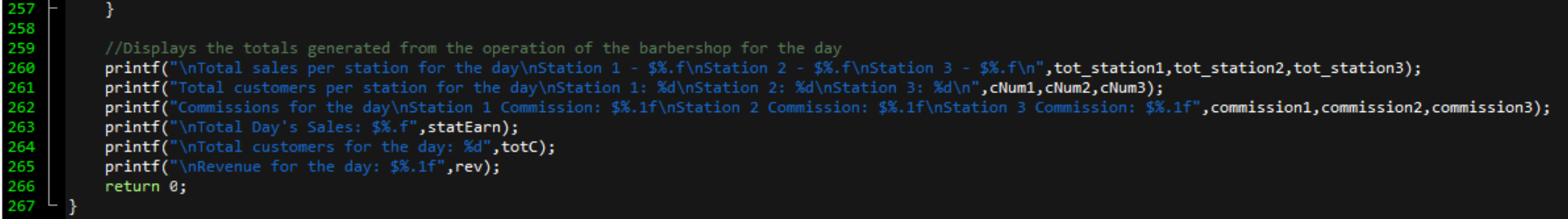




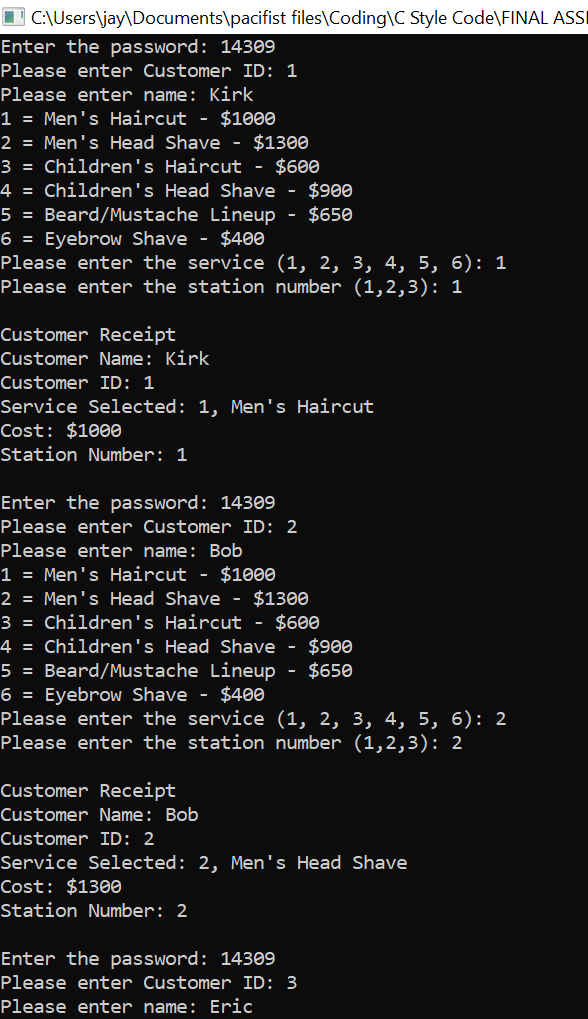


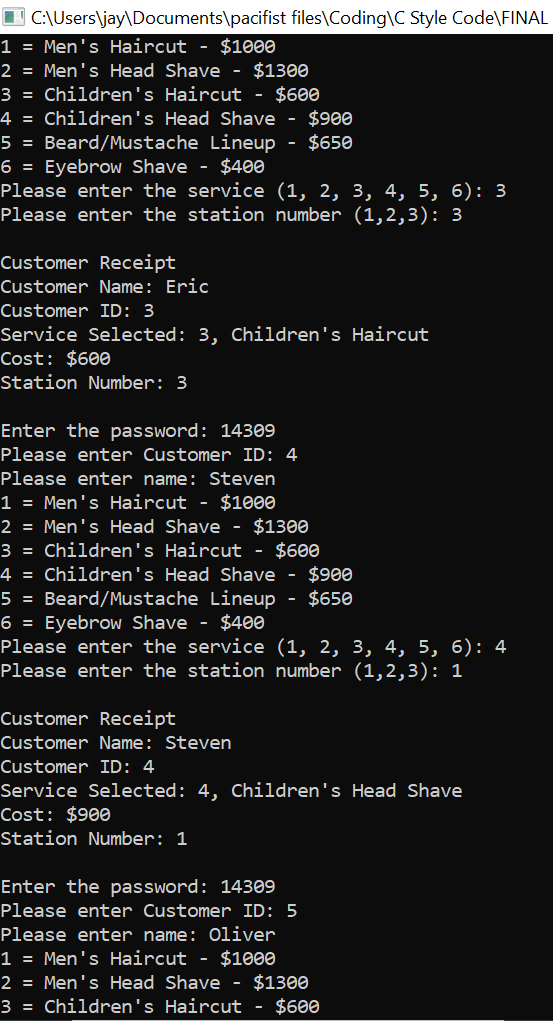


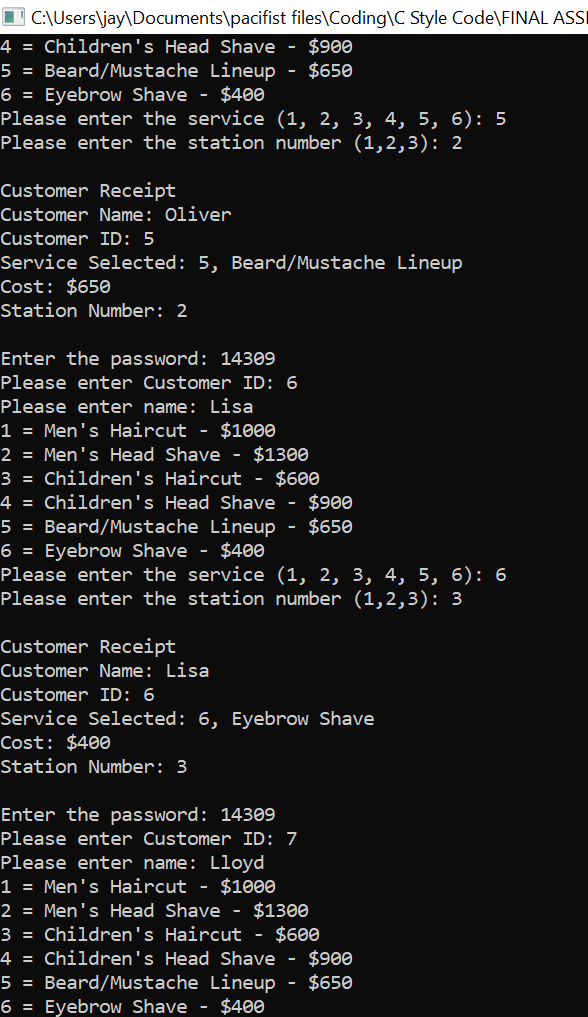


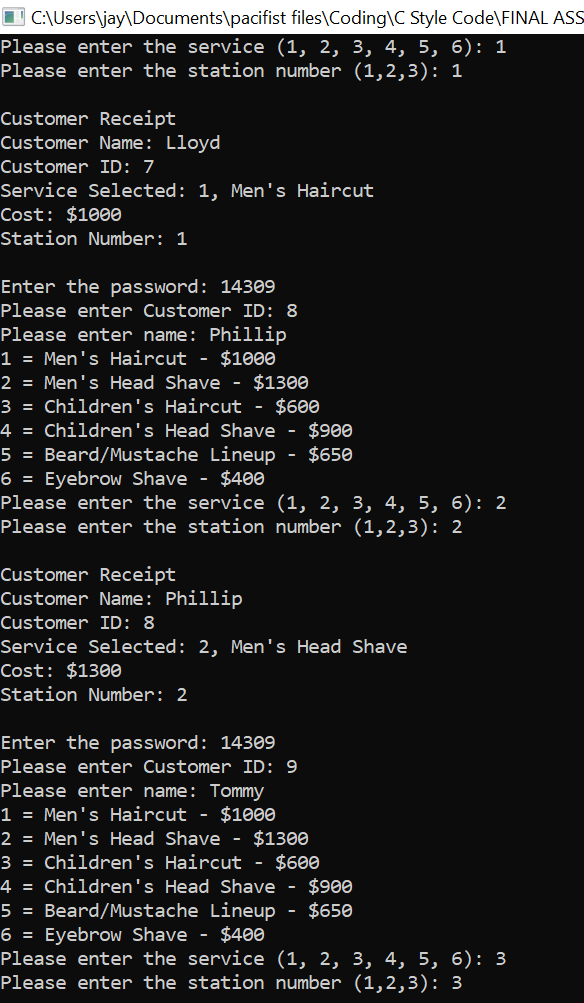


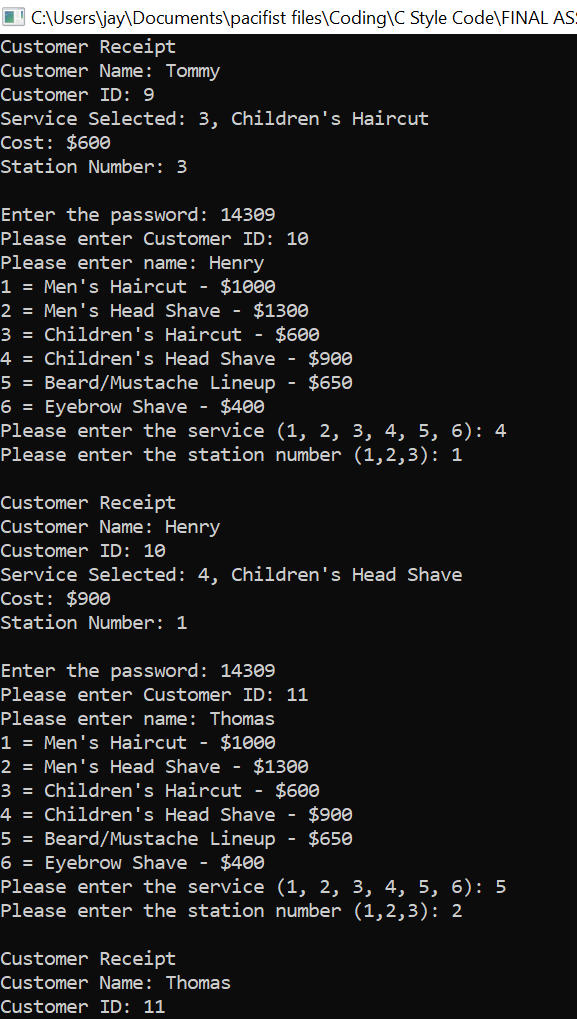
C Source Code Test Data Output

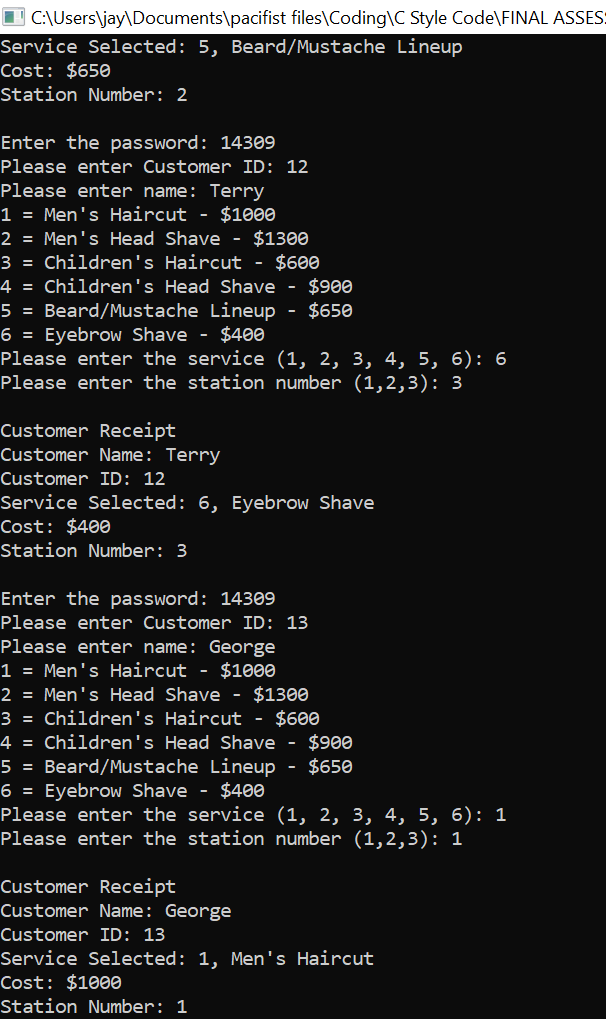


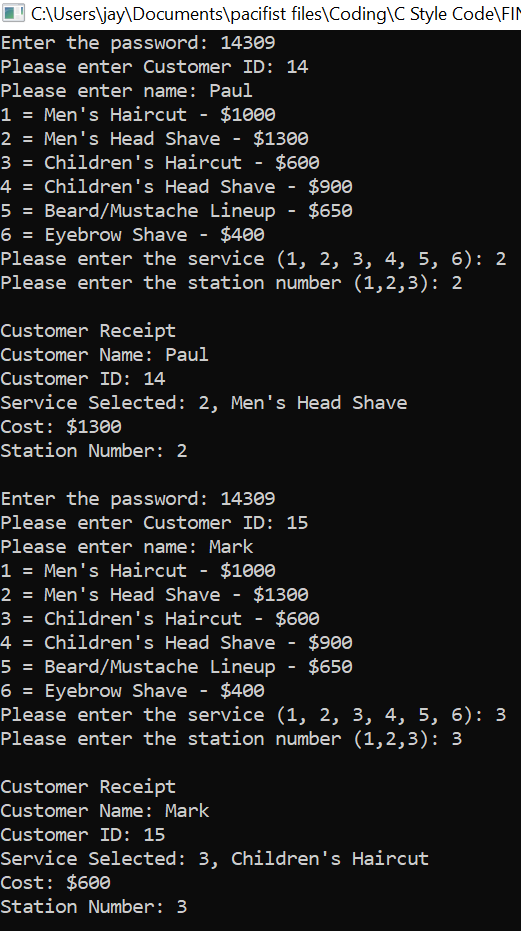


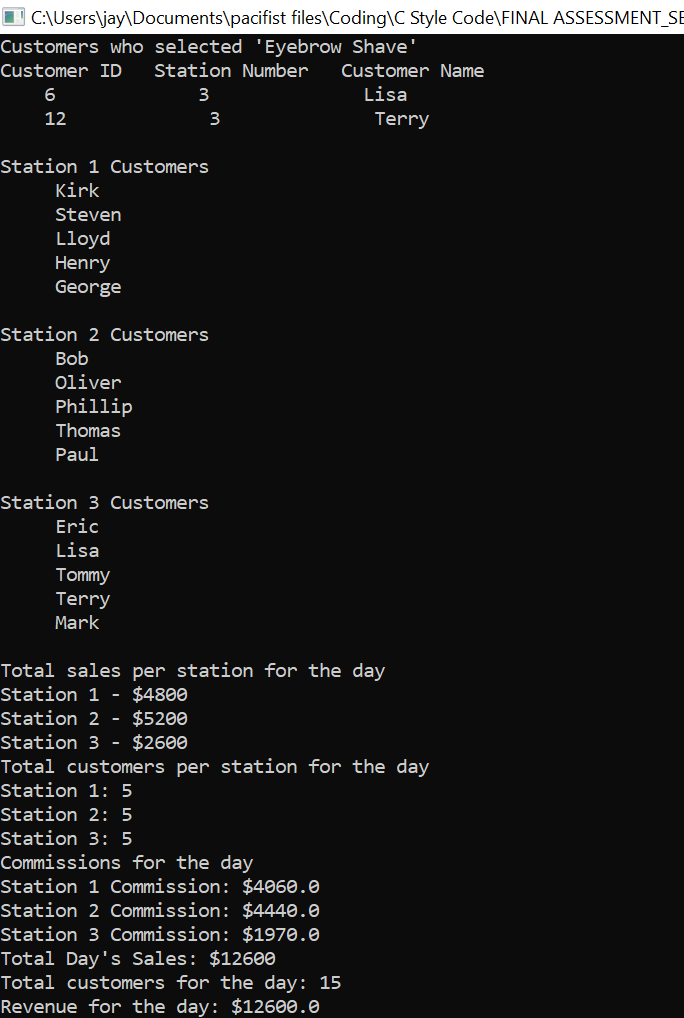


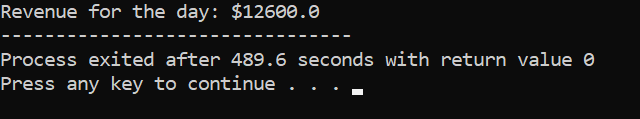












C Source Code Contribution:

Group members conducted a meeting during which they:

| Cameron, James | * Identified control structure, calculations and output for C source code |
| --- | --- |
| Ellis, Chevannese | * Facilitated meeting * Identified additional control structure, calculations and output for C Source Code |
| Myrie, Zachery | * Identified control structure, calculations and output for Pseudocode |
| Tennant, Aaliyah | * Identified control structure, calculations and output for Pseudocode |