

O algoritmo desenvolvido para a calculadora consiste em um bloco de código para inicializar a pilha e para a leitura do arquivo de entrada, adicionando cada elemento dele seguido de um espaço em branco em uma *String*.

```
Pilha pilha = new Pilha()

Path path1 = Paths.get("exemplo2.txt")
String s = ""
try(Scanner sc=new Scanner(Files.newBufferedReader(path1,Charset.defaultCharset()))
{
    while (sc.hasNext()) {
        s += sc.next()
        s += " "
    }
    sc.close()
} catch (IOException x) {
    System.err.format("Erro de E/S: %s%n", x)
}
```

Em seguida, a *String* formada com os dados de entradas é separada em um array de *Strings*, no qual se opera um *for-each* com um *switch/case* para decidir o que acontece com cada elemento e por fim, a pilha é impressa no terminal.

```
String[] splitted = s.split(" ")
for(String op : splitted){
    switch(op){
        case "+": {
            double last = pilha.pop()
            double secondToLast = pilha.pop()
            pilha.push((last + secondToLast))
            break
        }
        case "-": {
            double last = pilha.pop()
            double secondToLast = pilha.pop()
            pilha.push((last - secondToLast))
            break
        }
        case "*": {
            double last = pilha.pop()
            double secondToLast = pilha.pop()
            pilha.push((last * secondToLast))
            break
        }
        case "/": {
            double last = pilha.pop()
            double secondToLast = pilha.pop()
            pilha.push((last / secondToLast))
            break
        }
    }
}
```

```

case "swap": {
    double last = pilha.pop()
    double secondToLast = pilha.pop()
    pilha.push(last)
    pilha.push(secondToLast)
    break
}
case "chs": {
    double element = pilha.pop()
    pilha.push((-1 * element))
    break
}
case "pop": {
    pilha.pop()
    break
}
case "dup": {
    pilha.push(pilha.top())
    break
}
case "sqrt": {
    double element = Math.sqrt(pilha.top())
    pilha.push(element)
    break
}
default: {
    double element = Double.parseDouble(op)
    pilha.push(element)
    break
}
}
}
pilha.imprime()

```

Por apresentar somente um *for-each* para andar sobre *array* de *Strings*, o algoritmo tem notação $O(n)$, pois ele executa operações conforme o número de entradas dadas no arquivo.

RESULTADOS:

Exemplo 1:

Entrada: 123 234 456 / + 357 3 * swap -

Resultado: -946.051282

Exemplo 2:

Entrada: 69 785 201 - + 108 2 / swap – dup 792 chs + 432 567 612 709 813 + -

* + dup chs 8 swap - - * - 3547 93 54 12 45 dup 98 + chs dup 87 / pop

+ - chs * - pop + 899 + sqrt

Resultado: 1349909371.166667, 36741.112819

Exemplo 3:

Entrada: 11 22 33 44 55 66 77 88 + - + - 2 swap / dup chs 99 + 3 * 10 swap 20
30 40 50 51 52 - sqrt + + 60 - 70 + 2 * 80 80 - pop - - * pop + + chs
swap +

Resultado: 11.000000, 22.000000, -164.000000,

Exemplo 4:

Entrada: 110 220 330 440 550 660 770 880 990 + - + - + 2 swap / dup chs 99 +
3 * 10 swap 20 30 40 50 51 52 - sqrt + + 60 - 70 + 2 * 80 - pop - - * pop
+ + chs swap + 69 785 201 - + 108 2 / swap - dup 792 chs + 432 567
612 709 813 + - * + dup chs 8 swap - - * - +

Resultado: 110.000000, 1349903980.166667