UNIT CONVERTER

```c
#include <stdio.h>

// Length Conversions
double metersToFeet(double meters) {
    return meters * 3.28084;
}

double metersToInches(double meters) {
    return meters * 39.3701;
}

double feetToMeters(double feet) {
    return feet / 3.28084;
}

double inchesToMeters(double inches) {
    return inches / 39.3701;
}

// Temperature Conversions
double celsiusToFahrenheit(double celsius) {
    return (celsius * 9 / 5) + 32;
}

double fahrenheitToCelsius(double fahrenheit) {
    return (fahrenheit - 32) * 5 / 9;
}

// Weight Conversions
double kilogramsToPounds(double kilograms) {
    return kilograms * 2.20462;
}

double poundsToKilograms(double pounds) {
    return pounds / 2.20462;
}

// Volume Conversions
double litersToGallons(double liters) {
    return liters * 0.264172;
}
```

```c
double gallonsToLiters(double gallons) {
    return gallons / 0.264172;
}

// Time Conversions
double hoursToMinutes(double hours) {
    return hours * 60;
}

double minutesToHours(double minutes) {
    return minutes / 60;
}

// Currency Conversions
double usdToEuro(double usd) {
    return usd * 0.85;
}

double euroToUsd(double euro) {
    return euro / 0.85;
}

// Area Conversions
double squareMetersToSquareFeet(double sqmeters) {
    return sqmeters * 10.7639;
}

double squareFeetToSquareMeters(double sqfeet) {
    return sqfeet / 10.7639;
}

// Speed Conversions
double kphToMph(double kph) {
    return kph * 0.621371;
}

double mphToKph(double mph) {
    return mph / 0.621371;
}

int main() {
    int choice;
    double value;

    while (1) {
        printf("Unit Converter Menu:\n");
        printf("1. Length Conversions\n");
        printf("2. Temperature Conversions\n");
```

```
        printf("3. Weight Conversions\n");
        printf("4. Volume Conversions\n");
        printf("5. Time Conversions\n");
        printf("6. Currency Conversions\n");
        printf("7. Area Conversions\n");
        printf("8. Speed Conversions\n");
        printf("9. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        if (choice == 9) {
            printf("Exiting the program.\n");
            break;
        }

        if (choice < 1 || choice > 8) {
            printf("Invalid choice. Please enter a valid option.\n");
            continue;
        }

        printf("Enter the value to convert: ");
        scanf("%lf", &value);

        double result;
        switch (choice) {
            case 1:
                printf("Length Conversion Menu:\n");
                printf("1. Meters to Feet\n");
                printf("2. Meters to Inches\n");
                printf("3. Feet to Meters\n");
                printf("4. Inches to Meters\n");
                printf("Enter your choice: ");
                scanf("%d", &choice);

                switch (choice) {
                    case 1:
                        result = metersToFeet(value);
                        printf("%.2lf meters is equal to %.2lf feet.\n",
value, result);
                        break;
                    case 2:
                        result = metersToInches(value);
                        printf("%.2lf meters is equal to %.2lf inches.\n",
value, result);
                        break;
                    case 3:
                        result = feetToMeters(value);
```

```
                        printf("%.2lf feet is equal to %.2lf meters.\n",
value, result);
                        break;
                    case 4:
                        result = inchesToMeters(value);
                        printf("%.2lf inches is equal to %.2lf meters.\n",
value, result);
                        break;
                    default:
                        printf("Invalid choice. Please enter a valid
option.\n");
                }
                break;
            case 2:
                printf("Temperature Conversion Menu:\n");
                printf("1. Celsius to Fahrenheit\n");
                printf("2. Fahrenheit to Celsius\n");
                printf("Enter your choice: ");
                scanf("%d", &choice);

                switch (choice) {
                    case 1:
                        result = celsiusToFahrenheit(value);
                        printf("%.2lf degrees Celsius is equal to %.2lf
degrees Fahrenheit.\n", value, result);
                        break;
                    case 2:
                        result = fahrenheitToCelsius(value);
                        printf("%.2lf degrees Fahrenheit is equal to %.2lf
degrees Celsius.\n", value, result);
                        break;
                    default:
                        printf("Invalid choice. Please enter a valid
option.\n");
                }
                break;
            case 3:
                printf("Weight Conversion Menu:\n");
                printf("1. Kilograms to Pounds\n");
                printf("2. Pounds to Kilograms\n");
                printf("Enter your choice: ");
                scanf("%d", &choice);

                switch (choice) {
                    case 1:
                        result = kilogramsToPounds(value);
                        printf("%.2lf kilograms is equal to %.2lf pounds.\n",
value, result);
```

```
                        break;
                    case 2:
                        result = poundsToKilograms(value);
                        printf("%.2lf pounds is equal to %.2lf kilograms.\n",
value, result);
                        break;
                    default:
                        printf("Invalid choice. Please enter a valid
option.\n");
                }
                break;
            case 4:
                printf("Volume Conversion Menu:\n");
                printf("1. Liters to Gallons\n");
                printf("2. Gallons to Liters\n");
                printf("Enter your choice: ");
                scanf("%d", &choice);

                switch (choice) {
                    case 1:
                        result = litersToGallons(value);
                        printf("%.2lf liters is equal to %.2lf gallons.\n",
value, result);
                        break;
                    case 2:
                        result = gallonsToLiters(value);
                        printf("%.2lf gallons is equal to %.2lf liters.\n",
value, result);
                        break;
                    default:
                        printf("Invalid choice. Please enter a valid
option.\n");
                }
                break;
            case 5:
                printf("Time Conversion Menu:\n");
                printf("1. Hours to Minutes\n");
                printf("2. Minutes to Hours\n");
                printf("Enter your choice: ");
                scanf("%d", &choice);

                switch (choice) {
                    case 1:
                        result = hoursToMinutes(value);
                        printf("%.2lf hours is equal to %.2lf minutes.\n",
value, result);
                        break;
                    case 2:
```

```
                        result = minutesToHours(value);
                        printf("%.2lf minutes is equal to %.2lf hours.\n",
value, result);
                        break;
                    default:
                        printf("Invalid choice. Please enter a valid
option.\n");
                }
                break;
            case 6:
                printf("Currency Conversion Menu:\n");
                printf("1. USD to Euro\n");
                printf("2. Euro to USD\n");
                printf("Enter your choice: ");
                scanf("%d", &choice);

                switch (choice) {
                    case 1:
                        result = usdToEuro(value);
                        printf("%.2lf USD is equal to %.2lf Euro.\n", value,
result);
                        break;
                    case 2:
                        result = euroToUsd(value);
                        printf("%.2lf Euro is equal to %.2lf USD.\n", value,
result);
                        break;
                    default:
                        printf("Invalid choice. Please enter a valid
option.\n");
                }
                break;
            case 7:
                printf("Area Conversion Menu:\n");
                printf("1. Square Meters to Square Feet\n");
                printf("2. Square Feet to Square Meters\n");
                printf("Enter your choice: ");
                scanf("%d", &choice);

                switch (choice) {
                    case 1:
                        result = squareMetersToSquareFeet(value);
                        printf("%.2lf square meters is equal to %.2lf square
feet.\n", value, result);
                        break;
                    case 2:
                        result = squareFeetToSquareMeters(value);
```

```
                        printf("%.2lf square feet is equal to %.2lf square
meters.\n", value, result);
                        break;
                    default:
                        printf("Invalid choice. Please enter a valid
option.\n");
                }
                break;
            case 8:
                printf("Speed Conversion Menu:\n");
                printf("1. Kilometers per Hour to Miles per Hour\n");
                printf("2. Miles per Hour to Kilometers per Hour\n");
                printf("Enter your choice: ");
                scanf("%d", &choice);

                switch (choice) {
                    case 1:
                        result = kphToMph(value);
                        printf("%.2lf kilometers per hour is equal to %.2lf
miles per hour.\n", value, result);
                        break;
                    case 2:
                        result = mphToKph(value);
                        printf("%.2lf miles per hour is equal to %.2lf
kilometers per hour.\n", value, result);
                        break;
                    default:
                        printf("Invalid choice. Please enter a valid
option.\n");
                }
                break;
        }

    }
    return 0;
}
```

OUTPUT

Unit Converter Menu:

1. Length Conversions

2. Temperature Conversions
3. Weight Conversions
4. Volume Conversions
5. Time Conversions
6. Currency Conversions
7. Area Conversions
8. Speed Conversions
9. Exit
Enter your choice: 5
Enter the value to convert: 200
Time Conversion Menu:
1. Hours to Minutes
2. Minutes to Hours
Enter your choice: 2
200.00 minutes is equal to 3.33 hours.
Unit Converter Menu:
1. Length Conversions
2. Temperature Conversions
3. Weight Conversions
4. Volume Conversions
5. Time Conversions
6. Currency Conversions
7. Area Conversions
8. Speed Conversions
9. Exit
Enter your choice: 9
Exiting the program.

PROFILLING

Flat profile:

Each sample counts as 0.01 seconds.
no time accumulated

% time	cumulative seconds	self seconds	calls	self Ts/call	total Ts/call	name
0.00	0.00	0.00	1	0.00	0.00	minutesToHours

% time — the percentage of the total running time of the program used by this function.

cumulative seconds — a running sum of the number of seconds accounted for by this function and those listed above it.

self seconds — the number of seconds accounted for by this function alone. This is the major sort for this listing.

calls — the number of times this function was invoked, if this function is profiled, else blank.

self ms/call — the average number of milliseconds spent in this function per call, if this function is profiled, else blank.

total ms/call — the average number of milliseconds spent in this function and its descendents per call, if this function is profiled, else blank.

name — the name of the function. This is the minor sort for this listing. The index shows the location of the function in the gprof listing. If the index is in parenthesis it shows where it would appear in the gprof listing if it were to be printed.

Call graph (explanation follows)

granularity: each sample hit covers 4 byte(s) no time propagated

```
index % time    self  children    called     name
                0.00    0.00       1/1          main [17]
[1]     0.0    0.00    0.00       1         minutesToHours [1]
-----------------------------------------------
```

This table describes the call tree of the program, and was sorted by
the total amount of time spent in each function and its children.

Each entry in this table consists of several lines. The line with the
index number at the left hand margin lists the current function.
The lines above it list the functions that called this function,
and the lines below it list the functions this one called.
This line lists:

index A unique number given to each element of the table.
Index numbers are sorted numerically.
The index number is printed next to every function name so
it is easier to look up where the function is in the table.

% time This is the percentage of the `total' time that was spent
in this function and its children. Note that due to
different viewpoints, functions excluded by options, etc,
these numbers will NOT add up to 100%.

selfThis is the total amount of time spent in this function.

children This is the total amount of time propagated into this
function by its children.

called This is the number of times the function was called.
If the function called itself recursively, the number
only includes non-recursive calls, and is followed by
a `+' and the number of recursive calls.

name The name of the current function. The index number is
printed after it. If the function is a member of a
cycle, the cycle number is printed between the

function's name and the index number.

For the function's parents, the fields have the following meanings:

selfThis is the amount of time that was propagated directly from the function into this parent.

children This is the amount of time that was propagated from the function's children into this parent.

called This is the number of times this parent called the function `/' the total number of times the function was called. Recursive calls to the function are not included in the number after the `/'.

name This is the name of the parent. The parent's index number is printed after it. If the parent is a member of a cycle, the cycle number is printed between the name and the index number.

If the parents of the function cannot be determined, the word `<spontaneous>' is printed in the `name' field, and all the other fields are blank.

For the function's children, the fields have the following meanings:

selfThis is the amount of time that was propagated directly from the child into the function.

children This is the amount of time that was propagated from the child's children to the function.

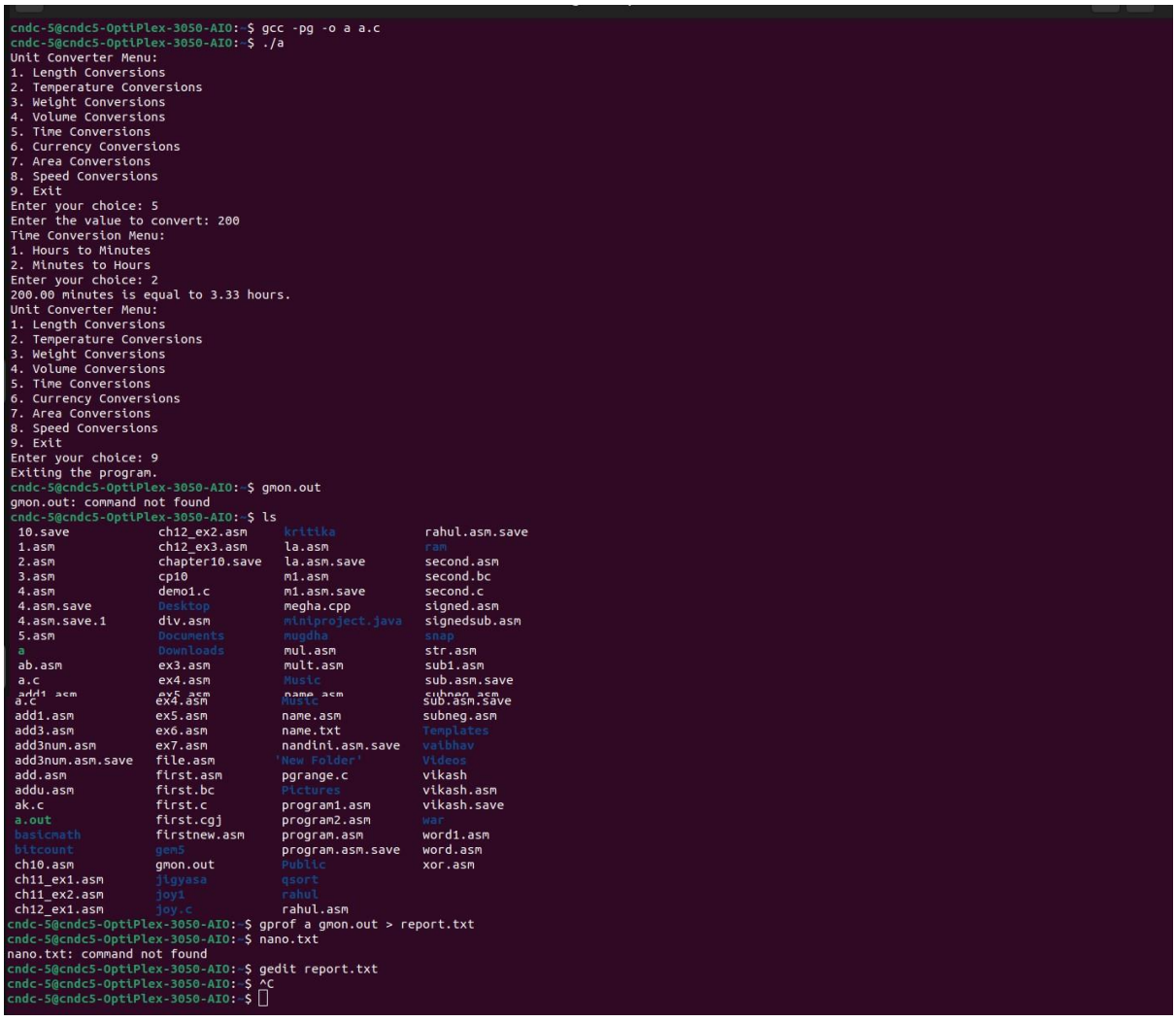
called This is the number of times the function called this child `/' the total number of times the child was called. Recursive calls by the child are not listed in the number after the `/'.

name This is the name of the child. The child's index number is printed after it. If the child is a member of a cycle, the cycle number is printed

between the name and the index number.

If there are any cycles (circles) in the call graph, there is an entry for the cycle-as-a-whole. This entry shows who called the cycle (as parents) and the members of the cycle (as children.) The `+' recursive calls entry shows the number of function calls that were internal to the cycle, and the calls entry for each member shows, for that member, how many times it was called from other members of the cycle.

[1] minutesToHours



DEBUGGING

```
 Reading symbols from a.out...
(gdb) break main
Breakpoint 1 at 0x14e1: file main.c, line 83.
(gdb) r
Starting program: /home/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Breakpoint 1, main () at main.c:83
83 int main() {
(gdb) n
88 printf("Unit Converter Menu:\n");
(gdb) n
Unit Converter Menu:
89 printf("1. Length Conversions\n");
(gdb) n
1. Length Conversions
90 printf("2. Temperature Conversions\n");
(gdb) n
2. Temperature Conversions
91 printf("3. Weight Conversions\n");
(gdb) n
3. Weight Conversions
92 printf("4. Volume Conversions\n");
(gdb) n
4. Volume Conversions
93 printf("5. Time Conversions\n");
(gdb) n
5. Time Conversions
94 printf("6. Currency Conversions\n");
(gdb) n
6. Currency Conversions
95 printf("7. Area Conversions\n");
(gdb) n
7. Area Conversions
96 printf("8. Speed Conversions\n");
(gdb) n
```

8. Speed Conversions
97 printf("9. Exit\n");
(gdb) n
9. Exit
98 printf("Enter your choice: ");
(gdb) n
99 scanf("%d", &choice);
(gdb) n
Enter your choice: 5
101 if (choice == 9) {
(gdb) n
106 if (choice < 1 || choice > 8) {
(gdb) n
111 printf("Enter the value to convert: ");
(gdb) n
112 scanf("%lf", &value);
(gdb) 200
Undefined command: "200". Try "help".
(gdb) n
Enter the value to convert: 200
115 switch (choice) {
(gdb) n
207 printf("Time Conversion Menu:\n");
(gdb) n
Time Conversion Menu:
208 printf("1. Hours to Minutes\n");
(gdb) n
1. Hours to Minutes
209 printf("2. Minutes to Hours\n");
(gdb) n
2. Minutes to Hours
210 printf("Enter your choice: ");
(gdb) n
211 scanf("%d", &choice);
(gdb) n
Enter your choice: 2
213 switch (choice) {
(gdb) n
219 result = minutesToHours(value);
(gdb) n

```
220 printf("%.2lf minutes is equal to %.2lf hours.\n", value, result);
(gdb) n
200.00 minutes is equal to 3.33 hours.
221 break;
(gdb) n
87 while (1) {
(gdb) n
88 printf("Unit Converter Menu:\n");
(gdb) n
Unit Converter Menu:
89 printf("1. Length Conversions\n");
(gdb) n
1. Length Conversions
90 printf("2. Temperature Conversions\n");
(gdb) n
2. Temperature Conversions
91 printf("3. Weight Conversions\n");
(gdb) n
3. Weight Conversions
92 printf("4. Volume Conversions\n");
(gdb) n
4. Volume Conversions
93 printf("5. Time Conversions\n");
(gdb) n
5. Time Conversions
94 printf("6. Currency Conversions\n");
(gdb) n
6. Currency Conversions
95 printf("7. Area Conversions\n");
(gdb) n
7. Area Conversions
96 printf("8. Speed Conversions\n");
(gdb) n
8. Speed Conversions
97 printf("9. Exit\n");
(gdb) n
9. Exit
98 printf("Enter your choice: ");
(gdb) n
99 scanf("%d", &choice);
```

```
(gdb) n
Enter your choice: 9
101 if (choice == 9) {
(gdb) n
102 printf("Exiting the program.\n");
(gdb) n
Exiting the program.
```

LATEX

```
 \documentclass{article}
\usepackage{graphicx} % Required for inserting images
\title{UNIT CONVERTER}
\date{}
\begin{document}
\maketitle
\textbf{{\LARGE AIM :}}\
\\
\\
\textsl{{\Large The aim of the project is to create a versatile unit converter program that allows users to easily convert values between different units of measurement in various categories. The project serves the purpose of providing a user-friendly tool for performing common unit conversions. Here are some key reasons and goals behind designing this project}}\\
\\
\\
\textbf{{\LARGE Utility:}}\\
\\
\textsl{{\Large The project is designed to be a practical utility for users who frequently need to convert values from one unit to another. It simplifies the conversion process and provides a single, easy-to-use interface for various types of conversions, from length and temperature to currency and speed.}}\\
\\
\\
\textbf{{\LARGE Educational Tool:}}\\
\\
\textsl{{\Large This project can serve as an educational tool, helping users understand unit conversions and the conversion factors between different units. By presenting a menu-
```

driven interface, it encourages users to explore and learn about conversions in different categories.}}\\
\\
\\
\\
\textbf{{\LARGE Convenience:}}\\
\\
\\
\textsl{{\Large Instead of relying on manual calculations or web-based converters, this project offers a convenient and efficient way to perform conversions. Users can quickly access the specific conversion they need without the need to leave the program or visit external websites.}}\\
\\
\\
\\
\textbf{{\huge OUTPUT}}\\
\\
{\Large Unit Converter Menu:\\
{\begin{enumerate}
\item Length Conversions
\item Temperature Conversions
\item Weight Conversions
\item Volume Conversions
\item Time Conversions
\item Currency Conversions
\item Area Conversions
\item Speed Conversions
\item Exit
\end{enumerate}}
Enter your choice: 5\\
\\

Enter the value to convert: 200\\
\\
Time Conversion Menu:\\
{\begin{enumerate}
\item Hours to Minutes
\item Minutes to Hours
\end{enumerate}}
Enter your choice: 2\\
\\
200.00 minutes is equal to 3.33 hours.\\
\\
Unit Converter Menu:\\
{\begin{enumerate}
\item Length Conversions
\item Temperature Conversions
\item Weight Conversions
\item Volume Conversions
\item Time Conversions
\item Currency Conversions
\item Area Conversions
\item Speed Conversions
\item Exit
{\end{enumerate}}
Enter your choice: 9\\
\\
Exiting the program.}
\end{document}