NAME - MEET JAIN **E.NO- 0801CS211058**

Starting date/time- 13 November 2022 , 8:00 p.m.
End date/time - 21 November 2022 , 7:30 p.m.
Total Time required - 1 Week
Total line of code - 602 lines in C language and 608 lines in java language
Number of functions - 11

MINI PROJECT

CO-24497: PROGRAMMING PRACTICES

OBJECTIVE OF THE PROJECT

TOPIC : UNIT CONVERTER

- THE PROJECT IS BASED ON THE CONVERSION OF UNITS
- WITH THE HELP OF THIS , YOU CAN CONVERT ONE UNIT TO DIFFERENT UNITS
- FOLLOWING ARE THE CONVERSION
  1) LENGTH (meter to miles and inches)
  2) AREA (meter square to hectares and acres)
  3) VOLUME (cubic meter to cubic yard and cubic foot)
  4) SPEED (meter per second to km per hour and mile per hour)
  5) WEIGHT (gram to ounce and pound)
  6) TEMPERATURE (degree fahrenheit to Celsius and Kelvin)
  7) POWER (Watt to Kilogram meter/second and imperial horsepower)
  8) PRESSURE (atm to Millimeter of mercury and kilopascal)
  9) CURRENCY (US Dollar to Indian Rupees and pound)
  10) ENERGY (joule to kilocalorie and watt hour)
  11) ANGLE (degree to radian and minute of arc)

FUNCTION DESCRIPTION

- length_conversion(): This function is used to convert the length in meter to miles and inches as chosen by user.
- Area_conversion(): This function is used to convert the area in meter square to hectares and acres as chosen by user.
- volume_conversion(): This function is used to convert the volume in cubic meter to cubic yard and cubic foot as chosen by user.
- speed_conversion(): This function is used to convert the speed in meter per second to kilometer per hour and mile per hour as chosen by user.
- weight_conversion(): This function is used to convert the weight in gram to ounce and pound as chosen by user.
- temperature_conversion(): This function is used to convert the temperature in degree fahrenheit to Celsius and Kelvin as chosen by user.
- power_conversion(): This function is used to convert the power in watt to kilogram meter per second and imperial horsepower as chosen by user.
- pressure_conversion():This function is used to convert the pressure in atm to millimeter of mercury and kilopascal as chosen by user.
- currency_conversion(): This function is used to convert the currency in US Dollar to Indian Rupees and pound as chosen by user.
- energy_conversion(): This function is used to convert the energy in joule to kilocalorie and watt hour as chosen by user.
- angle_conversion(): This function is used to convert the angle in degree to radian and minute of arc as chosen by user.

PROFILER REPORT

```
Flat profile:

Each sample counts as 0.01 seconds.
 no time accumulated

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

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

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

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

 total     the average number of milliseconds spent in this
ms/call    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.

Copyright (C) 2012-2022 Free Software Foundation, Inc.
```

```
Copyright (C) 2012-2022 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

                     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 [11]
[1]      0.0    0.00    0.00       1          Area_conversion [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%.

     self       This 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:

    self        This 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:

```

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

    self        This 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.

Copyright (C) 2012-2022 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

Index by function name

   [1] Area_conversion
```

GDB SCREENSHOTS

```
meet@meet-VirtualBox:~/Documents$ gcc -g pro.c
meet@meet-VirtualBox:~/Documents$ gdb a.out
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) break 1
Breakpoint 1 at 0x11d5: file pro.c, line 22.
(gdb) run
Starting program: /home/meet/Documents/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Enter the type of conversion you want to do : l->length , a->area , v->volume , s->speed , w-> weight , t-> temperature , p-> power , k-> pressure , c->currency , e-> energy , m-> ang
le
l

Breakpoint 1, length_conversion () at pro.c:22
22      {
(gdb) n
28          printf("Welcome to the conversion of length\n");
(gdb) n
Welcome to the conversion of length
29          printf("Enter 1-> convert length in meter to miles or\n");
(gdb) n
Enter 1-> convert length in meter to miles or
30          printf("Enter 2-> convert length in meter to inches\n");
(gdb) n
Enter 2-> convert length in meter to inches
31          scanf("%d",&choice_length);    // Taking user input to select choice using scanf
(gdb) n
1
33          if(choice_length==1)  // If choice is 1 , then following code will do the length conversion from meter to miles
(gdb) n
35              while(quit_length_conversion!='q')
```

```
33          if(choice_length==1)  // If choice is 1 , then following code will do the length conversion from meter to miles
(gdb) n
35              while(quit_length_conversion!='q')
(gdb) n
37                  printf("Enter the length in meter you want to convert into miles : ");
(gdb) n
38                  scanf("%f",&length_in_meter);
(gdb) n
Enter the length in meter you want to convert into miles : 34.5
39                  length_in_miles = 0.0006214 * length_in_meter;    // Since 1 meter = 0.0006214 miles
(gdb) n
40                  printf("The value of length in miles = %f\n", length_in_miles);
(gdb) print length_in_meter
$1 = 34.5
(gdb) n
The value of length in miles = 0.021438
41                  printf("If you want to quit , press q otherwise enter any key to continue\n");
(gdb) print length_in_miles
$2 = 0.0214383
(gdb) n
If you want to quit , press q otherwise enter any key to continue
42                  scanf("%s",&quit_length_conversion); // Taking user input that quits or continue the function based on character,user gives
(gdb) n
q
35              while(quit_length_conversion!='q')
(gdb) n
44              printf("Thank you\n");
(gdb) n
Thank you
64      }
(gdb) n
main () at pro.c:601
601         return 0;
(gdb) n
602     }
(gdb) n
__libc_start_call_main (main=main@entry=0x5555555568fd <main>, argc=argc@entry=1, argv=argv@entry=0x7fffffffe138) at ../sysdeps/nptl/libc_start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb)
```

```
meet@meet-VirtualBox:~/Documents$ gcc -g pro.c
meet@meet-VirtualBox:~/Documents$ gdb a.out
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) break Area_conversion
Breakpoint 1 at 0x13ef: file pro.c, line 70.
(gdb) run
Starting program: /home/meet/Documents/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Enter the type of conversion you want to do : l->length , a->area , v->volume , s->speed , w-> weight , t-> temperature , p-> power , k-> pressure , c->currency , e-> energy , m-> ang
le
a

Breakpoint 1, Area_conversion () at pro.c:70
70      {
(gdb) n
77          printf("Welcome to the conversion of Area\n");
(gdb) n
Welcome to the conversion of Area
78          printf("Enter 1-> convert Area in meter square to hectares or\n");
(gdb) n
Enter 1-> convert Area in meter square to hectares or
79          printf("Enter 2-> convert Area in meter square to acres\n");
(gdb) n
Enter 2-> convert Area in meter square to acres
80          scanf("%d",&choice_for_area);    // Taking user input to select choice using scanf
(gdb) n
1
81          if(choice_for_area==1)  // If choice is 1 , then following code will do the area conversion from meter square to hectares
(gdb) n
83              while(quit_Area_conversion!='q')
```

```
(gdb) n
83              while(quit_Area_conversion!='q')
(gdb) n
85                  printf("Enter the area in meter square you want to convert into hectares : ");
(gdb) n
86                  scanf("%f",&area_in_meter_square);
(gdb) n
Enter the area in meter square you want to convert into hectares : 34
87                  area_in_hectares = 0.0001 * area_in_meter_square;    // Since 1 hectare = 0.0001 meter square
(gdb) print area_in_meter_square
$1 = 34
(gdb) n
88                  printf("The value of area in hectares = %f\n", area_in_hectares);
(gdb) n
The value of area in hectares = 0.003400
89                  printf("If you want to quit , press q otherwise enter any key to continue\n");
(gdb) print area_in_hectares
$2 = 0.0034000000000000002
(gdb) n
If you want to quit , press q otherwise enter any key to continue
90                  scanf("%s",&quit_Area_conversion); // Taking user input that quits or continue the function based on character,user gives
(gdb) n
q
83              while(quit_Area_conversion!='q')
(gdb) n
92                  printf("Thank you\n");
(gdb) n
Thank you
113     }
(gdb) n
main () at pro.c:601
601         return 0;
(gdb) n
602     }
(gdb) n
__libc_start_call_main (main=main@entry=0x5555555568fd <main>, argc=argc@entry=1, argv=argv@entry=0x7fffffffe138) at ../sysdeps/nptl/libc_start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb)
```

```
meet@meet-VirtualBox:~/Documents$ gcc -g pro.c
meet@meet-VirtualBox:~/Documents$ gdb a.out
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) break volume_conversion
Breakpoint 1 at 0x1609: file pro.c, line 119.
(gdb) run
Starting program: /home/meet/Documents/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Enter the type of conversion you want to do : l->length , a->area , v->volume , s->speed , w-> weight , t-> temperature , p-> power , k-> pressure , c->currency , e-> energy , m-> ang
le
v

Breakpoint 1, volume_conversion () at pro.c:119
119     {
(gdb) n
125         printf("Welcome to the conversion of volume\n");
(gdb) n
Welcome to the conversion of volume
126         printf("Enter 1-> convert volume in cubic meter to cubic yard or\n");
(gdb) n
Enter 1-> convert volume in cubic meter to cubic yard or
127         printf("Enter 2-> convert volume in cubic meter to cubic foot\n");
(gdb) n
Enter 2-> convert volume in cubic meter to cubic foot
128         scanf("%d",&choice_volume);    // Taking user input to select choice using scanf
(gdb) n
2
130         if(choice_volume==1)   // If choice is 1 , then following code will do the volume conversion from cubic meter to cubic yard
(gdb) n
143         else if(choice_volume==2)  // If choice is 2 , then following code will do the volume conversion from cubic meter to cubic foot
```

```
130         if(choice_volume==1)   // If choice is 1 , then following code will do the volume conversion from cubic meter to cubic yard
(gdb) n
143         else if(choice_volume==2)  // If choice is 2 , then following code will do the volume conversion from cubic meter to cubic foot
(gdb) n
145           while(quit_volume_conversion!='q')
(gdb) n
147             printf("Enter the volume in cubic meter you want to convert into cubic foot : ");
(gdb) n
148             scanf("%f",&volume_in_cubic_meter);
(gdb) n
Enter the volume in cubic meter you want to convert into cubic foot : 344
149             volume_in_cubic_foot = 35.3147248 * volume_in_cubic_meter;  // Since 1 cubic meter = 35.3147248 cubic foot
(gdb) print volume_in_cubic_meter
$1 = 344
(gdb) n
150             printf("The value of volume in cubic foot = %f\n", volume_in_cubic_foot);
(gdb) n
The value of volume in cubic foot = 12148.265331
151             printf("If you want to quit , press q otherwise enter any key to continue\n");
(gdb) print volume_in_cubic_foot
$2 = 12148.2653312
(gdb) n
If you want to quit , press q otherwise enter any key to continue
152             scanf("%s",&quit_volume_conversion);  // Taking user input that quits or continue the function based on character,user gives
(gdb) n
q
145           while(quit_volume_conversion!='q')
(gdb) n
154             printf("Thank you\n");
(gdb) n
Thank you
161     }
(gdb) n
main () at pro.c:601
601         return 0;
(gdb) n
602     }
(gdb) n
__libc_start_call_main (main=main@entry=0x5555555568fd <main>, argc=argc@entry=1, argv=argv@entry=0x7fffffffe138) at ../sysdeps/nptl/libc_start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb)
```

```
meet@meet-VirtualBox:~/Documents$ gcc -g pro.c
meet@meet-VirtualBox:~/Documents$ gdb a.out
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) break speed_conversion
Breakpoint 1 at 0x1823: file pro.c, line 167.
(gdb) run
Starting program: /home/meet/Documents/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Enter the type of conversion you want to do : l->length , a->area , v->volume , s->speed , w-> weight , t-> temperature , p-> power , k-> pressure , c->currency , e-> energy , m-> angle
s

Breakpoint 1, speed_conversion () at pro.c:167
167     {
(gdb) n
173         printf("Welcome to the conversion of speed\n");
(gdb) n
Welcome to the conversion of speed
174         printf("Enter 1-> convert speed in meter per second to kilometer per hour or\n");
(gdb) n
Enter 1-> convert speed in meter per second to kilometer per hour or
175         printf("Enter 2-> convert speed in meter per second to mile per hour\n");
(gdb) n
Enter 2-> convert speed in meter per second to mile per hour
176         scanf("%d",&choice_speed);    // Taking user input to select choice using scanf
(gdb) n
2
178         if(choice_speed==1)  // If choice is 2 , then following code will do the speed conversion from meter per second to kilometer per hour
(gdb) n
191         else if(choice_speed==2)  // If choice is 2 , then following code will do the speed conversion from meter per second to mile per hour
```

```
(gdb) n
191         else if(choice_speed==2)  // If choice is 2 , then following code will do the speed conversion from meter per second to mile per hour
(gdb) n
193           while(quit_speed_conversion!='q')
(gdb) n
195             printf("Enter the speed in meter per second you want to convert into mile per hour : ");
(gdb) n
196             scanf("%f",&speed_in_meter_per_second);
(gdb) n
Enter the speed in meter per second you want to convert into mile per hour : 222
197             speed_in_mile_per_hour = 2.236936 * speed_in_meter_per_second;  // Since 1 meter per second = 2.236936 mile per hour
(gdb) print speed_in_meter_per_second
$1 = 222
(gdb) n
198             printf("The value of speed in mile per hour = %f\n", speed_in_mile_per_hour);
(gdb) n
The value of speed in mile per hour = 496.599792
199             printf("If you want to quit , press q otherwise enter any key to continue\n");
(gdb) print speed_in_mile_per_hour
$2 = 496.59979199999998
(gdb) n
If you want to quit , press q otherwise enter any key to continue
200             scanf("%s",&quit_speed_conversion); // Taking user input that quits or continue the function based on character,user gives
(gdb) n
q
193           while(quit_speed_conversion!='q')
(gdb) n
202             printf("Thank you\n");
(gdb) n
Thank you
209     }
(gdb) n
main () at pro.c:601
601         return 0;
(gdb) n
602     }
(gdb) n
__libc_start_call_main (main=main@entry=0x5555555568fd <main>, argc=argc@entry=1, argv=argv@entry=0x7fffffffe138) at ../sysdeps/nptl/libc_start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb)
```

```
meet@meet-VirtualBox:~/Documents$ gcc -g pro.c
meet@meet-VirtualBox:~/Documents$ gdb a.out
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) break weight_conversion
Breakpoint 1 at 0x1ad: file pro.c, line 215.
(gdb) run
Starting program: /home/meet/Documents/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Enter the type of conversion you want to do : l->length , a->area , v->volume , s->speed , w-> weight , t-> temperature , p-> power , k-> pressure , c->currency , e-> energy , m-> ang
le
w

Breakpoint 1, weight_conversion () at pro.c:215
215     {
(gdb) n
221         printf("Welcome to the conversion of weight\n");
(gdb) n
Welcome to the conversion of weight
222         printf("Enter 1-> convert weight in gram to ounce or\n");
(gdb) n
Enter 1-> convert weight in gram to ounce or
223         printf("Enter 2-> convert weight in gram to pound\n");
(gdb) n
Enter 2-> convert weight in gram to pound
224         scanf("%d",&choice_weight);    // Taking user input to select choice using scanf
(gdb) n
1
226         if(choice_weight==1) // If choice is 1 , then following code will do the weight conversion from gram to ounce
(gdb) n
228           while(quit_weight_conversion!='q')
```

```
(gdb) n
228           while(quit_weight_conversion!='q')
(gdb) n
230             printf("Enter the weight in gram you want to convert into ounce : ");
(gdb) n
231             scanf("%f",&weight_in_gram);
(gdb) n
Enter the weight in gram you want to convert into ounce : 786
232             weight_in_ounce = 0.035 * weight_in_gram;    // Since 1 gram = 0.035 ounce
(gdb) print weight_in_ounce
$1 = 0
(gdb) print weight_in_gram
$2 = 786
(gdb) n
233             printf("The value of weight in ounce = %f\n", weight_in_ounce);
(gdb) n
The value of weight in ounce = 27.510000
234             printf("If you want to quit , press q otherwise enter any key to continue\n");
(gdb) print weight_in_ounce
$3 = 27.510000000000002
(gdb) n
If you want to quit , press q otherwise enter any key to continue
235             scanf("%s",&quit_weight_conversion); // Taking user input that quits or continue the function based on character,user gives
(gdb) n
q
228           while(quit_weight_conversion!='q')
(gdb) n
237             printf("Thank you\n");
(gdb) n
Thank you
257     }
(gdb) n
main () at pro.c:601
601         return 0;
(gdb) n
602     }
(gdb) n
__libc_start_call_main (main=main@entry=0x5555555568fd <main>, argc=argc@entry=1, argv=argv@entry=0x7fffffffe138) at ../sysdeps/nptl/libc_start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb)
```

```
meet@meet-VirtualBox:~/Documents$ gcc -g pro.c
meet@meet-VirtualBox:~/Documents$ gdb a.out
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) break temperature_conversion
Breakpoint 1 at 0x1c57: file pro.c, line 263.
(gdb) run
Starting program: /home/meet/Documents/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Enter the type of conversion you want to do : l->length , a->area , v->volume , s->speed , w-> weight , t-> temperature , p-> power , k-> pressure , c->currency , e-> energy , m-> ang
le
t

Breakpoint 1, temperature_conversion () at pro.c:263
263     {
(gdb) n
269         printf("Welcome to the conversion of temperature\n");
(gdb) n
Welcome to the conversion of temperature
270         printf("Enter 1-> convert temperature in degree fahrenheit to celsius or\n");
(gdb) n
Enter 1-> convert temperature in degree fahrenheit to celsius or
271         printf("Enter 2-> convert temperature in degree fahrenheit to kelvin\n");
(gdb) n
Enter 2-> convert temperature in degree fahrenheit to kelvin
272         scanf("%d",&choice_temp);    // Taking user input to select choice using scanf
(gdb) n
2
274         if(choice_temp==1) // If choice is 1 , then following code will do the temperature conversion from fahrenheit to celsius
(gdb) n
287         else if(choice_temp==2)  // If choice is 2 , then following code will do the temperature conversion from fahrenheit to kelvin
```

```
(gdb) n
287         else if(choice_temp==2)  // If choice is 2 , then following code will do the temperature conversion from fahrenheit to kelvin
(gdb) n
289           while(quit_temperature_conversion!='q')
(gdb) n
291             printf("Enter the temperature in degree fahrenheit you want to convert into kelvin : ");
(gdb) n
292             scanf("%f",&temp_in_degree_fahrenheit);
(gdb) n
Enter the temperature in degree fahrenheit you want to convert into kelvin : 468
293             temp_in_kelvin = ((temp_in_degree_fahrenheit - 32)*(5/9)) + 273.15;  // Kelvin=(Fahrenheit-32)(5/9)+273.15
(gdb) print temp_in_degree_fahrenheit
$1 = 468
(gdb) n
294             printf("The value of temperature in kelvin = %f\n", temp_in_kelvin);
(gdb) n
The value of temperature in kelvin = 273.150000
295             printf("If you want to quit , press q otherwise enter any key to continue\n");
(gdb) print temp_in_kelvin
$2 = 273.14999999999998
(gdb) n
If you want to quit , press q otherwise enter any key to continue
296             scanf("%s",&quit_temperature_conversion); // Taking user input that quits or continue the function based on character,user gives
(gdb) n
q
289           while(quit_temperature_conversion!='q')
(gdb) n
298             printf("Thank you\n");
(gdb) n
Thank you
305     }
(gdb) n
main () at pro.c:601
601         return 0;
(gdb) n
602     }
(gdb) n
__libc_start_call_main (main=main@entry=0x5555555568fd <main>, argc=argc@entry=1, argv=argv@entry=0x7fffffffe138) at ../sysdeps/nptl/libc_start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb)
```

```
meet@meet-VirtualBox:~$ gcc -g pro.c
cc1: fatal error: pro.c: No such file or directory
compilation terminated.
meet@meet-VirtualBox:~$ cd Documents
meet@meet-VirtualBox:~/Documents$ gcc -g pro.c
meet@meet-VirtualBox:~/Documents$ gdb a.out
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) break power_conversion
Breakpoint 1 at 0x1e8f: file pro.c, line 311.
(gdb) run
Starting program: /home/meet/Documents/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Enter the type of conversion you want to do : l->length , a->area , v->volume , s->speed , w-> weight , t-> temperature , p-> power , k-> pressure , c->currency , e-> energy , m-> an
gle
p

Breakpoint 1, power_conversion () at pro.c:311
311     {
(gdb) n
317         printf("Welcome to the conversion of power\n");
(gdb) n
Welcome to the conversion of power
318         printf("Enter 1-> convert power in watt to power in kilogram meter per second or\n");
(gdb) n
Enter 1-> convert power in watt to power in kilogram meter per second or
319         printf("Enter 2-> convert power in watt to power in imperial horsepower\n");
(gdb) n
Enter 2-> convert power in watt to power in imperial horsepower
320         scanf("%d",&choice_power);    // Taking user input to select choice using scanf
(gdb) n
```

```
320         scanf("%d",&choice_power);    // Taking user input to select choice using scanf
(gdb) n
1
322         if(choice_power==1) // If choice is 1 , then following code will do the power conversion from watt to kg meter per second
(gdb) n
324           while(quit_power_conversion!='q')
(gdb) n
326             printf("Enter the power in watt you want to convert into kilogram meter per second : ");
(gdb) n
327             scanf("%f",&power_in_watt); // Taking input for the power in watt
(gdb) n
Enter the power in watt you want to convert into kilogram meter per second : 467
328             power_in_Kilogram_meter_per_second = 0.101971 * power_in_watt;   // Since 1 Watt = 0.101971 kilogram meter per second
(gdb) print power_in_watt
$1 = 467
(gdb) n
329             printf("The value of power in kilogram meter per second = %f\n", power_in_Kilogram_meter_per_second);
(gdb) n
The value of power in kilogram meter per second = 47.620457
330             printf("If you want to quit , press q otherwise enter any key to continue\n");
(gdb) print power_in_Kilogram_meter_per_second
$2 = 47.620457000000002
(gdb) n
If you want to quit , press q otherwise enter any key to continue
331             scanf("%s",&quit_power_conversion); // Taking user input that quits or continue the function based on character,user gives
(gdb) n
q
324           while(quit_power_conversion!='q')
(gdb) n
333             printf("Thank you\n");
(gdb) n
Thank you
353     }
(gdb) n
main () at pro.c:601
601         return 0;
(gdb) n
602     }
(gdb) n
__libc_start_call_main (main=main@entry=0x5555555568fd <main>, argc=argc@entry=1, argv=argv@entry=0x7fffffffe138) at ../sysdeps/nptl/libc_start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb)
```

```
meet@meet-VirtualBox:~/Documents$ gcc -g pro.c
meet@meet-VirtualBox:~/Documents$ gdb a.out
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) break pressure_conversion
Breakpoint 1 at 0x2ba8: file pro.c, line 359.
(gdb) run
Starting program: /home/meet/Documents/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Enter the type of conversion you want to do : l->length , a->area , v->volume , s->speed , w-> weight , t-> temperature , p-> power , k-> pressure , c->currency , e-> energy , m-> an
gle
k

Breakpoint 1, pressure_conversion () at pro.c:359
359     {
(gdb) n
365         printf("Welcome to the conversion of pressure\n");
(gdb) n
Welcome to the conversion of pressure
366         printf("Enter 1-> convert pressure in atm to millimeter of mercury or\n");
(gdb) n
Enter 1-> convert pressure in atm to millimeter of mercury or
367         printf("Enter 2-> convert pressure in atm to pressure in kilopascal\n");
(gdb) n
Enter 2-> convert pressure in atm to pressure in kilopascal
368         scanf("%d",&choice_pressure);    // Taking user input to select choice using scanf
(gdb) n
2
370         if(choice_pressure==1) // If choice is 1 , then following code will do the pressure conversion from atm to mm of Hg
(gdb) n
383         else if(choice_pressure==2)  // If choice is 2 , then following code will do the pressure conversion from atm to kilopascal
```

```
(gdb) n
383         else if(choice_pressure==2)  // If choice is 2 , then following code will do the pressure conversion from atm to kilopascal
(gdb) n
385           while(quit_pressure_conversion!='q')
(gdb) n
387             printf("Enter the pressure in atm you want to convert into kilopascal : ");
(gdb) n
388             scanf("%f",&pressure_in_atm);
(gdb) n
Enter the pressure in atm you want to convert into kilopascal : 55
389             pressure_in_kilopascal = 101.325 * pressure_in_atm;  // Since 1 atm = 101.325 kilopascal
(gdb) print pressure_in_atm
$1 = 55
(gdb) n
390             printf("The value of pressure in kilopascal = %f\n", pressure_in_kilopascal);
(gdb) n
The value of pressure in kilopascal = 5572.875000
391             printf("If you want to quit , press q otherwise enter any key to continue\n");
(gdb) print pressure_in_kilopascal
$2 = 5572.875
(gdb) n
If you want to quit , press q otherwise enter any key to continue
392             scanf("%s",&quit_pressure_conversion); // Taking user input that quits or continue the function based on character,user gives
(gdb) n
q
385           while(quit_pressure_conversion!='q')
(gdb) n
394             printf("Thank you\n");
(gdb) n
Thank you
401     }
(gdb) n
main () at pro.c:601
601     return 0;
(gdb) n
602     }
(gdb) n
__libc_start_call_main (main=main@entry=0x5555555568fd <main>, argc=argc@entry=1, argv=argv@entry=0x7fffffffe138) at ../sysdeps/nptl/libc_start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb)
```

```
meet@meet-VirtualBox:~/Documents$ gcc -g pro.c
meet@meet-VirtualBox:~/Documents$ gdb a.out
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) break currency_conversion
Breakpoint 1 at 0x22bf: file pro.c, line 407.
(gdb) run
Starting program: /home/meet/Documents/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Enter the type of conversion you want to do : l->length , a->area , v->volume , s->speed , w-> weight , t-> temperature , p-> power , k-> pressure , c->currency , e-> energy , m-> an
gle
c

Breakpoint 1, currency_conversion () at pro.c:407
407     {
(gdb) n
413         printf("Welcome to the conversion of currency\n");
(gdb) n
Welcome to the conversion of currency
414         printf("Enter 1-> convert currency in US dollar to Indian rupees or\n");
(gdb) n
Enter 1-> convert currency in US dollar to Indian rupees or
415         printf("Enter 2-> convert currency in US dollar to pound\n");
(gdb) n
Enter 2-> convert currency in US dollar to pound
416         scanf("%d",&choice_currency);    // Taking user input to select choice using scanf
(gdb) n
1
418         if(choice_currency==1)  // If choice is 1 , then following code will do the currency conversion from US dollar to rupees
(gdb) n
420           while(quit_currency_conversion!='q')
```

```
420         while(quit_currency_conversion!='q')
(gdb) n
422           printf("Enter the currency in US Dollar you want to convert into Indian Rupees : ");
(gdb) n
423           scanf("%f",&currency_in_US_dollar); // Taking input for the currency in US dollar
(gdb) n
Enter the currency in US Dollar you want to convert into Indian Rupees : 666
424           currency_in_rupees = 80.975 * currency_in_US_dollar;   // Since 1 US Dollar = 80.975 rupees
(gdb) print currency_in_US_dollar
$1 = 666
(gdb) n
425           printf("The value of currency in Indian rupees = %f\n", currency_in_rupees);
(gdb) n
The value of currency in Indian rupees = 53929.350000
426           printf("If you want to quit , press q otherwise enter any key to continue\n");
(gdb) print currency_in_rupees
$2 = 53929.349999999999
(gdb) n
If you want to quit , press q otherwise enter any key to continue
427           scanf("%s",&quit_currency_conversion);  // Taking user input that quits or continue the function based on character,user gives
(gdb) n
q
420         while(quit_currency_conversion!='q')
(gdb) n
429           printf("Thank you\n");
(gdb) n
Thank you
449     }
(gdb) n
main () at pro.c:601
601         return 0;
(gdb) n
602     }
(gdb) n
__libc_start_call_main (main=main@entry=0x5555555568fd <main>, argc=argc@entry=1, argv=argv@entry=0x7fffffffe138) at ../sysdeps/nptl/libc_start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb)
```

```
meet@meet-VirtualBox:~/Documents$ gcc -g pro.c
meet@meet-VirtualBox:~/Documents$ gdb a.out
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) break energy_conversion
Breakpoint 1 at 0x24d9: file pro.c, line 455.
(gdb) n
The program is not being run.
(gdb) run
Starting program: /home/meet/Documents/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Enter the type of conversion you want to do : l->length , a->area , v->volume , s->speed , w-> weight , t-> temperature , p-> power , k-> pressure , c->currency , e-> energy , m-> an
gle
e

Breakpoint 1, energy_conversion () at pro.c:455
455     {
(gdb) n
461         printf("Welcome to the conversion of energy\n");
(gdb) n
Welcome to the conversion of energy
462         printf("Enter 1-> convert energy in joule to kilocalories or\n");
(gdb) n
Enter 1-> convert energy in joule to kilocalories or
463         printf("Enter 2-> convert energy in joule to watt hour\n");
(gdb) n
Enter 2-> convert energy in joule to watt hour
464         scanf("%d",&choice_energy);    // Taking user input to select choice using scanf
(gdb) n
2
466         if(choice_energy==1)   // If choice is 1 , then following code will do the energy conversion from joule to kilocalories
```

```
2
466         if(choice_energy==1)   // If choice is 1 , then following code will do the energy conversion from joule to kilocalories
(gdb) n
479         else if(choice_energy==2) // If choice is 2 , then following code will do the energy conversion from joule to watt hour
(gdb) n
481           while(quit_energy_conversion!='q')
(gdb) n
483           printf("Enter the energy in joules you want to convert into watt hour : ");
(gdb) n
484           scanf("%f",&energy_in_joule);      // Taking input for the energy in joules
(gdb) n
Enter the energy in joules you want to convert into watt hour : 864
485           energy_in_watt_hour = 0.000277 * energy_in_joule;  // Since 1 joule = 0.000277 watt hour
(gdb) n
486           printf("The value of energy in watt hour = %f\n", energy_in_watt_hour);
(gdb) print energy_in_joule
$1 = 864
(gdb) n
The value of energy in watt hour = 0.239328
487           printf("If you want to quit , press q otherwise enter any key to continue\n");
(gdb) n
If you want to quit , press q otherwise enter any key to continue
488           scanf("%s",&quit_energy_conversion); // Taking user input that quits or continue the function based on character,user gives
(gdb) print energy_in_watt_hour
$2 = 0.23932800000000001
(gdb) n
q
481           while(quit_energy_conversion!='q')
(gdb) n
490           printf("Thank you\n");
(gdb) n
Thank you
497     }
(gdb) n
main () at pro.c:601
601         return 0;
(gdb) n
602     }
(gdb) n
__libc_start_call_main (main=main@entry=0x5555555568fd <main>, argc=argc@entry=1, argv=argv@entry=0x7fffffffe138) at ../sysdeps/nptl/libc_start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb)
```

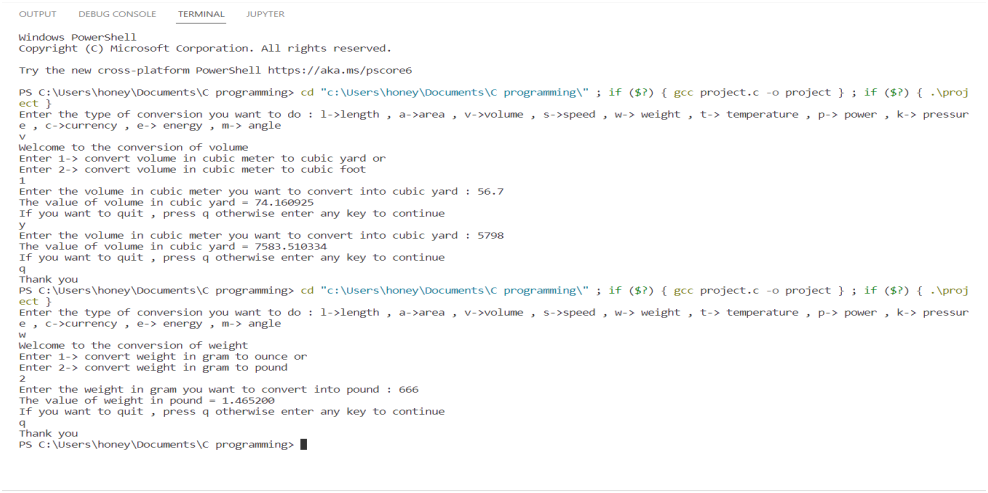
```
meet@meet-VirtualBox:~/Documents$ gcc -g pro.c
meet@meet-VirtualBox:~/Documents$ gdb a.out
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) break angle_conversion
Breakpoint 1 at 0x2ef0: file pro.c, line 503.
(gdb) run
Starting program: /home/meet/Documents/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Enter the type of conversion you want to do : l->length , a->area , v->volume , s->speed , w-> weight , t-> temperature , p-> power , k-> pressure , c->currency , e-> energy , m-> an
gle
m

Breakpoint 1, angle_conversion () at pro.c:503
503     {
(gdb) n
509         printf("Welcome to the conversion of angle\n");
(gdb) n
Welcome to the conversion of angle
510         printf("Enter 1-> convert angle in degree to radian or\n");
(gdb) n
Enter 1-> convert angle in degree to radian or
511         printf("Enter 2-> convert angle in degree to minute of arc\n");
(gdb) n
Enter 2-> convert angle in degree to minute of arc
512         scanf("%d",&choice_angle);    // Taking user input to select choice using scanf
(gdb) n
1
514         if(choice_angle==1) // If choice is 1 , then following code will do the angle conversion from degree to radian
(gdb) n
516           while(quit_angle_conversion!='q')
```

```
(gdb) n
516           while(quit_angle_conversion!='q')
(gdb) n
518             printf("Enter the angle in degree you want to convert into radian : ");
(gdb) n
519             scanf("%f",&angle_in_degree); // Taking input for the angle in degree
(gdb) n
Enter the angle in degree you want to convert into radian : 60
520             angle_in_radian = 0.01745 * angle_in_degree;   // Since 1 degree = 0.01745 radian
(gdb) print angle_in_degree
$1 = 60
(gdb) n
521             printf("The value of angle in radian = %f\n", angle_in_radian);
(gdb) n
The value of angle in radian = 1.047000
522             printf("If you want to quit , press q otherwise enter any key to continue\n");
(gdb) n
If you want to quit , press q otherwise enter any key to continue
523             scanf("%s",&quit_angle_conversion); // Taking user input that quits or continue the function based on character,user gives
(gdb) n
q
516           while(quit_angle_conversion!='q')
(gdb) print angle_in_radian
$2 = 1.0469999999999999
(gdb) n
525             printf("Thank you\n");
(gdb) n
Thank you
545     }
(gdb) n
main () at pro.c:601
601         return 0;
(gdb) n
602     }
(gdb) n
__libc_start_call_main (main=main@entry=0x5555555568fd <main>, argc=argc@entry=1, argv=argv@entry=0x7fffffffe138) at ../sysdeps/nptl/libc_start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb)
```

OUTPUT SCREENSHOTS

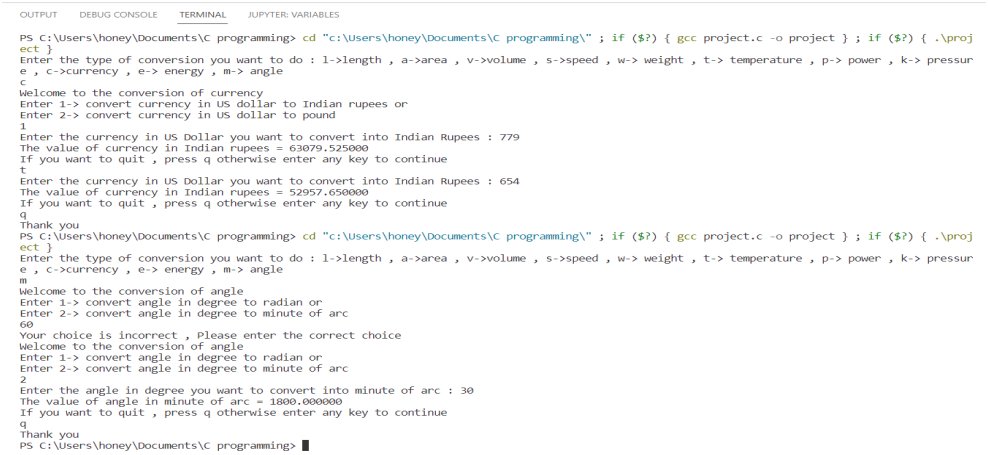


```
OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\honey\Documents\C programming> cd "c:\Users\honey\Documents\C programming\" ; if ($?) { gcc project.c -o project } ; if ($?) { .\proj
ect }
Enter the type of conversion you want to do : l->length , a->area , v->volume , s->speed , w-> weight , t-> temperature , p-> power , k-> pressur
e , c->currency , e-> energy , m-> angle
v
Welcome to the conversion of volume
Enter 1-> convert volume in cubic meter to cubic yard or
Enter 2-> convert volume in cubic meter to cubic foot
1
Enter the volume in cubic meter you want to convert into cubic yard : 56.7
The value of volume in cubic yard = 74.160925
If you want to quit , press q otherwise enter any key to continue
y
Enter the volume in cubic meter you want to convert into cubic yard : 5798
The value of volume in cubic yard = 7583.510334
If you want to quit , press q otherwise enter any key to continue
q
Thank you
PS C:\Users\honey\Documents\C programming> cd "c:\Users\honey\Documents\C programming\" ; if ($?) { gcc project.c -o project } ; if ($?) { .\proj
ect }
Enter the type of conversion you want to do : l->length , a->area , v->volume , s->speed , w-> weight , t-> temperature , p-> power , k-> pressur
e , c->currency , e-> energy , m-> angle
w
Welcome to the conversion of weight
Enter 1-> convert weight in gram to ounce or
Enter 2-> convert weight in gram to pound
2
Enter the weight in gram you want to convert into pound : 666
The value of weight in pound = 1.465200
If you want to quit , press q otherwise enter any key to continue
q
Thank you
PS C:\Users\honey\Documents\C programming>
```



```
OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER: VARIABLES

PS C:\Users\honey\Documents\C programming> cd "c:\Users\honey\Documents\C programming\" ; if ($?) { gcc project.c -o project } ; if ($?) { .\proj
ect }
Enter the type of conversion you want to do : l->length , a->area , v->volume , s->speed , w-> weight , t-> temperature , p-> power , k-> pressur
e , c->currency , e-> energy , m-> angle
c
Welcome to the conversion of currency
Enter 1-> convert currency in US dollar to Indian rupees or
Enter 2-> convert currency in US dollar to pound
1
Enter the currency in US Dollar you want to convert into Indian Rupees : 779
The value of currency in Indian rupees = 63079.525000
If you want to quit , press q otherwise enter any key to continue
t
Enter the currency in US Dollar you want to convert into Indian Rupees : 654
The value of currency in Indian rupees = 52957.650000
If you want to quit , press q otherwise enter any key to continue
q
Thank you
PS C:\Users\honey\Documents\C programming> cd "c:\Users\honey\Documents\C programming\" ; if ($?) { gcc project.c -o project } ; if ($?) { .\proj
ect }
Enter the type of conversion you want to do : l->length , a->area , v->volume , s->speed , w-> weight , t-> temperature , p-> power , k-> pressur
e , c->currency , e-> energy , m-> angle
m
Welcome to the conversion of angle
Enter 1-> convert angle in degree to radian or
Enter 2-> convert angle in degree to minute of arc
60
Your choice is incorrect , Please enter the correct choice
Welcome to the conversion of angle
Enter 1-> convert angle in degree to radian or
Enter 2-> convert angle in degree to minute of arc
2
Enter the angle in degree you want to convert into minute of arc : 30
The value of angle in minute of arc = 1800.000000
If you want to quit , press q otherwise enter any key to continue
q
Thank you
PS C:\Users\honey\Documents\C programming>
```