Exercise 1

My Project

Main Page | Files | Search

Macros | Functions

area.c File Reference

```
#include <stdio.h>
```

Include dependency graph for area.c:

Macros

#define **PI** 3.1415
Mathematical constant PI. More...

#define **RADIUS_M** 7.82
Radius in meters. More...

Functions

float **calculate_area** (float radius)
float **calculate_perimeter** (float radius)
int **main** ()

Macro Definition Documentation

Function Documentation

◆ calculate_area()

float calculate_area (float radius)

Calculates the Area of the circle. Formula: Area = PI*r^2

Parameters

[in] **radius**

[out] **area**

◆ calculate_perimeter()

float calculate_perimeter (float radius)

Calculates the Perimeter of the circle. Formula: Perimeter = 2*PI*r

Parameters

[in] **radius**

[out] **perimeter**

Exercise 6

Q1)
Write the function strindex(s,t) which returns the position of the rightmost occurrence of t in s, or -1 if there is none.

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise1$ cd out
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise1/out$ ls
main
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise1/out$ ./main
Found: 6
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise1/out$
```

Q2)
Extend atof to handle scientific notation of the form
123.45e-6
where a floating-point number may be followed by e or E and an optionally signed exponent.

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise2/out$ ./main
Enter string:
123.45e-6
Length = 9
Floating-point value = 0.000123vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise2/out$
```

Q3)
Given the basic framework, it's straightforward to extend the calculator. Add the modulus (%) operator and provisions for negative Numbers.

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise3/out$ ./main
10 20 30 -5 -2 +
        -7
```

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise3/out$ ./main
200 10 %
        0
100 0.0 %
erro:zero divisor
        100
```

Q4)
Add access to library functions like sin, exp, and pow.

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise4/out$ ./main
Value 2 ^ 3 = 8.000000
exponential value = 162754.791419
The cosine of 60.000000 is 0.500000 degrees
The sine of 60.000000 is 0.866025 degrees
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise4/out$
```

Q5)
Add the commands to print the top elements of the stack without popping, to duplicate it, and to swap the top two elements. Add a command to clear the stack.

```
10 20 30 40 50 ?
topmost=50.000000
secondmost=40.000000
```

```
 100 120 130 140 d
 topmost=140.000000second topmost=140.000000
```

```
150 160 170 190 s
top=190.000000 and second=170.000000
new topmost=170.000000
new secondmost=190.000000       170
```

```
c

stack cleared
error: stack empty
```

Q6)
Write a routine ungets(s) that will push back an entire string onto the input. Should ungets know about buf and bufp, or should it just use ungetch?

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise6/out$ ./main
enter string:
hello world

Characters retrieved from the buffer:
dlrow olleh
```

Q7)
Adapt the ideas of printd to write a recursive version of itoa; that is, convert an integer into a string by calling a recursive routine.

```
vanshit@66JC9F2-Desk:~/in
123vanshit@66JC9F2-Desk:~
```

Q8)
Write a recursive version of the function reverse(s), which reverses the string s in place.

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise8/out$ ./main
dlrow ollehvanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise8/out$
```

Q9)

Suppose that there will never be more than one character of pushback. Modify getch and ungetch accordingly.

```
vrk@vrk-VirtualBox:~/testingpurpose$ ./obj
Enter character: h
h
No space left for ungetch
vrk@vrk-VirtualBox:~/testingpurpose$ ./obj
Enter character: hello
h
No space left for ungetch
vrk@vrk-VirtualBox:~/testingpurpose$
```

Q10)
Our getch and ungetch do not handle a pushed-back EOF correctly. Decide what their properties ought to be if an EOF is pushed back, then implement your design.

```
vrk@vrk-VirtualBox:~/testingpurpose$ ./obj
Enter a string: hello
EOF added
String in the buffer: olleh
```

Q11)
An alternate organization uses getline to read an entire input line; this makes getch and ungetch unnecessary. Revise the calculator to use this approach.

```
vrk@vrk-VirtualBox:~/testingpurpose$ ./obj
11 12 14 15 18 20 + ?
topmost=38.000000
secondmost=15.000000
```

Q12)
Modify getop so that it doesn't need to use ungetch.

```
vrk@vrk-VirtualBox:~/te
12 13 14 5 6 7 * ?
topmost=42.000000
secondmost=5.000000
```

Q13)
Define a macro swap(t,x,y) that interchanges two arguments of type t.

```
Before swap: a = 11, b = 10
After swap: a = 10, b = 11
```

Q14)
Add commands for handling variables. (It's easy to provide twenty-six variables with single-letter names.) Add a variable for the most recently printed value.

```
#include <ctype.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAXLINE 100
#define NUMBER '0'
#define VARIABLE 'b' // Assume single-letter variable names

#define BUFSIZE 100
#define MAXVAL 100

int sp = 0;
int bufp = 0;

double val[MAXVAL];
char buf[BUFSIZE];

char variables[26]; // Array to store variables ('a' to 'z')
```

```
int mgetLine(char[], int);

void push(double);
double pop(void);
int getop(char[]);

/* reverse polish calculator */
int main(void)
{
    int type;
    double op2;
    char s[MAXLINE];

    while ((type = getop(s)) != EOF)
    {
        switch (type)
        {
        case NUMBER:
            push(atof(s));
            break;
        case VARIABLE: // Handle variables
            push(variables[s[0]]);
            break;
        case '+':
            push(pop() + pop());
            break;
        case '*':
            push(pop() * pop());
            break;
        case '-':
            op2 = pop();
            push(pop() - op2);
            break;
        case '/':
            op2 = pop();
            if (op2 != 0.0)
                push(pop() / op2);
```

```
    else
        printf("error: zero divisor\n");
    break;
case '%':
    op2 = pop();
    if (op2 != 0.0)
        push(fmod(pop(), op2));
    else
        printf("error: zero divisor\n");
    break;
case '?':
    if (sp >= 2)
    {
        if (val[sp - 1] == VARIABLE) {
            printf("topmost=%c\n", variables[s[0]]);
            printf("secondmost=%lf\n", val[sp - 2]);
        }
        else
        {
            printf("topmost=%lf\n", val[sp - 1]);
            printf("secondmost=%lf\n", val[sp - 2]);
        }
    }
    else
    {
        printf("insufficient elements\n");
    }
    break;
case 'd':
    if (sp > 0)
    {
        double d = val[sp - 1];
        push(d);
        printf("topmost=%lf\n", val[sp - 1]);
        printf("second topmost=%lf\n", val[sp - 2]);
    }
    else
```

```
            {
                printf("error: stack empty\n");
            }
            break;
        case 's':
            if (sp >= 2)
            {
                printf("top=%lf and second=%lf\n", val[sp - 1], val[sp - 2]);
                double topmost = pop();       // Pop the topmost element
                double secondTopmost = pop(); // Pop the second topmost element
                push(topmost);
                push(secondTopmost);
                printf("new topmost=%lf\n", val[sp - 1]);
                printf("new secondmost=%lf\n", val[sp - 2]);
            }
            else
            {
                printf("insufficient elements\n");
            }
            break;
        case 'c':
            printf("stack cleared\n");
            sp = 0;
            break;
        default:
            printf("error: unknown command %s\n", s);
            break;
        }
    }
    return 0;
}

void push(double f)
{
    if (sp < MAXVAL)
        val[sp++] = f;
    else
```

```
        printf("error: stack full, can't push %g\n", f);
}

double pop(void)
{
    if (sp > 0)
        return val[--sp];
    else
    {
        printf("error: stack empty\n");
        return 0.0;
    }
}

int mgetLine(char s[], int lim)
{
    int c, i;

    for (i = 0; i < lim - 1 && (c = getchar()) != EOF && c != '\n'; ++i)
    {
        s[i] = c;
    }

    if (c == '\n')
    {
        s[i] = c;
        ++i;
    }

    s[i] = '\0';

    return i;
}

char line[MAXLINE];
int li = 0;
```

```
int getop(char s[])
{
    int i, c;

    if (line[li] == '\0')
        if (mgetLine(line, MAXLINE) == 0)
            return EOF;
        else
            li = 0;

    /* Skip whitespace */
    while ((s[0] = c = line[li++]) == ' ' || c == '\t')
    {
        ;
    }

    if (!isdigit(c) && c != '.' && c != '-')
        return c;

    i = 0;

    if (isdigit(c))
        while (isdigit(s[++i] = c = line[li++]))
            ;

    if (c == '.')
        while (isdigit(s[++i] = c = line[li++]))
            ;
    s[i] = '\0';
    li--;

    if (isalpha(s[0]) && strlen(s) == 1) // Check if it's a single-letter variable
        return VARIABLE;

    return NUMBER;
}
```

```
11 13 14 15 16 b ?
topmost=0.000000
secondmost=16.000000
```

Exercise 7

Q1)
As written, getint treats a + or - not followed by a digit as a valid representation of zero. Fix it to push such a character back on the input.

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandArray/Exercise1/out$ ./main
12
+45
-35
6
9
12      45      -35     6       9       vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandArray/Exercise1/out$
```

Q2)
Write getfloat, the floating-point analog of getint. What type does getfloat return as its function value?

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandarray/Exercise2/out$ ./main
3.14
3.140000        vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandarray/Exercise2/out$
```

Q3)
Write a pointer version of the function strcat that we showed in Chapter 2: strcat(s,t) copies the string t to the end of s. ?

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandarray/Exercise3$ make all
gcc app.c -o main
mv main ./out
gcc app.c -c
mv *.o ./build
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandarray/Exercise3$ cd out
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandarray/Exercise3/out$ ls
main
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandarray/Exercise3/out$ ./main
Concatenated String: helloworld ok
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandarray/Exercise3/out$
```

Q4)
Write versions of the library functions strncpy, strncat, and strncmp, which operate on at most the first n characters of their argument strings. For example, strncpy(s,t,n) copies at most n characters of t to s. Full descriptions are in Appendix B.

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandarray/Exercise4/out$ ./main
Strings are not equal
world
Concatenated String: helloworld o
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandarray/Exercise4/out$
```

Q5)
Rewrite appropriate programs from earlier chapters and exercises with pointers instead of array indexing. Good possibilities include getline (Chapters 1 and 4), atoi, itoa, and their variants (Chapters 2, 3, and 4), reverse (Chapter 3), and strindex and getop (Chapter 4).

```
vanshit@66JC9F2-Desk:~/in
Found: 6
hello
Length of the string: 5

123
dlrow olleh
8 9 10 11 12 ?
topmost=12.000000
secondmost=11.000000    1
```

Exercise -8

Exercise 1

Our version of getword does not properly handle underscores, string constants, comments, or preprocessor control lines. Write a better version.

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/StructureandUnion/Exercise1/out$ ./main
x=y+z
Word: x
Word: =
Word: y
Word: +
Word: z
```

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/StructureandUnion/Exercise1/out$ ./main
#include <stdio.h>
Word: include <stdio.h>
```

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/StructureandUnion/Exercise1/out$ ./main
/* this is a comment */
Word: /* this is a comment */
```

Exercise 2

Write a program that prints the distinct words in its input sorted into decreasing order of frequency of occurrence. Precede each word by its count.

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/StructureandUnion/Exercise2/out$ ./main
Word Count in Decreasing Order:
portal--->      count:3
A--->   count:2
science--->     count:2
computer--->    count:1
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/StructureandUnion/Exercise2/out$
```

Exercise 3

Write a cross-referencer that prints a list of all words in a document, and for each word, a list of the line numbers on which it occurs. Remove noise words like ``the,'' ``and,'' and so on.

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/StructureandUnion/Exercise3/out$ ./main
The word 'Hello' belongs to line 1
The word 'world' belongs to line 1
The word 'good' belongs to line 2
The word 'evening' belongs to line 2
The word 'afternoon' belongs to line 3
The word 'is' belongs to line 3
The word 'blessed' belongs to line 3
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/StructureandUnion/Exercise3/out$
```

Exercise 4

Write a function undef that will remove a name and definition from the table maintained by lookup and install.

```
0   Moph    ---
1   ---
2   ---
3   ---
4   ---
5   ---
6   Jacob   ---
7   Kate    ---
8   ---
9   lilly   ---
```

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/StructureandUnion/Exercise4/out$ ./main
found lilly
deleted word=lilly
0       Moph    ---
1       ---
2       ---
3       ---
4       ---
5       ---
6       Jacob   ---
7       Kate    ---
8       ---
9       ---
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/StructureandUnion/Exercise4/out$
```

Exercise 5

Implement a simple version of the #define processor (i.e., no arguments) suitable for use with C programs, based on the routines of this section. You may also find getch and ungetch helpful.

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/StructureandUnion/Exercise5/out$ ./main
0       Moph    ---
1       ---
2       ---
3       ---
4       jake    ---
5       ---
6       Jacob   ---
7       Kate    ---
8       ---
9       lilly   ---
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/StructureandUnion/Exercise5/out$
```

Exercise 10

Q1)
Write a program that converts upper case to lower or lower case to upper, depending on the name it is invoked with, as found in argv[0].

```
vrk@vrk-VirtualBox:~/testingpurpose$ ./main lower
Enter text to convert: HELLO
hello
vrk@vrk-VirtualBox:~/testingpurpose$ ./main upper
Enter text to convert: hello
HELLO
```

Q2)
Write a program that will print arbitrary input in a sensible way. As a minimum, it should print non-graphic characters in octal or hexadecimal according to local custom, and break long text lines.

```
vrk@vrk-VirtualBox:~/testingpurpose$ ./main
This is a longer str
ing with more \011 t
han 20 characters.\012
vrk@vrk-VirtualBox:~/testingpurpose$
```

Q3)
Revise minprintf to handle more of the other facilities of printf.

```
vrk@vrk-VirtualBox:~/testingpurpose$ ./main
Hello,World
10
13
hello
11.000000vrk@vrk-VirtualBox:~/testingpurpose$
```

Q4)
Write a program to compare two files, printing the first line where they differ.

```
vrk@vrk-VirtualBox:~/testingpurpose/fileio$ ./main
this is of file1.txt:new is change
this is of file2.txt:new is the change
files are not identicalvrk@vrk-VirtualBox:~/testingpurpose/fileio$
```

Q5)
Write a program to print a set of files, starting each new one on a new page, with a title and a running page count for each file.

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/InputandOutput/Exercise5/out$ ./main

-----
Title: /home/vanshit/internship/git/vanshit_kamdar_idp/InputandOutput/Exercise5/file1.txt
Page: 1
hello world
changes meet
new people

-----
Title: /home/vanshit/internship/git/vanshit_kamdar_idp/InputandOutput/Exercise5/file2.txt
Page: 2
nice to meet you
good to see you
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/InputandOutput/Exercise5/out$
```

Q6)
Modify the pattern finding program of Chapter 5 to take its input from a set of named files or, if no files are named as arguments, from the standard input. Should the file name be printed when a matching line is found?

```
vrk@vrk-VirtualBox:~/testingpurpose/fileio$ cat file1.txt
hello world is
new is change
good deeds work
vrk@vrk-VirtualBox:~/testingpurpose/fileio$ ./main file1.txt
Pattern not foundvrk@vrk-VirtualBox:~/testingpurpose/fileio$
```

```
vrk@vrk-VirtualBox:~/testingpurpose/fileio$ cat file1.txt
hello world is
new is change
good deeds work
ould is work
vrk@vrk-VirtualBox:~/testingpurpose/fileio$ gcc app.c -o main
vrk@vrk-VirtualBox:~/testingpurpose/fileio$ vi app.c
vrk@vrk-VirtualBox:~/testingpurpose/fileio$ gcc app.c -o main
vrk@vrk-VirtualBox:~/testingpurpose/fileio$ ./main file1.txt
Pattern found at line:ould is work
```

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/InputandOutput/Exercise6/out$ ./main /home/vanshit/internship/git/vanshit_kamdar_idp/InputandOutput/Exercise6/file1.txt
Pattern found at line:ould is work
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/InputandOutput/Exercise6/out$
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/InputandOutput/Exercise6/out$
```

Q7)
Functions like isupper can be implemented to save space or to save time. Explore both Possibilities.

```
vrk@vrk-VirtualBox:~/testingpurpose/fileio$ ./main

Number of upper case present in the sentence is : 3
vrk@vrk-VirtualBox:~/testingpurpose/fileio$
```

Q8)
Write a private version of scanf analogous to minprintf from the previous section.

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/InputandOutput/Exercise8/out$ ./main
enter integer number=10
number= 10

enter float number=11.56
number= 11.560000

enter string number=hello
string= hello
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/InputandOutput/Exercise8/out$
```

Exercise 11

Static library:

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise1$ ar rcs libsample.a adddemo.o add.o div.o mul.o sub.o
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise1$ ls
add.c  adddemo.c  adddemo.o  add.o  div.c  div.o  head.h  libsample.a  mul.c  mul.o  sub.c  sub.o
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise1$ gcc add.o -o main -L . -lsample
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise1$ ls
add.c  adddemo.c  adddemo.o  add.o  div.c  div.o  head.h  libsample.a  main  mul.c  mul.o  sub.c  sub.o
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise1$ ./main
```

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise1$ ./mai

 10 + 20 = 30
 10 - 20 = -10
 10 * 20 = 200
 10 / 20 = 0vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exerci
```

Dynamic library:

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ vi adddemo.c
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ vi add.c
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ vi sub.c
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ vi mul.c
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ vi mul.c
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ vi div.c
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ vi adddemo.c
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ vi head.h
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ ls
add.c  adddemo.c  div.c  head.h  mul.c  sub.c
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ gcc -c add.c sub.c div.c mul.c -fpic
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ ls
add.c  adddemo.c  add.o  div.c  div.o  head.h  mul.c  mul.o  sub.c  sub.o
```

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ gcc *.o -shared -o libsample.so
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ ls
add.c  adddemo.c  add.o  div.c  div.o  head.h  libsample.so  mul.c  mul.o  sub.c  sub.o
```

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ gcc adddemo.c -c
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ ls
add.c  adddemo.c  adddemo.o  add.o  div.c  div.o  head.h  libsample.so  mul.c  mul.o  sub.c  sub.o
```

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ gcc -o main adddemo.o -L. -lsample
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ ls
add.c  adddemo.c  adddemo.o  add.o  div.c  div.o  head.h  libsample.so  main  mul.c  mul.o  sub.c  sub.o
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ ./main
./main: error while loading shared libraries: libsample.so: cannot open shared object file: No such file or directory
```

```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ export LD_LIBRARY_PATH=:/home/vanshit/internship/git/vanshit_kamdar_idp/stat
icanddynamiclibrary/exercise2
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ ./main
10 + 20 = 30
10 - 20 = -10
10 * 20 = 200
10 / 20 = 0vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$
```