

Doxygen

file:///home/vanshit/internship/git/vanshit_kamdar_idp/Doxygen/html/files.html

My Project

Main PageFiles

Search

File List

Here is a list of all files with brief descriptions:

area.c

main.c

Generated by [doxygen](#) 1.8.17

My Project

My Project

Main PageFiles

Search

area.c File Reference

Macros | Functions

#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

Doxygen GUI frontend +

File

Settings

Help

Step 1: Specify the working directory from which doxygen will run

/home/vanshit/internship/git/vanshit_kamdar_idp/Doxygen

Select...

Step 2: Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation

Wizard

Expert

Run

Topics

Project

Mode

Output

Diagrams

Provide some information about the project you are documenting

Project name:

My Project

Project synopsis:

Project version or id:

Project logo:

Select...

No Project logo selected.

Specify the directory to scan for source code

Source code directory:

ship/git/vanshit_kamdar_idp/Doxygen

Select...

☒ Scan recursively

Specify the directory where doxygen should put the generated documentation

Destination directory:

ship/git/vanshit_kamdar_idp/Doxygen

Select...

Previous

Next

Function Documentation

◆ calculate_area()

float calculate_area (float **radius**)

Calculates the Area of the circle. Formula: $\text{Area} = \text{PI} * r^2$

Parameters

[in] **radius**

[out] **area**

◆ calculate_perimeter()

float calculate_perimeter (float **radius**)

Calculates the Perimeter of the circle. Formula: $\text{Perimeter} = 2 * \text{PI} * r$

Parameters

[in] **radius**

[out] **perimeter**

FunctionsandProgramStructure

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 printf 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
Original string: Hello, world!
Reversed string: !dlrow ,olleH
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise8/out$
```

Q9)

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.

```
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): a
Enter variable name: xyz
Enter variable value: 10
Variable 'xyz' assigned with value 10
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): r
Enter variable name to retrieve: xyz
Value of variable 'xyz': 10
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): d
Stack contents:
Variable name: xyz Value: 10
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): a
Enter variable name: kush
Enter variable value: 20
Variable 'kush' assigned with value 20
```

```
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): a
Enter variable name: jig
Enter variable value: 30
Variable 'jig' assigned with value 30
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): d
Stack contents:
Variable name: jig Value: 30
Variable name: kush Value: 20
Variable name: xyz Value: 10
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): r
Enter variable name to retrieve: jigs
Variable 'jigs' not found
```

```
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): r
Enter variable name to retrieve: jhusi
Variable 'jhusi' not found
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): r
Enter variable name to retrieve: Xyz
Variable 'Xyz' not found
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): r
Enter variable name to retrieve: kush
Value of variable 'kush': 20
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): p
Popped variable: jig    Value: 30
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): d
Stack contents:
Variable name: kush Value: 20
Variable name: xyz Value: 10
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): p
Popped variable: kush    Value: 20
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): d
Stack contents:
Variable name: xyz Value: 10
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): p
Popped variable: xyz    Value: 10
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): d
No elements present
Enter command (a for assign, r for retrieve, d for display stack, p for pop, q for quit): q
Exiting...
```


BasicPointersandArray

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).

```
enter string:abcdabc
```

```
enter pattern to be searched in the string:abc
```

```
Found: 4
```

```
enter number you want to convert to string:145  
145
```

```
enter string to reverse:hello
```

```
olleh
```

```
12 13 14 15 16 ?  
topmost=16.000000  
secondmost=15.000000
```

Q6)

Write the function strend(s,t), which returns 1 if the string t occurs at the end of the string s, and zero otherwise.

```
enter first string=hello world  
enter second string to check whether it exists at the end=world  
string occurs at the end of the string.
```

```
enter first string=hello world  
enter second string to check whether it exists at the end=world!  
string doesn't occur at the end
```

StructureandUnion

Exercise 1

Our version of getword does not properly handle underscore, 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 */
```

```
"nicetomeetyou"
Word: "nicetomeetyou"
```

```
"hello"_nice
Word: "hello"
Word: _nice
```

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 `insert`.

```
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 6:

Write a program that reads a C program and prints in alphabetical order each group of variable names that are identical in the first 6 characters.

● **vanshit@66JC9F2-Desl**

```
appleVariable  
bananaVariable  
carrotVariable  
dogVariable  
elephantVariable  
frogVariable  
goatVariable  
horseVariable  
iceVariable  
jumpVariable
```

Input and Output

Q1)

Write a program that converts upper case to lower or lower case to upper, depending on the name it is invoked with.

```
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 string with more \011 than 20 characters.\012
vrk@vrk-VirtualBox:~/testingpurpose$
```

Q3)

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

```
Enter a string: hi
hi
Enter an integer: 20
20
Enter an integer in octal: 16
20
Enter a string: khush
khush
Enter a floating-point number: 3.14
3.140000
```

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$
```

Q9)

Rewrite the postfix calculator of Chapter 4 to use `scanf` and/or `sscanf` to do the input and number conversion.

```
13 14 15 16 -19 + ?
topmost=-3.000000
secondmost=15.000000
18 15 14 11 10 9 - ?
topmost=-1.000000
secondmost=11.000000
18 15 22 27 26 50 + ?
topmost=76.000000
secondmost=27.000000
20 25 30 35 40 -45 -10 - ?
topmost=35.000000
secondmost=40.000000
```


Staticanddynamiclibrary

Static library:

```
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise1$ gcc add.c div.c mul.c sub.c -c
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise1$ 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/exercise1$ ar -rcs libsample.a ./*.o
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise1$ ls
add.c adddemo.c 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 adddemo.c -c
● 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 adddemo.o -L. -lsample -o main
● 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
10 + 20 = 30
10 - 20 = -10
10 * 20 = 200
○ 10 / 20 = 0vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise1$
```

Dynamic library:

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

```
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ gcc -shared -fPIC -o libsample.so ./*.o
● 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 main mul.c mul.o sub.c sub.o
```

```
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ gcc -c adddemo.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/staticanddynamiclibrary/exercise2/libsample.so
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$ ./main
10 + 20 = 30
10 - 20 = -10
10 * 20 = 200
○ vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/staticanddynamiclibrary/exercise2$
```

Compilation stages

```
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp$ cd Compilationstages/
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/Compilationstages$ gcc -E index.c
# 1 "index.c"
# 1 "<built-in>"
# 1 "<command-line>"
```

```
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/Compilationstages$ gcc -E index.c > index.i
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/Compilationstages$ ls
index.c  index.i
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/Compilationstages$ gcc -S index.i
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/Compilationstages$ ls
index.c  index.i  index.s
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/Compilationstages$ cat index.s
.file "index.c"
.text
.section .rodata
.LC0:
.string "Hello, World!"
.text
.globl main
.type main, @function
main:
.LFB0:
.cfi startproc
```

```
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/Compilationstages$ gcc -c index.s
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/Compilationstages$ ls
index.c  index.i  index.o  index.s
```

```
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/Compilationstages$ gcc -o index.o
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/Compilationstages$ ls
a.out  index.c  index.i  index.o  index.s
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/Compilationstages$ ./a.out
Hello, World!
● vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/Compilationstages$
```

Function Pointers

```
enter first number:10
enter second number:20

sum=30
```

```
first value=40
second value=50
sum of two values=90
```