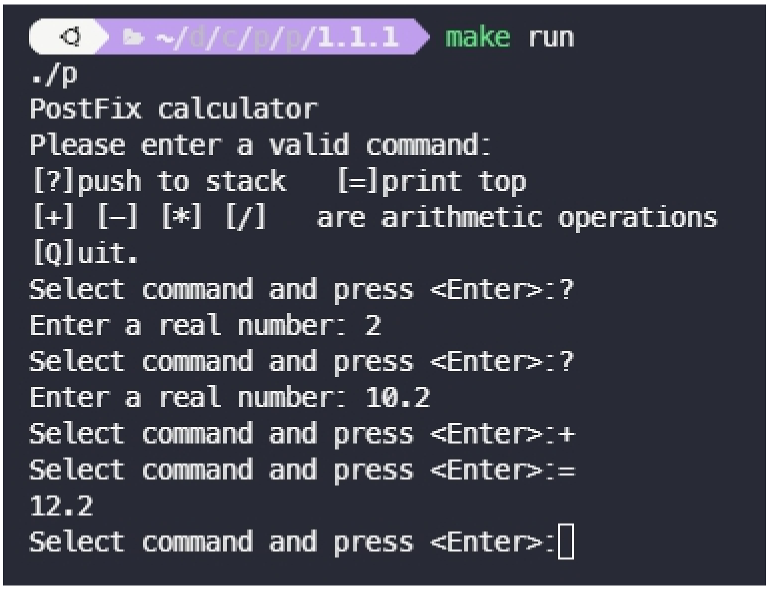
Solution and testing

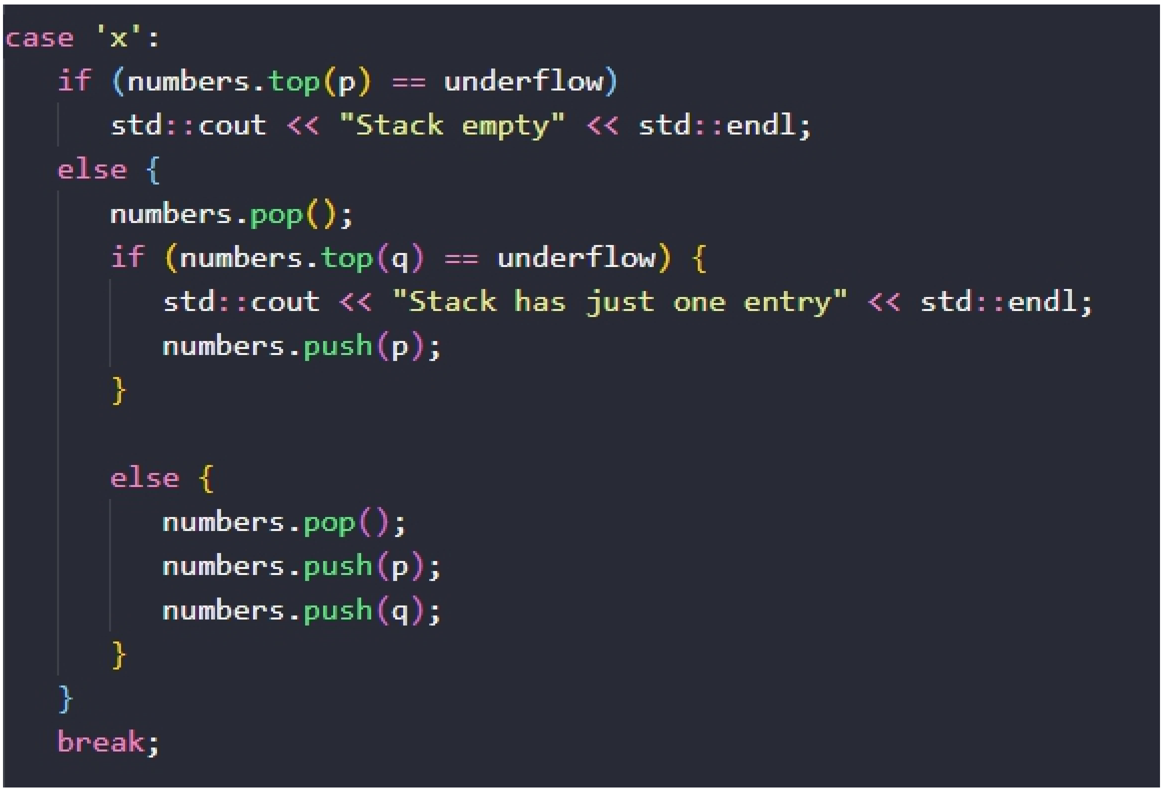
1.1.1



```
~/d/c/p/p/1.1.1 make run
./p
PostFix calculator
Please enter a valid command:
[?]push to stack   [=]print top
[+] [-] [*] [/]   are arithmetic operations
[Q]uit.
Select command and press <Enter>:?
Enter a real number: 2
Select command and press <Enter>:?
Enter a real number: 10.2
Select command and press <Enter>:+
Select command and press <Enter>:=
12.2
Select command and press <Enter>:
```

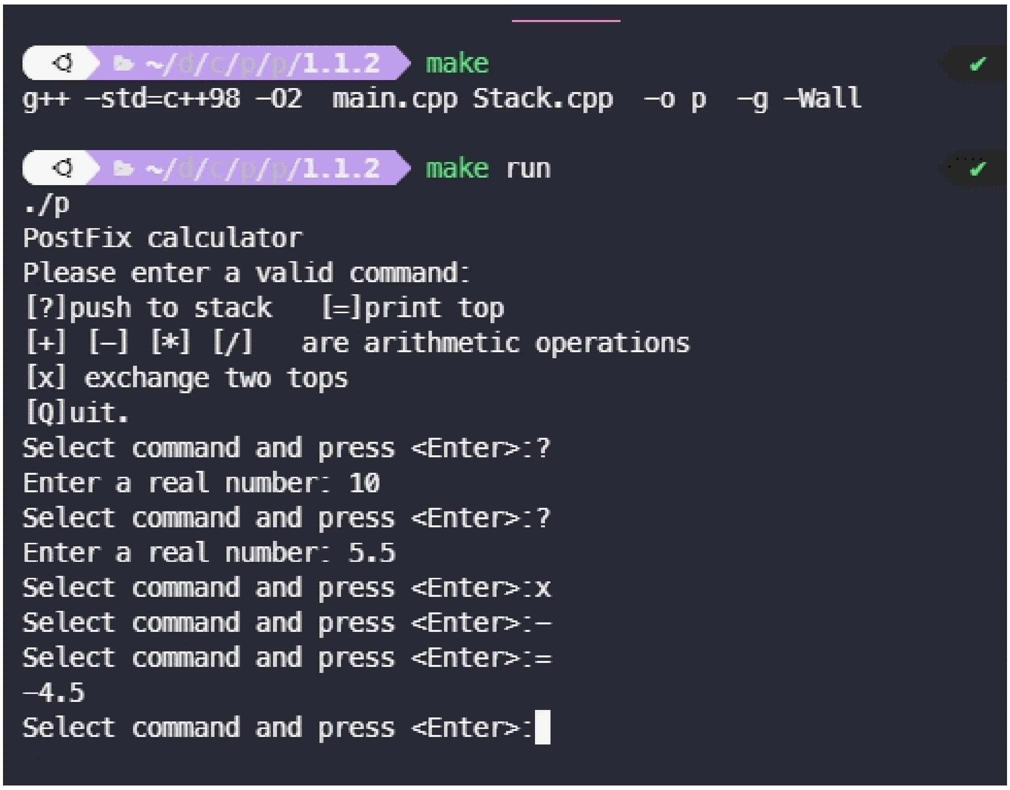
1.1.2

- Swap p and q position by taking them out of stack and change push orders.



```cpp
case 'x':
    if (numbers.top(p) == underflow)
        std::cout << "Stack empty" << std::endl;
    else {
        numbers.pop();
        if (numbers.top(q) == underflow) {
            std::cout << "Stack has just one entry" << std::endl;
            numbers.push(p);
        }

        else {
            numbers.pop();
            numbers.push(p);
            numbers.push(q);
        }
    }
    break;
```



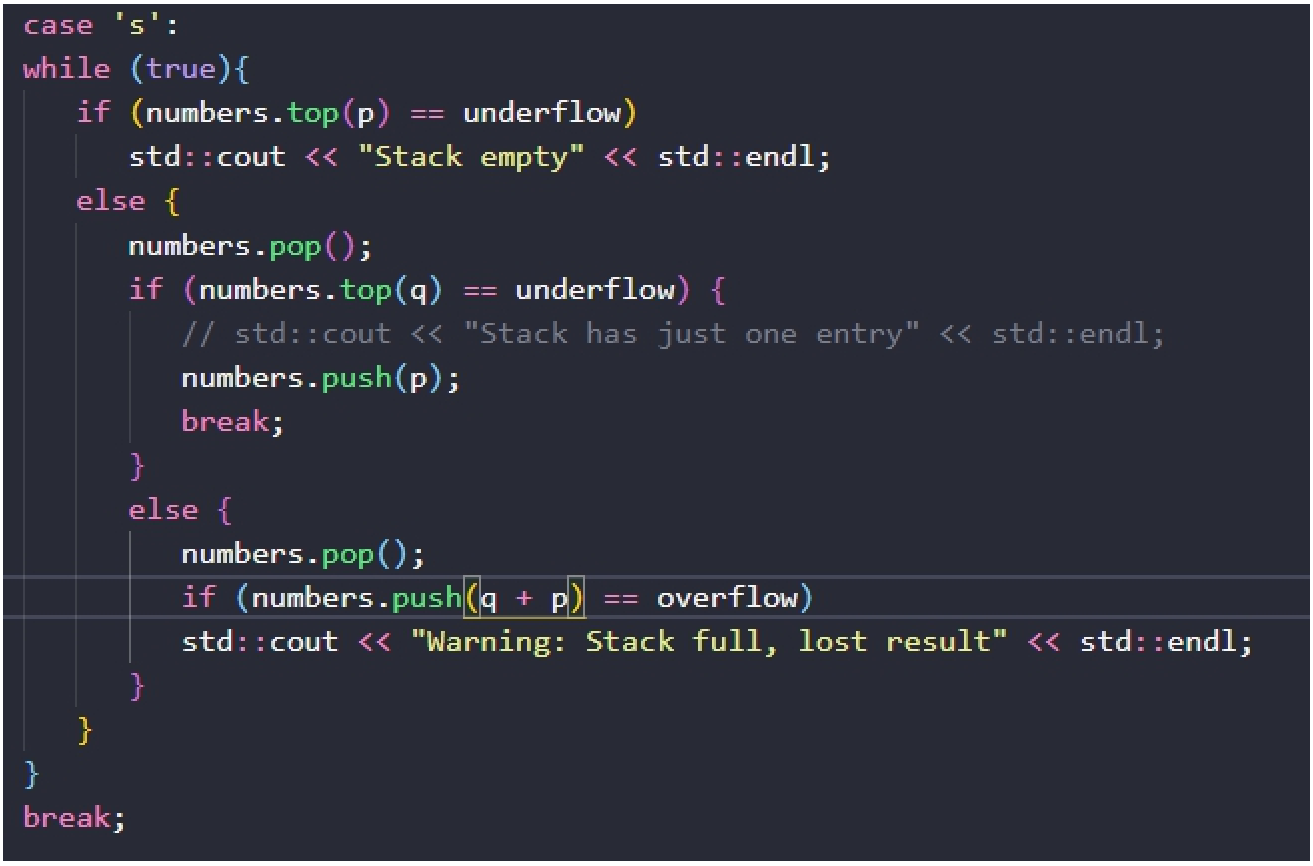
```
~/d/c/p/p/1.1.2 make
g++ -std=c++98 -O2  main.cpp Stack.cpp  -o p  -g -Wall

~/d/c/p/p/1.1.2 make run
./p
PostFix calculator
Please enter a valid command:
[?]push to stack   [=]print top
[+] [-] [*] [/]   are arithmetic operations
[x] exchange two tops
[Q]uit.
Select command and press <Enter>:?
Enter a real number: 10
Select command and press <Enter>:?
Enter a real number: 5.5
Select command and press <Enter>:x
Select command and press <Enter>:-
Select command and press <Enter>:=
-4.5
Select command and press <Enter>:
```

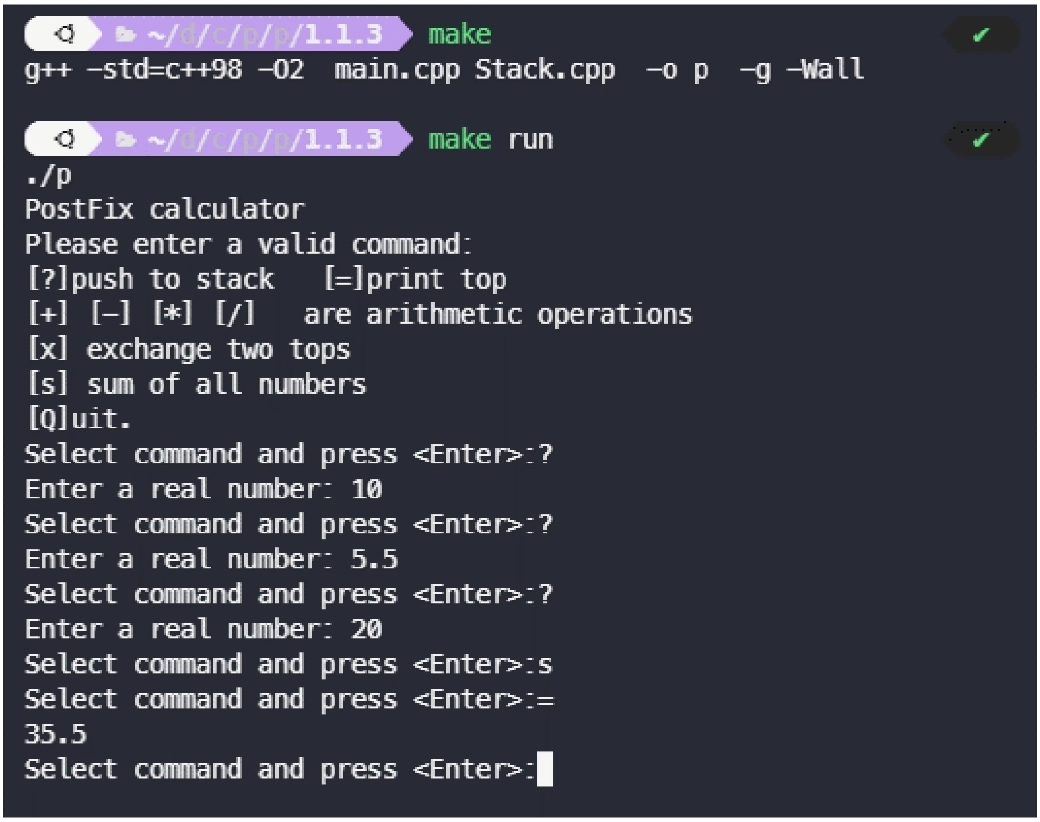
1.1.3

- Add a loop into already source code for plus function.

- Add a break point when Stack have only one entry.



```cpp
case 's':
while (true){
    if (numbers.top(p) == underflow)
        std::cout << "Stack empty" << std::endl;
    else {
        numbers.pop();
        if (numbers.top(q) == underflow) {
            // std::cout << "Stack has just one entry" << std::endl;
            numbers.push(p);
            break;
        }
        else {
            numbers.pop();
            if (numbers.push(q + p) == overflow)
            std::cout << "Warning: Stack full, lost result" << std::endl;
        }
    }
}
break;
```



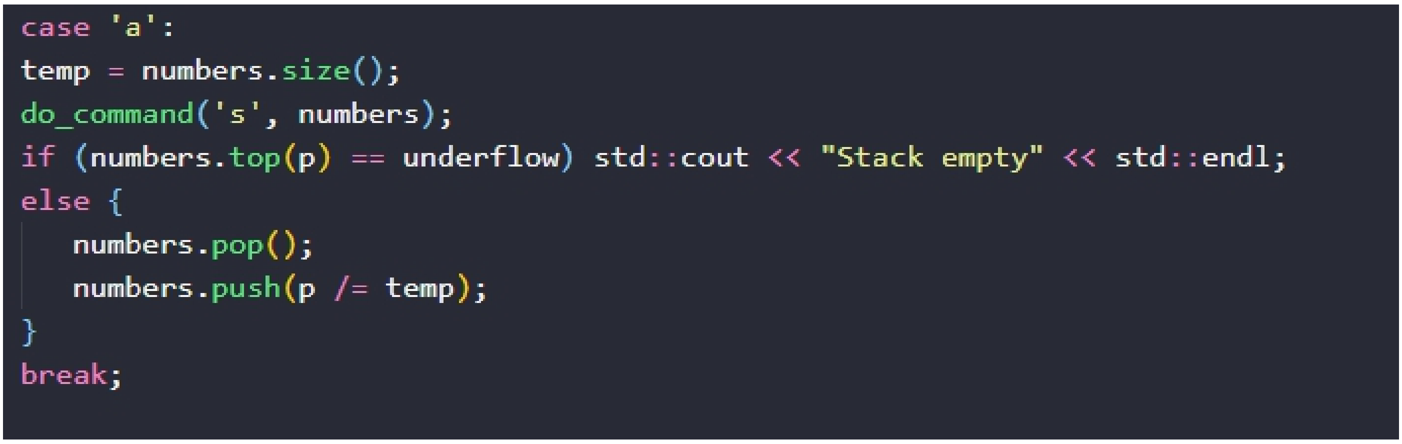
```
~/d/c/p/p/1.1.3  make
g++ -std=c++98 -O2  main.cpp Stack.cpp  -o p  -g -Wall

~/d/c/p/p/1.1.3  make run
./p
PostFix calculator
Please enter a valid command:
[?]push to stack   [=]print top
[+] [-] [*] [/]   are arithmetic operations
[x] exchange two tops
[s] sum of all numbers
[Q]uit.
Select command and press <Enter>:?
Enter a real number: 10
Select command and press <Enter>:?
Enter a real number: 5.5
Select command and press <Enter>:?
Enter a real number: 20
Select command and press <Enter>:s
Select command and press <Enter>:=
35.5
Select command and press <Enter>:
```

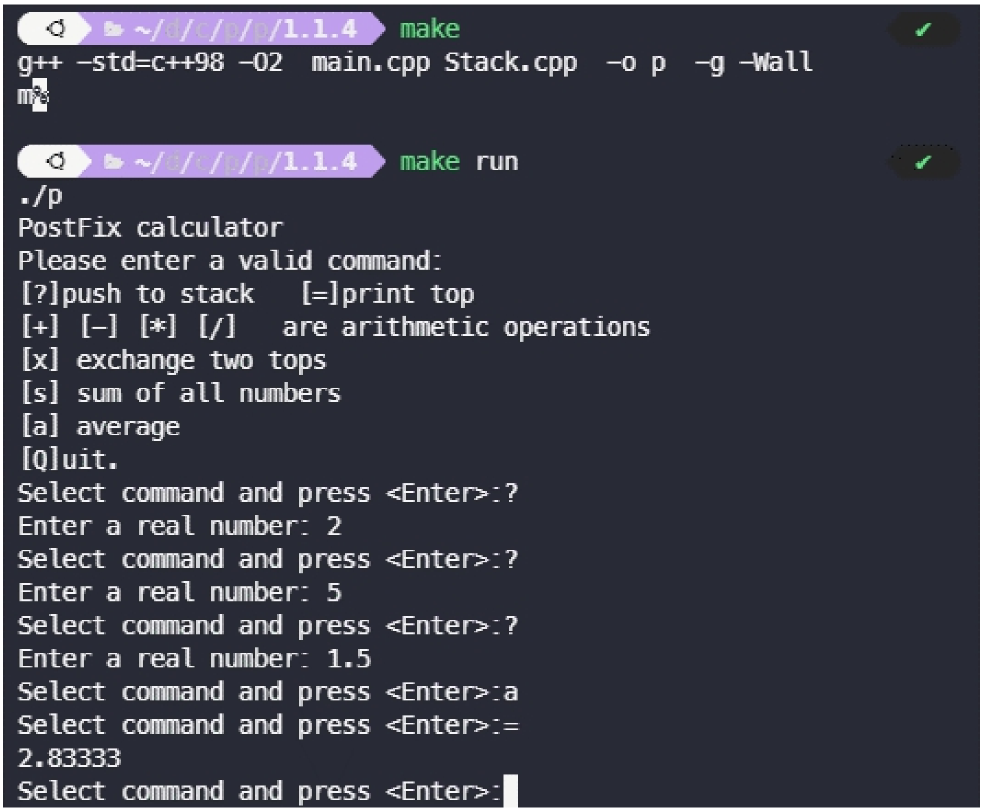
1.1.4

- Add one function .size() in Stack class to return size.

- Run sum command first and use that sum to find average by devising Sum and temp (size of previous command stack)



```
case 'a':
temp = numbers.size();
do_command('s', numbers);
if (numbers.top(p) == underflow) std::cout << "Stack empty" << std::endl;
else {
    numbers.pop();
    numbers.push(p /= temp);
}
break;
```



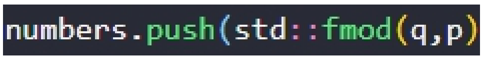
```
~/d/c/p/p/1.1.4  make                                          ✔
g++ -std=c++98 -O2  main.cpp Stack.cpp  -o p  -g -Wall
m%

~/d/c/p/p/1.1.4  make run                                      ✔
./p
PostFix calculator
Please enter a valid command:
[?]push to stack   [=]print top
[+] [-] [*] [/]   are arithmetic operations
[x] exchange two tops
[s] sum of all numbers
[a] average
[Q]uit.
Select command and press <Enter>:?
Enter a real number: 2
Select command and press <Enter>:?
Enter a real number: 5
Select command and press <Enter>:?
Enter a real number: 1.5
Select command and press <Enter>:a
Select command and press <Enter>:=
2.83333
Select command and press <Enter>:
```

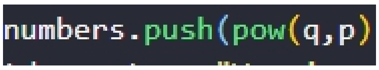
1.1.5

- For additional operators, I just added math.h library and add syntaxes fit with the requirements.

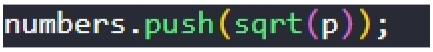
- Using argc and argv to get input from command and using condition for ignore this function if the command have "-p"



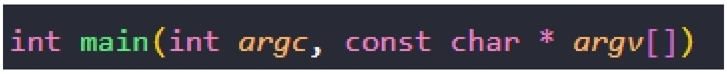
```
numbers.push(std::fmod(q,p)
```



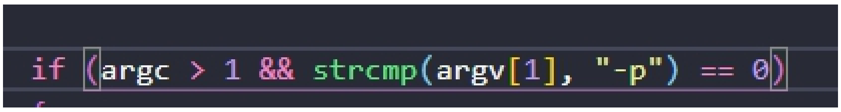
```
numbers.push(pow(q,p)
```



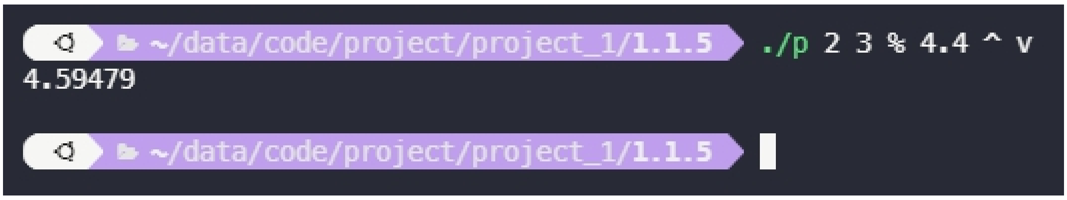
```
numbers.push(sqrt(p));
```



```
int main(int argc, const char * argv[])
```

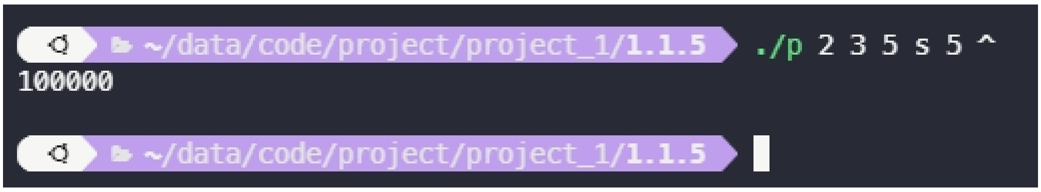


```
  if (argc > 1 && strcmp(argv[1], "-p") == 0)
```



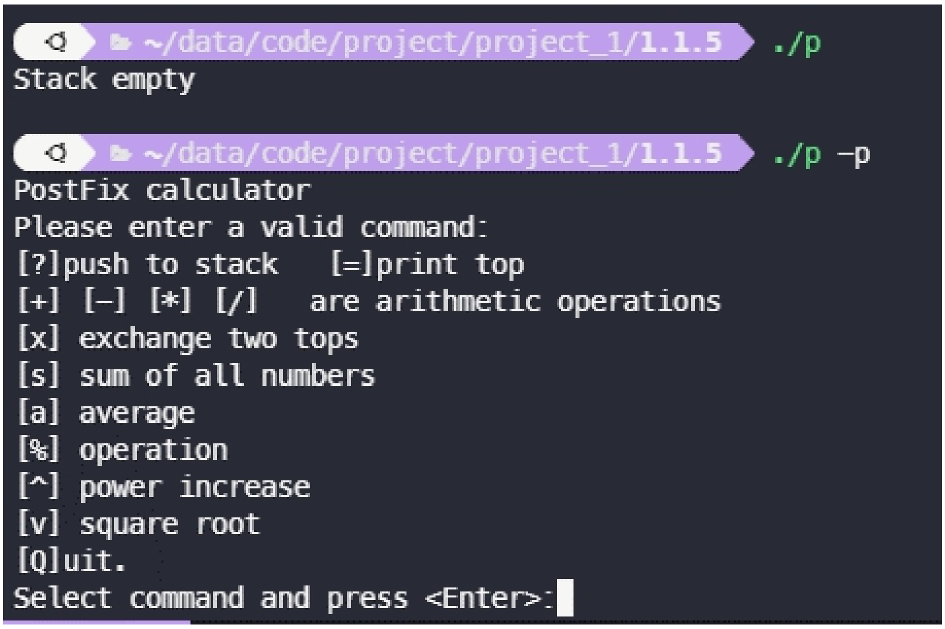
```
~/data/code/project/project_1/1.1.5  ./p 2 3 % 4.4 ^ v
4.59479

~/data/code/project/project_1/1.1.5
```



```
~/data/code/project/project_1/1.1.5  ./p 2 3 5 s 5 ^
100000

~/data/code/project/project_1/1.1.5
```



```
~/data/code/project/project_1/1.1.5  ./p
Stack empty

~/data/code/project/project_1/1.1.5  ./p -p
PostFix calculator
Please enter a valid command:
[?]push to stack   [=]print top
[+] [-] [*] [/]   are arithmetic operations
[x] exchange two tops
[s] sum of all numbers
[a] average
[%] operation
[^] power increase
[v] square root
[Q]uit.
Select command and press <Enter>:
```