```c
#ifndef PRINT
#define PRINT

#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

#include "../datatypes/enum.h"
#include "../datatypes/struct.h"

#define MAX_LEN 128

void DisplayImage(FILE *fptr)
{
    char readString[MAX_LEN];

    while (fgets(readString, sizeof(readString), fptr) != NULL)
        printf("%s", readString);
}

void DisplayTitle(char *filename)
{
    FILE *fptr = NULL;

    if ((fptr = fopen(filename, "r")) == NULL)
    {
        fprintf(stderr, "Error opening %s!\n", filename);
        exit(1);
    }
    DisplayImage(fptr);
}

void NoMemoryAlert()
{
    printf("\n================================================================================\n");
    printf("============================ Not enough memory! ============================\n");
    printf("================================================================================\n\n");
}

void DimensionSelectionInstructions()
{
    printf("\n================================================================================\n");
    printf("Calculate 2D or 3D object's properties? Type \"Exit\" if you want to leave the program:\n");
    printf("1. 2D\n2. 3D\n");
    printf("Type in your choice here: ");
}

void WrongDimensionInput()
{
    printf("\n================================================================================\n");
    printf("============== Invalid input! Please follow the instructions! ==============\n");
    printf("============ Key in \"2D\", \"3D\" or the choice index (\"1\" or \"2\"). ===========\n");
    printf("====================== Type \"Exit\" to leave the program. ======================\n");
    printf("================================================================================\n\n");
    printf("Type in your choice again here: ");
}

void ShapeSelectionInstructions()
{
    printf("\nSelect the shape to calculate its properties. Type \"Exit\" if you want to leave the program or \"Back\" if you want to reselect the
dimension:\n");
    printf("1. Rectangle\n2. Square\n3. Circle\n");
    printf("Type in your choice here: ");
}

void WrongShapeInput()
{
    printf("\n================================================================================\n");
    printf("============== Invalid input! Please follow the instructions! ==============\n");
    printf("================ Key in \"Rectangle\", \"Square\" or \"Circle\". ================\n");
    printf("====================== Type \"Exit\" to leave the program. ======================\n");
    printf("================== Type \"Back\" to reselect the dimension. ==================\n");
    printf("================================================================================\n\n");
    printf("Type in your choice again here: ");
}

void ObjectSelectionInstructions()
{
    printf("\nSelect the object to calculate its properties. Type \"Exit\" if you want to leave the program or \"Back\" if you want to reselect the
dimension:\n");
    printf("1. Cuboid\n2. Cube\n3. Sphere\n4. Cone\n");
    printf("Type in your choice here: ");
}

void WrongObjectInput()
{
    printf("\n================================================================================\n");
    printf("============== Invalid input! Please follow the instructions! ==============\n");
    printf("============== Key in \"Cuboid\", \"Cube\", \"Sphere\" or \"Cone\". ==============\n");
    printf("====================== Type \"Exit\" to leave the program. ======================\n");
    printf("================== Type \"Back\" to reselect the dimension. ==================\n");
    printf("================================================================================\n\n");
    printf("Type in your choice again here: ");
}

void UnitSelectionInstructions()
{
    printf("\nSelect the input unit:\n");
    printf("1. m\n2. dm\n3. cm\n4. mm\n");
    printf("Select unit: ");
}

void WrongUnitInput()
{
```

```
    printf("\n================================================================================\n");
    printf("============== Invalid input! Please follow the instructions! ==============\n");
    printf("====================== Enter \"m\", \"dm\", \"cm\" or \"mm\". ======================\n");
    printf("============= Or enter the choice index: \"1\", \"2\", \"3\" or \"4\". =============\n");
    printf("================================================================================\n\n");
    printf("Enter again here: ");
}

void DisplayResults(enum shape shape, double result_1, double result_2)
{
    bool is2D = false;

    printf("\nCalculation results:\n");
    printf("    _____________________________________________________________________________________________");

    if (shape == Rectangle || shape == Square || shape == Circle)
    {
        is2D = true;
    }

    if (is2D)
    {
        if (shape != Circle)
        {
            printf("\n   |    Perimeter  | ");
        }
        else
        {
            printf("\n   | Circumference | ");
        }

        printf("%12.2g m   | %12.2g dm   | %12.2g cm   | %12.2g mm   |\n", result_1, result_1 * 10, result_1 * 1E2, result_1 * 1E3);
        printf("   |_______________|_____________|_____________|_____________|\n");
        printf("   |      Area     | %12.2g m^2 | %12.2g dm^2 | %12.2g cm^2 | %12.2g mm^2 |\n", result_2, result_2 * 1E2, result_2 * 1E4, result_2 * 1E6);
        printf("   |_______________|_____________|_____________|_____________|_____________|\n");
    }
    else
    {
        printf("___________\n");
        printf("   |    Surface area  | %12.2g m^2   | %12.2g dm^2   | %12.2g cm^2   | %12.2g mm^2   |\n", result_1, result_1 * 1E2, result_1 * 1E4, result_1 *
1E6);
        printf("   |__________________|_____________|_____________|_____________|_____________|\n");
        printf("   |      Volume      | %12.2g m^3   | %12.2g dm^3   | %12.2g cm^3   | %12.2g mm^3   |\n", result_2, result_2 * 1E3, result_2 * 1E6, result_2 *
1E9);
        printf("   |__________________|_____________|_____________|_____________|_____________|\n");
    }
}

void ParamaterSelectionInstructions(char *parameter)
{
    printf("\nEnter the %s parameter\n", parameter);
    printf("Enter the value here: ");
}

void NumericInputAlert(bool isNumeric)
{
    if (isNumeric)
    {
        printf("\n================================================================================\n");
        printf("======================== Enter a positive number! ========================\n");
        printf("================================================================================\n");
    }
    else
    {
        printf("\n================================================================================\n");
        printf("============================== Enter a number! ==============================\n");
        printf("================================================================================\n");
    }
    printf("Enter again here: ");
}

void ProcessSelectionInstructions()
{
    printf("\nSelect:\n");
    printf("1. History\t- To view the calculation history.\n");
    printf("2. Calculate\t- To calculate again.\n");
    printf("3. Exit\t\t- To leave the program.\n");
    printf("Enter your choice here: ");
}

void WrongProcessInput()
{
    printf("\n================================================================================\n");
    printf("============== Invalid input! Please follow the instructions! ==============\n");
    printf("============ Key in \"History\", \"Calculate\" or \"Exit\" ===========\n");
    printf("================================================================================\n\n");
    printf("Type in your choice again here: ");
}

void ShapeAndObjectSelectionInstructions()
{
    printf("\nSelect any of the option\n");
    printf("1. Rectangle\n2. Square\n3. Circle\n");
    printf("4. Cuboid\n5. Cube\n6. Sphere\n7. Cone\n");
    printf("Type in your choice here: ");
}

void WrongShapeAndObjectInput()
{
    printf("\n================================================================================\n");
    printf("============== Invalid input! Please follow the instructions! ==============\n");
    printf("============ Key in \"Rectangle\", \"Square\", \"Circle\", \"Cuboid\", \"Cube\", \"Sphere\", \"Cone\" ===========\n");
    printf("================================================================================\n\n");
    printf("Type in your choice again here: ");
```

```
}

void DisplayHistoryTable(enum shape shape, struct History *history, double *means, double *stds)
{
    int i;

    switch (shape)
    {
    case Rectangle:
        if (history->count[0] == 0)
        {
            printf("\n================================================================================\n");
            printf("=============== The rectangle calculation history is empty. ===============\n");
            printf("================================================================================\n");
        }
        else
        {
            printf("\nCalculation Histroy of Rectangle\n");
            printf("    _____________________________________________________________________________________\n");
            printf("   |         Index         |       Width       |       Lenth       |     Perimeter     |        Area        |\n");
            printf("   |_______________________|___________________|___________________|___________________|____________________|\n");

            for (i = 0; i < history->count[0]; i++)
            {
                printf("   | %11d           | %12.2g m    | %12.2g m    | %12.2g m    | %12.2g m^2    |\n", i + 1, history->rectangles[i].width, history-
>rectangles[i].length, history->rectangles[i].perimeter, history->rectangles[i].area);
                printf("   |_______________________|___________________|___________________|___________________|____________________|\n");
            }

            printf("   |         Mean          | %12.2g m    | %12.2g m    | %12.2g m    | %12.2g m^2    |\n", means[0], means[1], means[2], means[3]);
            printf("   |_______________________|___________________|___________________|___________________|____________________|\n");
            printf("   |  Standard Deviation   | %12.2g m    | %12.2g m    | %12.2g m    | %12.2g m^2    |\n", stds[0], stds[1], stds[2], stds[3]);
            printf("   |_______________________|___________________|___________________|___________________|____________________|\n");
        }

        break;

    case Square:
        if (history->count[1] == 0)
        {
            printf("\n================================================================================\n");
            printf("================= The square calculation history is empty. =================\n");
            printf("================================================================================\n");
        }
        else
        {
            printf("\nCalculation Histroy of Square\n");
            printf("    _____________________________________________________________________\n");
            printf("   |         Index         |    Side Lenth     |     Perimeter     |        Area        |\n");
            printf("   |_______________________|___________________|___________________|____________________|\n");

            for (i = 0; i < history->count[1]; i++)
            {
                printf("   | %11d           | %12.2g m    | %12.2g m    | %12.2g m^2    |\n", i + 1, history->squares[i].length, history->squares[i].perimeter,
history->squares[i].area);
                printf("   |_______________________|___________________|___________________|____________________|\n");
            }

            printf("   |         Mean          | %12.2g m    | %12.2g m    | %12.2g m^2    |\n", means[0], means[1], means[2]);
            printf("   |_______________________|___________________|___________________|____________________|\n");
            printf("   |  Standard Deviation   | %12.2g m    | %12.2g m    | %12.2g m^2    |\n", stds[0], stds[1], stds[2]);
            printf("   |_______________________|___________________|___________________|____________________|\n");
        }

        break;

    case Circle:
        if (history->count[2] == 0)
        {
            printf("\n================================================================================\n");
            printf("================= The circle calculation history is empty. =================\n");
            printf("================================================================================\n");
        }
        else
        {
            printf("\nCalculation Histroy of Circle\n");
            printf("    _____________________________________________________________________\n");
            printf("   |         Index         |      Radius       |   Circumference   |        Area        |\n");
            printf("   |_______________________|___________________|___________________|____________________|\n");

            for (i = 0; i < history->count[2]; i++)
            {
                printf("   | %11d           | %12.2g m    | %12.2g m    | %12.2g m^2    |\n", i + 1, history->circles[i].radius, history->circles[i].circumference,
history->squares[i].area);
                printf("   |_______________________|___________________|___________________|____________________|\n");
            }

            printf("   |         Mean          | %12.2g m    | %12.2g m    | %12.2g m^2    |\n", means[0], means[1], means[2]);
            printf("   |_______________________|___________________|___________________|____________________|\n");
            printf("   |  Standard Deviation   | %12.2g m    | %12.2g m    | %12.2g m^2    |\n", stds[0], stds[1], stds[2]);
            printf("   |_______________________|___________________|___________________|____________________|\n");
        }

        break;

    case Cuboid:
        if (history->count[3] == 0)
        {
            printf("\n================================================================================\n");
            printf("================= The cuboid calculation history is empty. =================\n");
            printf("================================================================================\n");
        }
        else
        {
            printf("\nCalculation Histroy of Cuboid\n");
```

```
            printf("    _____________________________________________________________________________________________\n");
            printf("   |         Index         |       Width       |       Lenth       |       Height      |    Surface Area    |       Volume       |\n");
            printf("   |_______________________|___________________|___________________|___________________|____________________|____________________|\n");

            for (i = 0; i < history->count[3]; i++)
            {
                printf("   | %11d           | %12.2g m    | %12.2g m    | %12.2g m    | %12.2g m^2    | %12.2g m^3    |\n", i + 1, history->cuboids[i].width, history->cuboids[i].length, history->cuboids[i].height, history->cuboids[i].area, history->cuboids[i].volume);
                printf("   |_______________________|___________________|___________________|___________________|____________________|____________________|\n");
            }

            printf("   |         Mean          | %12.2g m    | %12.2g m    | %12.2g m    | %12.2g m^2    | %12.2g m^3    |\n", means[0], means[1], means[2], means[3], means[4]);
            printf("   |_______________________|___________________|___________________|___________________|____________________|____________________|\n");
            printf("   |  Standard Deviation   | %12.2g m    | %12.2g m    | %12.2g m    | %12.2g m^2    | %12.2g m^3    |\n", stds[0], stds[1], stds[2], stds[3], stds[4]);
            printf("   |_______________________|___________________|___________________|___________________|____________________|____________________|\n");
        }

        break;

    case Cube:
        if (history->count[4] == 0)
        {
            printf("\n=============================================================================\n");
            printf("================= The cube calculation history is empty. ==================\n");
            printf("=============================================================================\n");
        }
        else
        {
            printf("\nCalculation Histroy of Cube\n");
            printf("    ____________________________________________________________________\n");
            printf("   |         Index         |    Side Length    |    Surface Area    |       Volume       |\n");
            printf("   |_______________________|___________________|____________________|____________________|\n");

            for (i = 0; i < history->count[4]; i++)
            {
                printf("   | %11d           | %12.2g m    | %12.2g m^2    | %12.2g m^3    |\n", i + 1, history->cubes[i].length, history->cubes[i].area, history->cubes[i].volume);
                printf("   |_______________________|___________________|____________________|____________________|\n");
            }

            printf("   |         Mean          | %12.2g m    | %12.2g m^2    | %12.2g m^3    |\n", means[0], means[1], means[2]);
            printf("   |_______________________|___________________|____________________|____________________|\n");
            printf("   |  Standard Deviation   | %12.2g m    | %12.2g m^2    | %12.2g m^3    |\n", stds[0], stds[1], stds[2]);
            printf("   |_______________________|___________________|____________________|____________________|\n");
        }

        break;

    case Sphere:
        if (history->count[5] == 0)
        {
            printf("\n=============================================================================\n");
            printf("================= The sphere calculation history is empty. =================\n");
            printf("=============================================================================\n");
        }
        else
        {
            printf("\nCalculation Histroy of Sphere\n");
            printf("    ____________________________________________________________________\n");
            printf("   |         Index         |      Radius       |    Surface Area    |       Volume       |\n");
            printf("   |_______________________|___________________|____________________|____________________|\n");

            for (i = 0; i < history->count[5]; i++)
            {
                printf("   | %11d           | %12.2g m    | %12.2g m^2    | %12.2g m^3    |\n", i + 1, history->spheres[i].radius, history->spheres[i].area, history->spheres[i].volume);
                printf("   |_______________________|___________________|____________________|____________________|\n");
            }

            printf("   |         Mean          | %12.2g m    | %12.2g m^2    | %12.2g m^3    |\n", means[0], means[1], means[2]);
            printf("   |_______________________|___________________|____________________|____________________|\n");
            printf("   |  Standard Deviation   | %12.2g m    | %12.2g m^2    | %12.2g m^3    |\n", stds[0], stds[1], stds[2]);
            printf("   |_______________________|___________________|____________________|____________________|\n");
        }

        break;

    case Cone:
        if (history->count[6] == 0)
        {
            printf("\n=============================================================================\n");
            printf("================= The cone calculation history is empty. ==================\n");
            printf("=============================================================================\n");
        }
        else
        {
            printf("\nCalculation Histroy of Cone\n");
            printf("    _____________________________________________________________________________________\n");
            printf("   |         Index         |       Radius      |       Height      |    Surface Area    |       Volume       |\n");
            printf("   |_______________________|___________________|___________________|____________________|____________________|\n");

            for (i = 0; i < history->count[6]; i++)
            {
                printf("   | %11d           | %12.2g m    | %12.2g m    | %12.2g m^2    | %12.2g m^3    |\n", i + 1, history->cones[i].radius, history->cones[i].height, history->cones[i].area, history->cones[i].volume);
                printf("   |_______________________|___________________|___________________|____________________|____________________|\n");
            }

            printf("   |         Mean          | %12.2g m    | %12.2g m    | %12.2g m^2    | %12.2g m^3    |\n", means[0], means[1], means[2], means[3]);
            printf("   |_______________________|___________________|___________________|____________________|____________________|\n");
            printf("   |  Standard Deviation   | %12.2g m    | %12.2g m    | %12.2g m^2    | %12.2g m^3    |\n", stds[0], stds[1], stds[2], stds[3]);
            printf("   |_______________________|___________________|___________________|____________________|____________________|\n");
        }
```

```
        break;
    }
}

#endif
```