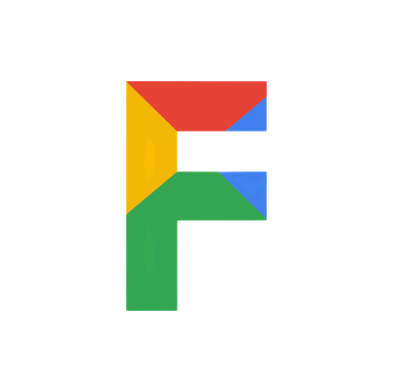
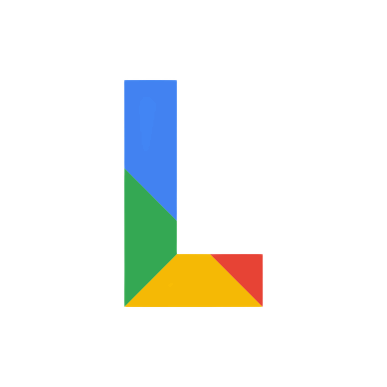
{   };



Course: CL1002 – Programming Fundamentals.

Instructor: Sir Muhammad Usman.

Submitted by: Muhammad Rehan

Roll no. 22P-9106

Class: BSE-1A (Fall 2022)

Assignment No 3

Date: December 11th, 2022.

Department of Computer Science

Problem 01:

Code:

#include <stdio.h>

#include <stdlib.h>

int \*arraycreation();

void unionset(int \*Setone, int \*Settwo, int ArrayOnesize, int ArrayTwosize, int union\_size, int result[]);

int arraysize;

int main(void)

{

    printf("Enter the elements of Set One (-1 to stop):\n");

    int \*Setone = arraycreation();

    int ArrayOnesize = arraysize;

    printf("Enter the elements of Set Two (-1 to stop):\n");

    int \*Settwo = arraycreation();

    int ArrayTwosize = arraysize;

    int same = 0;

    for (int i = 0; i < ArrayOnesize; i++)

    {

        for (int j = 0; j < ArrayTwosize; j++)

            if (Setone[i] == Settwo[j])

                same++;

    }

    int union\_size = (ArrayOnesize + ArrayTwosize) - same;

    int result[union\_size];

    unionset(Setone, Settwo, ArrayOnesize, ArrayTwosize, union\_size, result);

    return 0;

}

int \*arraycreation()

{

    int \*Numbers = NULL;

    size\_t NumberOfAllocatedElements = 0;

    int TemporaryNumber;

    int j = 1;

    arraysize = 0;

    while (786)

    {

        printf("Enter the number no %d : ", j++);

        scanf("%d", &TemporaryNumber);

        if (TemporaryNumber == -1)

            break;

        int \*NewNumbers = realloc(Numbers, ++NumberOfAllocatedElements \* sizeof \*NewNumbers);

        if (!NewNumbers)

        {

            fprintf(stderr, "Error, unable to allocate memory.\n");

            exit(EXIT\_FAILURE);

        }

        Numbers = NewNumbers;

        Numbers[NumberOfAllocatedElements - 1] = TemporaryNumber;

        arraysize++;

    }

    j = 0;

    return Numbers;

}

void unionset(int \*Setone, int \*Settwo, int ArrayOnesize, int ArrayTwosize, int union\_size, int result[])

{

    int i = 0, j = 0, x = 0, count = ArrayOnesize, flag = 1;

    for (i = 0; i < ArrayOnesize; i++)

    {

        result[i] = Setone[i];

    }

    for (i = 0; i < ArrayTwosize; i++)

    {

        flag = 1;

        x = 0;

        for (j = 0; flag == 1 && j < ArrayOnesize; j++)

        {

            if (Setone[j] == Settwo[i])

            {

                flag = 0;

            }

        }

        if (flag == 1)

        {

            result[count] = Settwo[i];

            count++;

        }

    }

    printf("Vettore generato:\n");

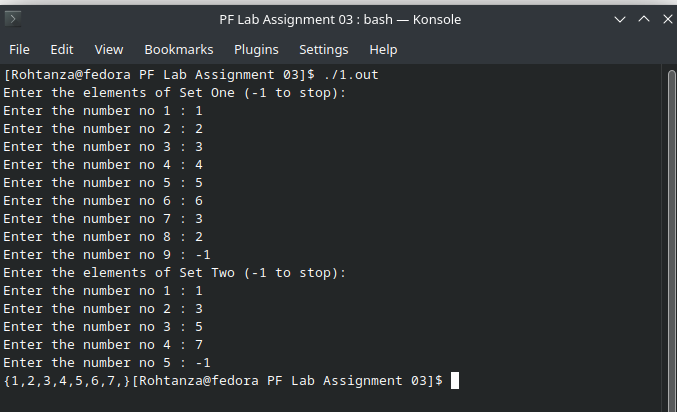
    for (i = 0; i < count; i++)

        printf(" %d ", result[i]);

    return ;

}

Screenshots:



Problem 02:

Code:

#include <stdio.h>

int main()

{

    int R\_one, R\_two, C\_one, C\_two;

    printf("Enter the Row and Coloum of Matrix One :");

    scanf("%d %d", &R\_one, &C\_one);

    printf("Enter the Row and Coloum of Matrix Two :");

    scanf("%d %d", &R\_two, &C\_two);

    if (C\_one != R\_two)

    {

        puts("\nCan't Multiply because Row of Matrix one isn't equal to Colume of Matrix two.\n");

        return 1;

    }

    int MatrixOne[R\_one][C\_one];

    int MatrixTwo[R\_two][C\_two];

    int Result[R\_one][C\_two];

    puts("Enter Matrix One:");

    for (int i = 0; i < R\_one; i++)

    {

        for (int j = 0; j < C\_one; j++)

        {

            printf("\tEnter the Element of Row:%d and Col:%d: ", (i + 1), (j + 1));

            scanf("%d", &MatrixOne[i][j]);

        }

    }

    puts("Enter Matrix Two:");

    for (int i = 0; i < R\_two; i++)

    {

        for (int j = 0; j < C\_two; j++)

        {

            printf("\tEnter the Element of Row:%d and Col:%d: ", (i + 1), (j + 1));

            scanf("%d", &MatrixTwo[i][j]);

        }

    }

    for (int i = 0; i < R\_one; i++)

    {

        for (int j = 0; j < C\_two; j++)

        {

            Result[i][j] = 0;

            for (int k = 0; k < C\_one; k++)

            {

                Result[i][j] += MatrixOne[i][k] \* MatrixTwo[k][j];

            }

        }

    }

    puts("Here's the Resultant Matric:\n");

    for (int i = 0; i < R\_one; i++)

    {

        for (int j = 0; j < C\_two; j++)

        {

            printf(" | %2d", Result[i][j]);

        }

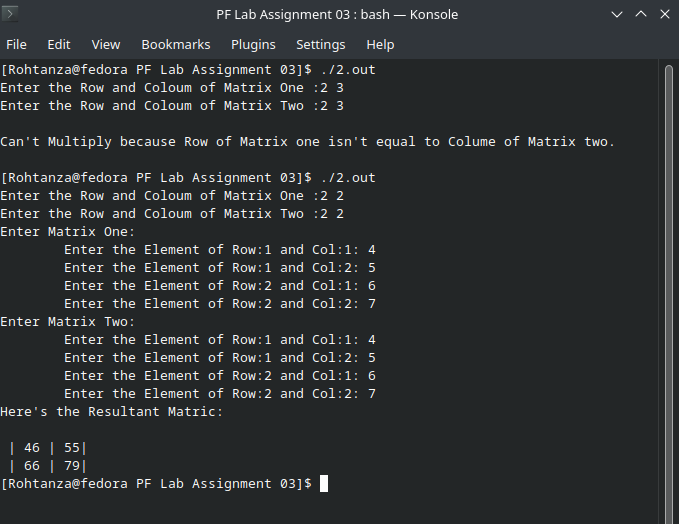
        printf("| \n");

    }

    return 0;

}

Screenshot:



Problem 03:

Code:

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

float CGPA(char \*\*grades, float points[]);

int main()

{

    char \*grades[] = {"A", "A-", "B+", "B", "B-", "C+", "C", "C-", "D+", "D", "F"};

    float points[] = {4.0, 3.67, 3.33, 3.0, 2.67, 2.33, 2.0, 1.67, 1.33, 1.0, 0};

    printf("The CPGA is %.2f : ", CGPA(grades, points));

    return 0;

}

float CGPA(char \*\*grades, float points[])

{

    int Subject;

    float gradepoints, point,credit\_hour\_sum;

    credit\_hour\_sum = gradepoints = 0;

    printf("Enter the count of the Subjects : ");

    scanf("%d", &Subject);

    for (int i = 0; i < Subject; i++)

    {

        char grade[3];

        int j = 0;

        float credit\_hour;

        printf("\nEnter Subject no \'%d\' :", (i + 1));

        scanf("%s", grade);

        printf("\nEnter the Credit hour of The Subject no \'%d\' :", (i + 1));

        scanf("%f", &credit\_hour);

        while (1)

        {

            if (strcmp(grades[j], grade) == 0)

            {

                point = points[j];

                break;

            }

            j++;

            if (j > 11)

            {

                puts("Enter a valid Grade\nTry Again");

                exit(1);

            }

        }

        gradepoints = gradepoints + (point \* credit\_hour);

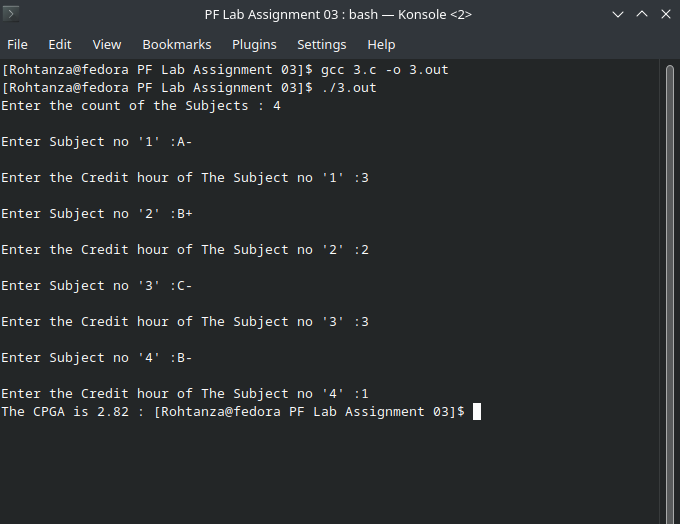
        credit\_hour\_sum += credit\_hour;

    }

    return (gradepoints / credit\_hour\_sum);

}

Screenshot:



Problem 04:

Code:

#include <stdio.h>

#include <stdlib.h>

#define Very\_Vanilla\_Chiller\_Small 361

#define Very\_Vanilla\_Chiller\_Regular 409

#define Cocoa\_Loco\_Small 361

#define Cocoa\_Loco\_Regular 409

#define CookiesCream\_Small 361

#define CookiesCream\_Regular 409

#define HazelNut\_Small 369

#define HazelNut\_Regular 461

#define Chocolate\_Macadania\_Small 369

#define Chocolate\_Macadania\_Regular 461

#define Italian\_Small 369

#define Italian\_Regular 461

#define Caramel\_Small 369

#define Caramel\_Regular 461

#define Tiramisu\_Small 399

#define Tiramisu\_Regular 509

#define Toffe\_Small 399

#define Toffe\_Regular 509

#define Siganture\_Small 300

#define Siganture\_Regular 374

#define Mocha\_Small 300

#define Mocha\_Regular 361

#define IcedCaramel\_Small 378

#define IcedCaramel\_Regular 430

#define Americano\_Small 252

#define Americano\_Regular 274

#define Blueberry\_Small 250

#define Blueberry\_Regular 291

#define Lychee\_Small 250

#define Lychee\_Regular 291

#define GreenApple\_Small 250

#define GreenApple\_Regular 291

#define Peach\_Small 250

#define Peach\_Regular 291

#define AppleSoda\_Small 335

#define AppleSoda\_Regular 348

#define Lime\_Small 335

#define Lime\_Regular 361

#define Peachtea\_Small 239

#define Peachtea\_Regular 291

#define Lemontea\_Small 239

#define Lemontea\_Regular 291

#define Lycheetea\_Small 239

#define Lycheetea\_Regular 291

#define IcedChoco\_Small 348

#define IcedChoco\_Regular 365

#define WhiteChoco\_Small 348

#define WhiteChoco\_Regular 365

#define Delight\_Small 348

#define Delight\_Regular 400

#define IcedLime\_Small 335

#define IcedLime\_Regular 365

#define AppleChiller\_Small 335

#define AppleChiller\_Regular 365

#define ChaiChiller\_Small 348

#define ChaiChiller\_Regular 400

#define GreenTeaChiiler\_Small 348

#define GreenTeaChiiler\_Regular 400

void Lines(void);

int Type(int Budget);

int flavour4Expresso(int Budget);

int flavour4OverIce(int Budget);

int flavour4Cholocate(int Budget);

int flavour4Fusion(int Budget);

int Bill(int Order);

int Budget;

int main()

{

    int Choice;

    Lines();

    // Showing a Greeting Message.

    printf("\nHi There, Welcome to the Glouria Jeans.\n\nKindly Enter your Budget:");

    scanf("%d", &Budget);

    if (Budget < 239)

    {

        // Checking if budget is less than Budget.

        printf("\nSorry our least expensive Product costs \"239\"\nYou'r just \"%d\" buck short.\n", 200 - Budget);

        Lines();

        printf("\n");

        return 0;

    }

    // Menu();

    Type(Budget);

    return 0;

}

int Type(int Budget)

{

    Lines();

    int Type;

    printf("\n\nEnter:\n\"1\" for Expresso & Mocha Chillers\n\"2\" for Over Ice\n\"3\" for Cholocalate Chillers\n\"4\" for Fusion\n=>");

    scanf("%d", &Type);

    if (Type == -1)

    {

        // invalid Entry checker

        printf("-1");

        exit(0);

    }

    switch (Type)

    {

        // Giving the user an option to select either cup or cone to order

    case 1:

        flavour4Expresso(Budget);

        break;

    case 2:

        flavour4OverIce(Budget);

        break;

    case 3:

        flavour4Cholocate(Budget);

        break;

    case 4:

        flavour4Fusion(Budget);

        break;

    case -1:

        // invalid Entry checker

        printf("-1");

        break;

    default:

        break;

    }

}

int flavour4Expresso(int Budget)

{

    Lines();

    int flavour;

    int Size;

    printf("\nEnter the Flavour you like :):\n\*1 for Very Vanilla Chiller\n\*2 for Cocoa Loco\n\*3 for Cookies N' Cream\n\*4 for Hazaelnut Mocha Chiller\n\*5 for Chocolate Macadamia Chillar\n\*6 for Italian Cholio Chiller\n\*7 for Caramel Nut Chiller\n\*8 for Tiramisu Chiller\n\*9 for Toffe Nut Chiiler\n=>");

    scanf("%d", &flavour);

    if (flavour == -1)

    {

        // invalid Entry checker

        printf("-1");

        exit(0);

    }

    switch (flavour)

    {

        // Giving users an option to select flavor to order.

    case 1:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Very\_Vanilla\_Chiller\_Small, Very\_Vanilla\_Chiller\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(Very\_Vanilla\_Chiller\_Small);

            break;

        case 2:

            Bill(Very\_Vanilla\_Chiller\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 2:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Cocoa\_Loco\_Small, Cocoa\_Loco\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(Cocoa\_Loco\_Small);

            break;

        case 2:

            Bill(Cocoa\_Loco\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 3:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", CookiesCream\_Small, CookiesCream\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(CookiesCream\_Small);

            break;

        case 2:

            Bill(CookiesCream\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 4:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", HazelNut\_Small, HazelNut\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(HazelNut\_Small);

            break;

        case 2:

            Bill(HazelNut\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 5:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Chocolate\_Macadania\_Small, Chocolate\_Macadania\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(Chocolate\_Macadania\_Small);

            break;

        case 2:

            Bill(Chocolate\_Macadania\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 6:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Italian\_Small, Italian\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(Italian\_Small);

            break;

        case 2:

            Bill(Italian\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 7:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Caramel\_Small, Caramel\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(Caramel\_Small);

            break;

        case 2:

            Bill(Caramel\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 8:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Tiramisu\_Small, Tiramisu\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(Tiramisu\_Small);

            break;

        case 2:

            Bill(Tiramisu\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 9:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Toffe\_Small, Toffe\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(Toffe\_Small);

            break;

        case 2:

            Bill(Toffe\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    default:

    Lines();

        printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

        // Remaining User to read the program carefully,

        // Calling the function again.

        Type(Budget);

        break;

    }

}

int flavour4OverIce(int Budget)

{

    Lines();

    int flavour;

    int Size;

    printf("\nEnter the Flavour you like :):\n\*1 for Signature Iced Coffe\n\*2 for Iced Mocha\n\*3 for Iced Caramel Latte\n\*4 for Iced Americano\n\*5 for BlueBerry Lemonade\n\*6 for Lychee Lemonade\n\*7 for Green Apple Lemonade\n\*8 for Peach Lemonade\n\*9 for Apple Soda\n\*10 for Lime Soda\n\*11 for Ice Teas\n=>");

    scanf("%d", &flavour);

    if (flavour == -1)

    {

        // invalid Entry checker

        printf("-1");

        exit(0);

    }

    switch (flavour)

    {

        // Giving users an option to select flavor to order.

    case 1:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Siganture\_Small, Siganture\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(Siganture\_Small);

            break;

        case 2:

            Bill(Siganture\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 2:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Mocha\_Small, Mocha\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(Mocha\_Small);

            break;

        case 2:

            Bill(Mocha\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 3:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", IcedCaramel\_Small, IcedCaramel\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(IcedCaramel\_Small);

            break;

        case 2:

            Bill(IcedCaramel\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 4:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Americano\_Small, Americano\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(Americano\_Small);

            break;

        case 2:

            Bill(Americano\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 5:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Blueberry\_Small, Blueberry\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(Blueberry\_Small);

            break;

        case 2:

            Bill(Blueberry\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 6:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Lychee\_Small, Lychee\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(Lychee\_Small);

            break;

        case 2:

            Bill(Lychee\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 7:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", GreenApple\_Small, GreenApple\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(GreenApple\_Small);

            break;

        case 2:

            Bill(GreenApple\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 8:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Peach\_Small, Peach\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(Peach\_Small);

            break;

        case 2:

            Bill(Peach\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 9:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", AppleSoda\_Small, AppleSoda\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(AppleSoda\_Small);

            break;

        case 2:

            Bill(AppleSoda\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 10:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Lime\_Small, Lime\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(Lime\_Small);

            break;

        case 2:

            Bill(Lime\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 11:

        int TeaType;

        printf("\nEnter the Tea you would like:\n\*1 for Peach Tea for %d \n\*2 for Lemon Tea for %d\n\*3 for Lychee Tea for %d\n=>");

        switch (TeaType)

        {

        case 1:

            Lines();

            printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Peachtea\_Small, Peachtea\_Regular);

            scanf("%d", &Size);

            switch (Size)

            {

            case 1:

                Bill(Peachtea\_Small);

                break;

            case 2:

                Bill(Peachtea\_Regular);

                break;

            default:

            Lines();

                printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

                // Remaining User to read the program carefully,

                // Calling the function again.

                flavour4Expresso(Budget);

                break;

            }

            break;

        case 2:

            Lines();

            printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Lemontea\_Small, Lemontea\_Regular);

            scanf("%d", &Size);

            switch (Size)

            {

            case 1:

                Bill(Lemontea\_Small);

                break;

            case 2:

                Bill(Lemontea\_Regular);

                break;

            default:

            Lines();

                printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

                // Remaining User to read the program carefully,

                // Calling the function again.

                flavour4Expresso(Budget);

                break;

            }

            break;

        case 3:

            Lines();

            printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Lycheetea\_Small, Lycheetea\_Regular);

            scanf("%d", &Size);

            switch (Size)

            {

            case 1:

                Bill(Lycheetea\_Small);

                break;

            case 2:

                Bill(Lycheetea\_Regular);

                break;

            default:

            Lines();

                printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

                // Remaining User to read the program carefully,

                // Calling the function again.

                flavour4Expresso(Budget);

                break;

            }

            break;

        }

    default:

    Lines();

        printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

        // Remaining User to read the program carefully,

        // Calling the function again.

        Type(Budget);

        break;

    }

}

int flavour4Cholocate(int Budget)

{

    Lines();

    int flavour;

    int Size;

    printf("\nEnter the Flavour you like :):\n\*1 for Orginal Iced Chiller\n\*2 for White Iced Chocolate\n\*3 for Chocolate Delight\n=>");

    scanf("%d", &flavour);

    if (flavour == -1)

    {

        // invalid Entry checker

        printf("-1");

        exit(0);

    }

    switch (flavour)

    {

        // Giving users an option to select flavor to order.

    case 1:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", IcedChoco\_Small, IcedChoco\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(IcedChoco\_Small);

            break;

        case 2:

            Bill(IcedChoco\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 2:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", WhiteChoco\_Small, WhiteChoco\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(WhiteChoco\_Small);

            break;

        case 2:

            Bill(WhiteChoco\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 3:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", Delight\_Small, Delight\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(Delight\_Small);

            break;

        case 2:

            Bill(Delight\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    default:

    Lines();

        printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

        // Remaining User to read the program carefully,

        // Calling the function again.

        Type(Budget);

        break;

    }

}

int flavour4Fusion(int Budget)

{

    Lines();

    int flavour;

    int Size;

    printf("\nEnter the Flavour you like :):\n\*1 Iced Lime\n\*2 for Apple Chiller\n\*3 for Chai Chiller\n\*4 for Green Tea Chiller\n=>");

    scanf("%d", &flavour);

    if (flavour == -1)

    {

        // invalid Entry checker

        printf("-1");

        exit(0);

    }

    switch (flavour)

    {

        // Giving users an option to select flavor to order.

    case 1:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", IcedLime\_Small, IcedLime\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(IcedLime\_Small);

            break;

        case 2:

            Bill(IcedLime\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 2:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", AppleChiller\_Small, AppleChiller\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(AppleChiller\_Small);

            break;

        case 2:

            Bill(AppleChiller\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 3:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", ChaiChiller\_Small, ChaiChiller\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(ChaiChiller\_Small);

            break;

        case 2:

            Bill(ChaiChiller\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    case 4:

        Lines();

        printf("\nEnter the Size you would like:\n\*1 for Small for %d \n\*2 for Regular for %d\n=>", GreenTeaChiiler\_Small, GreenTeaChiiler\_Regular);

        scanf("%d", &Size);

        switch (Size)

        {

        case 1:

            Bill(GreenTeaChiiler\_Small);

            break;

        case 2:

            Bill(GreenTeaChiiler\_Regular);

            break;

        default:

        Lines();

            printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

            // Remaining User to read the program carefully,

            // Calling the function again.

            flavour4Expresso(Budget);

            break;

        }

        break;

    default:

    Lines();

        printf("\nAt least read it carefully don't enter useless entries try again.\nthis time read everything.\n");

        // Remaining User to read the program carefully,

        // Calling the function again.

        Type(Budget);

        break;

    }

}

void Lines(void)

{

    // Just a loop to create lines.

    for (int i = 1; i < 80; i++)

    {

        printf("-");

    }

}

int Bill(int Order)

{

    // This Function proceeds the bill with the selected item.

    int Countiue;

    Lines();

    printf("\nYour Bill is \"%d\" and your Current Budget is %d", Order, Budget - Order);

    // Updating the current budget.

    Budget = Budget - Order;

    // Asking if the user wants to Continue.

    printf("\n\nWould you like to continue? if yes kindly Press \"1\" or Press any key to exit.\n=>");

    scanf("%d", &Countiue);

    if (Countiue == -1)

    {

        // invalid Entry checker

        printf("-1");

        exit(0);

    }

    if (Countiue == 1)

    {

        // if the player wants to continue then the player just has a budget of more than 239.

        if (Budget >= 239)

        {

            // If yes then proceed to order again.

            Type(Budget);

        }

        else

        {

            // If the Budget is less than 239 then excites the program.

            printf("\nSorry you are out of Budget :(\n\nHope to See you Soon.\n");

            exit(0);

        }

    }

    else

    {

        // If a player wants to leave the program then say the player the greetings.

        printf("\nThank you for Shopping here.\n\n See you soon :)\n");

        exit(0);

    }

}

Screenshot: 