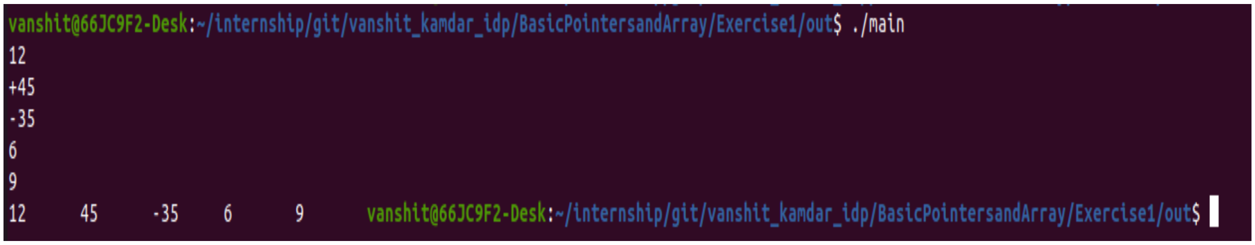
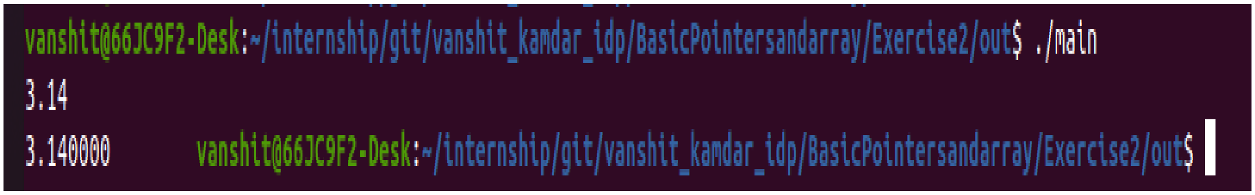
Exercise 7

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.



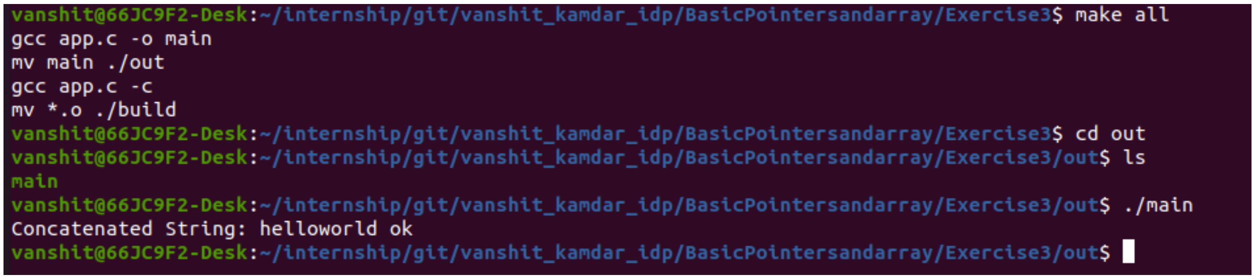
```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandArray/Exercise1/out$ ./main
12
+45
-35
6
9
12      45      -35     6       9       vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandArray/Exercise1/out$
```

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



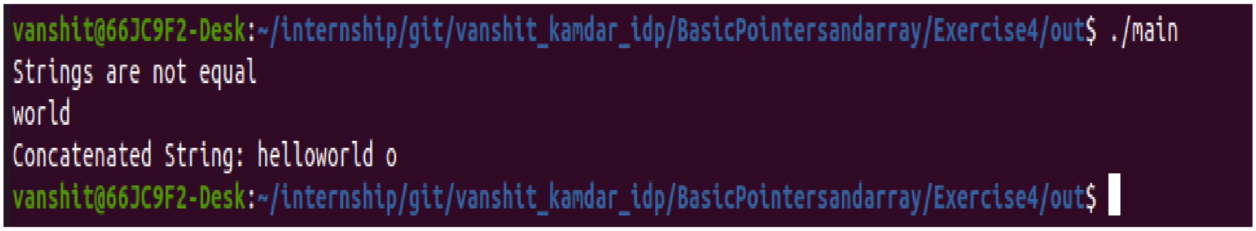
```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandarray/Exercise2/out$ ./main
3.14
3.140000        vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandarray/Exercise2/out$
```

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. ?



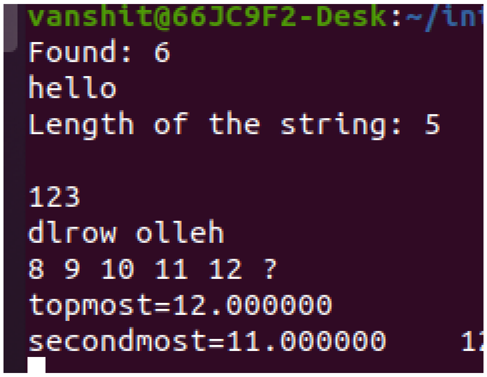
```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandarray/Exercise3$ make all
gcc app.c -o main
mv main ./out
gcc app.c -c
mv *.o ./build
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandarray/Exercise3$ cd out
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/BasicPointersandarray/Exercise3/out$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

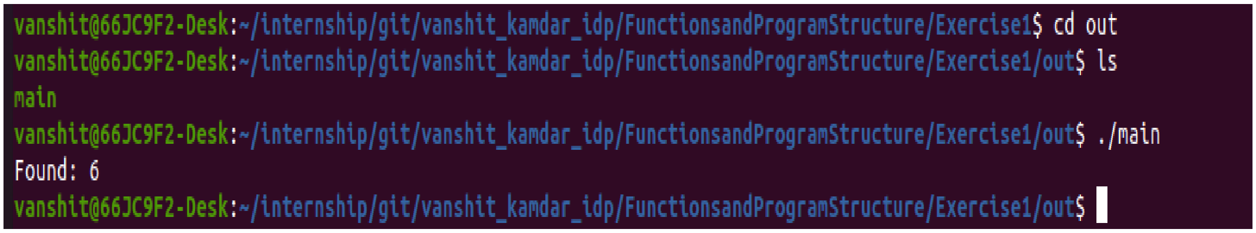
Q5)
Rewrite appropriate programs from earlier chapters and exercises with pointers instead of array indexing. Good possibilities include getline (Chapters 1 and 4), atoi, itoa, and their variants (Chapters 2, 3, and 4), reverse (Chapter 3), and strindex and getop (Chapter 4).



```
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
```

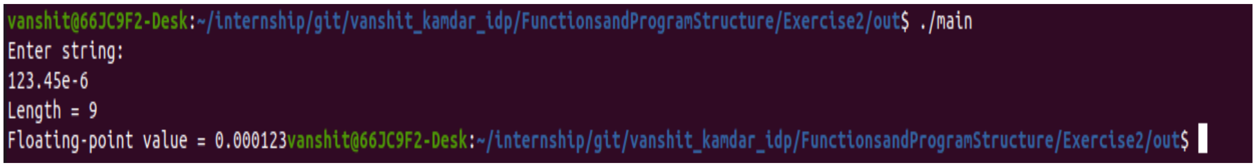
Exercise 6

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.



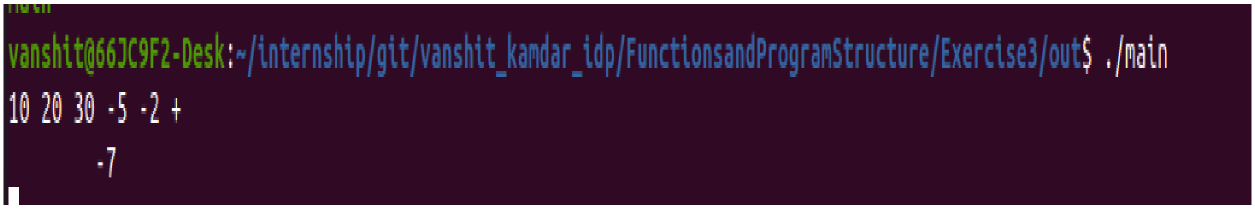
```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise1$ cd out
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise1/out$ ls
main
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise1/out$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

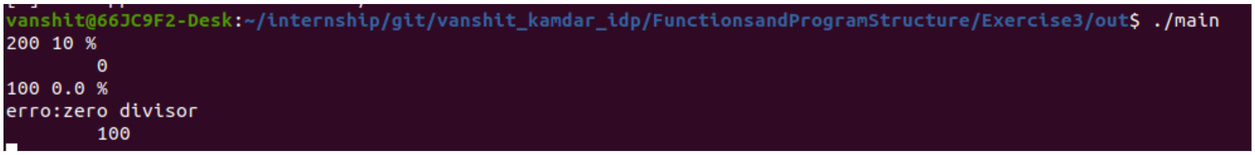


```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise2/out$ ./main
Enter string:
123.45e-6
Length = 9
Floating-point value = 0.000123vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise2/out$
```

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

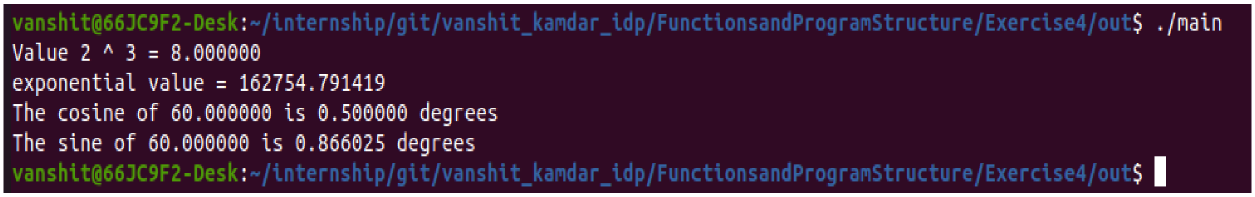


```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise3/out$ ./main
10 20 30 -5 -2 +
        -7
```



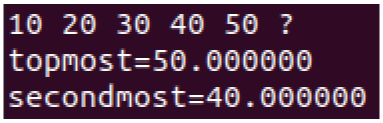
```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise3/out$ ./main
200 10 %
        0
100 0.0 %
erro:zero divisor
        100
```

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

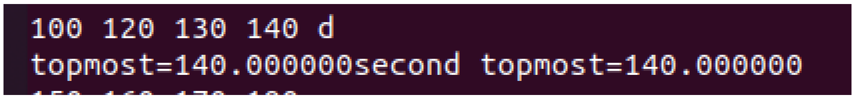


```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise4/out$ ./main
Value 2 ^ 3 = 8.000000
exponential value = 162754.791419
The cosine of 60.000000 is 0.500000 degrees
The sine of 60.000000 is 0.866025 degrees
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise4/out$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

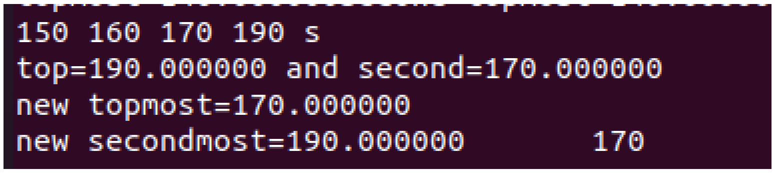
Q5)
Add the commands to print the top elements of the stack without popping, to duplicate it, and to swap the top two elements. Add a command to clear the stack.



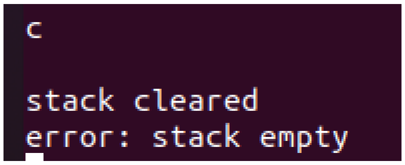
```
10 20 30 40 50 ?
topmost=50.000000
secondmost=40.000000
```



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

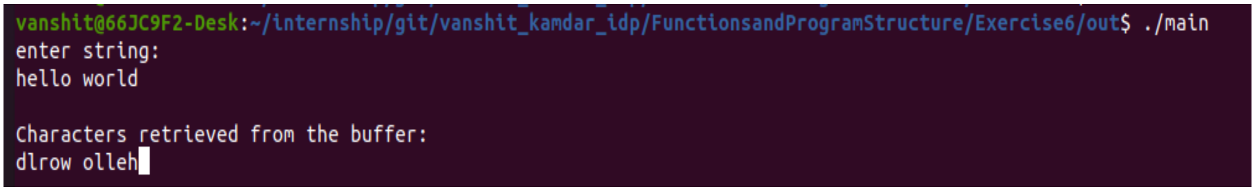


```
150 160 170 190 s
top=190.000000 and second=170.000000
new topmost=170.000000
new secondmost=190.000000        170
```



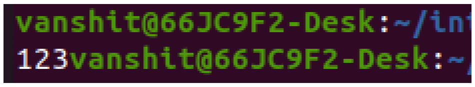
```
 c

 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?



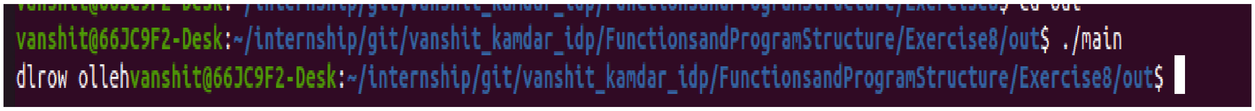
```
vanshit@66JC9F2-Desk:~/internship/git/vanshit_kamdar_idp/FunctionsandProgramStructure/Exercise6/out$ ./main
enter string:
hello world

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:~/int
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$
```