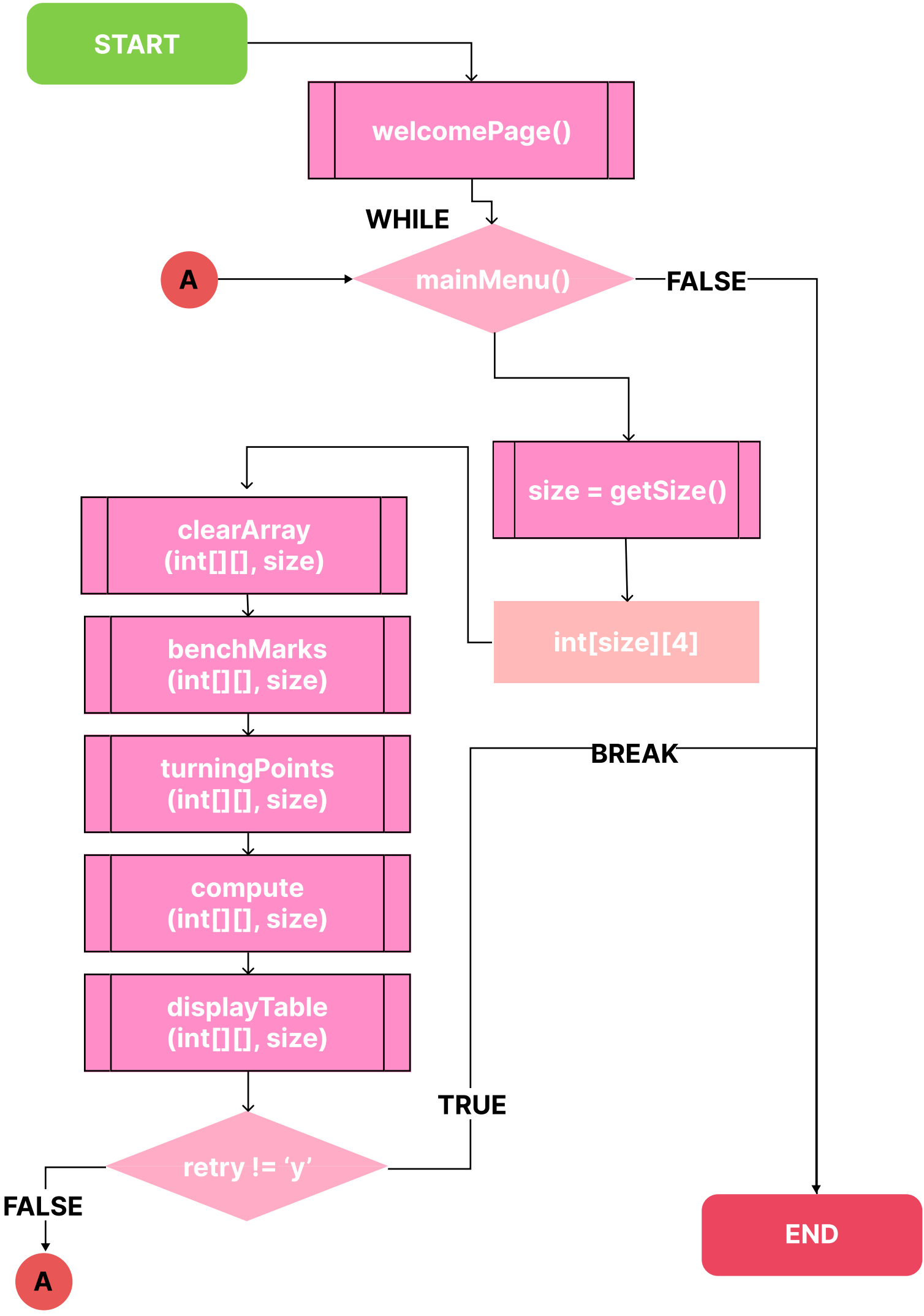
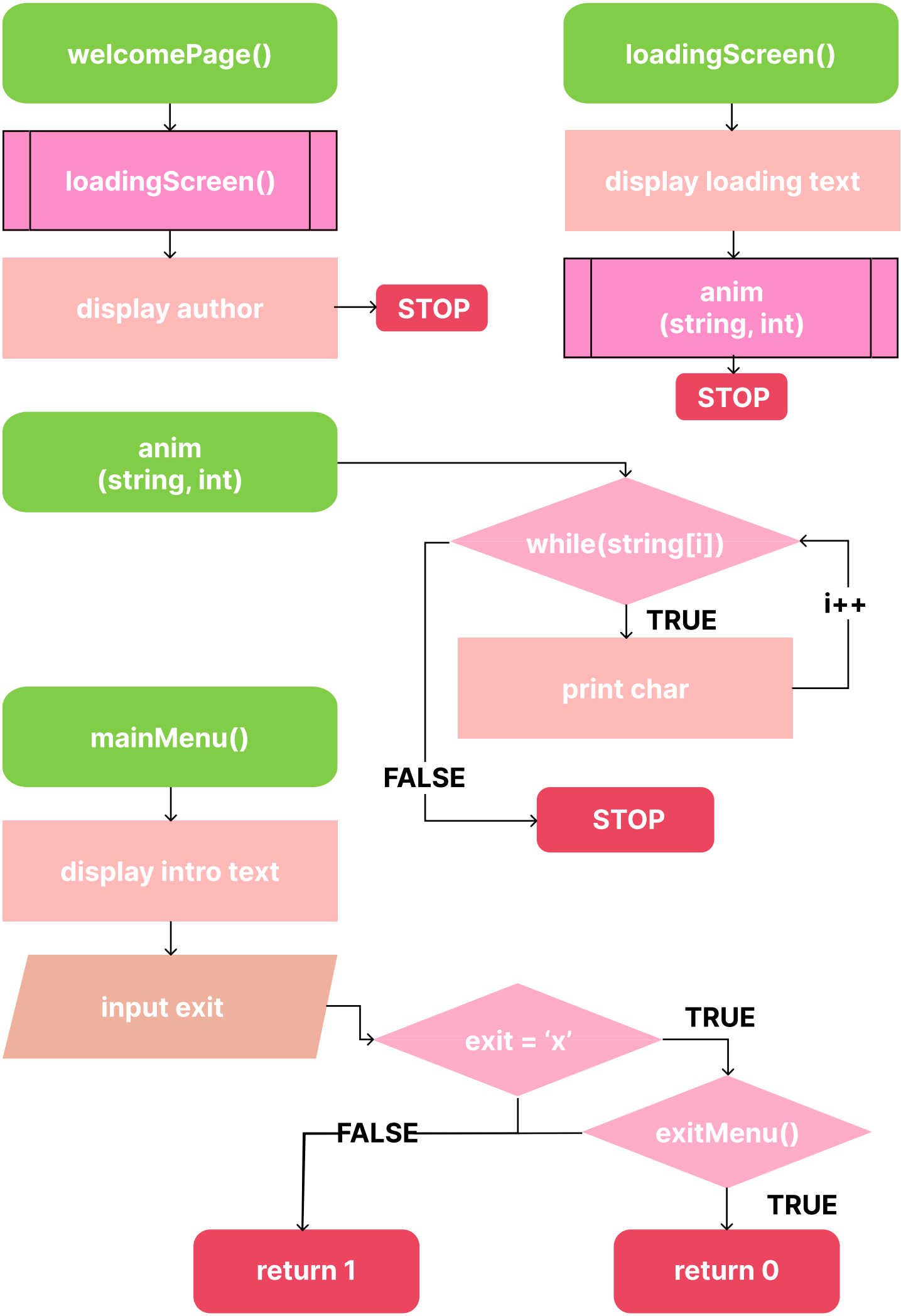
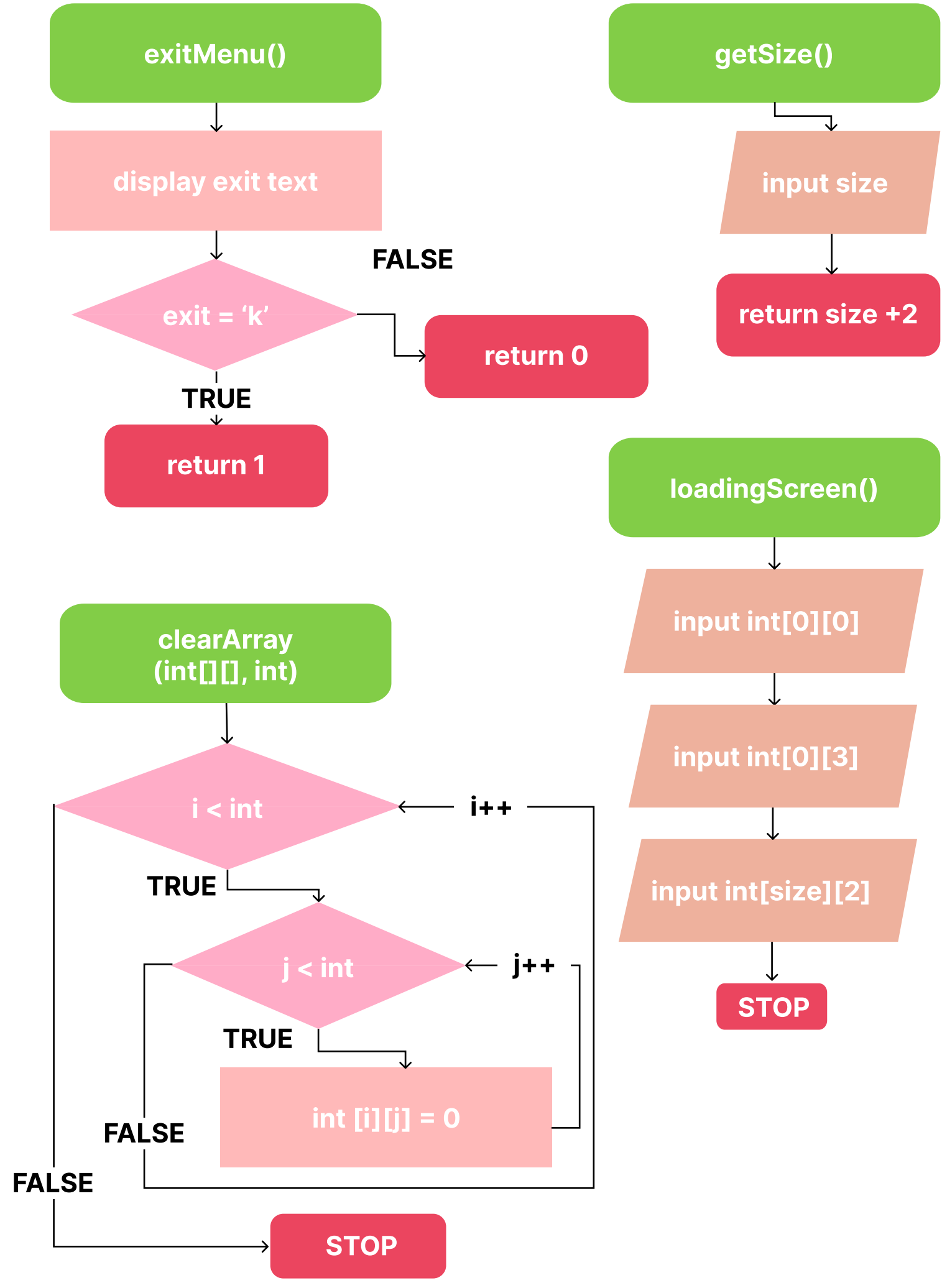
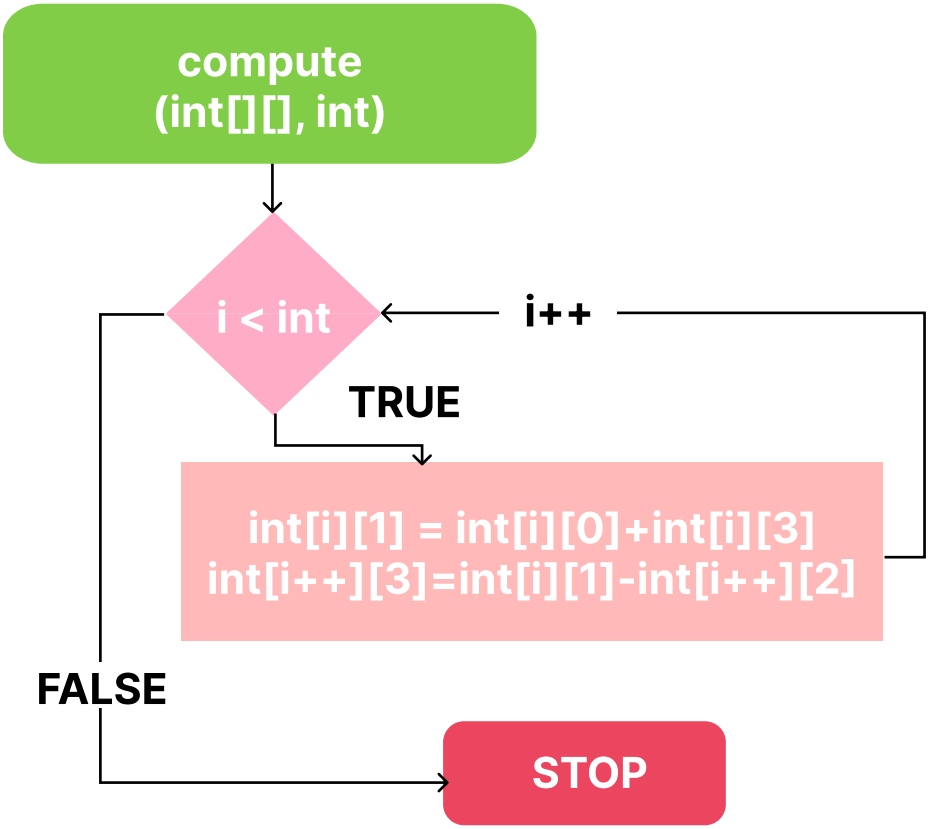
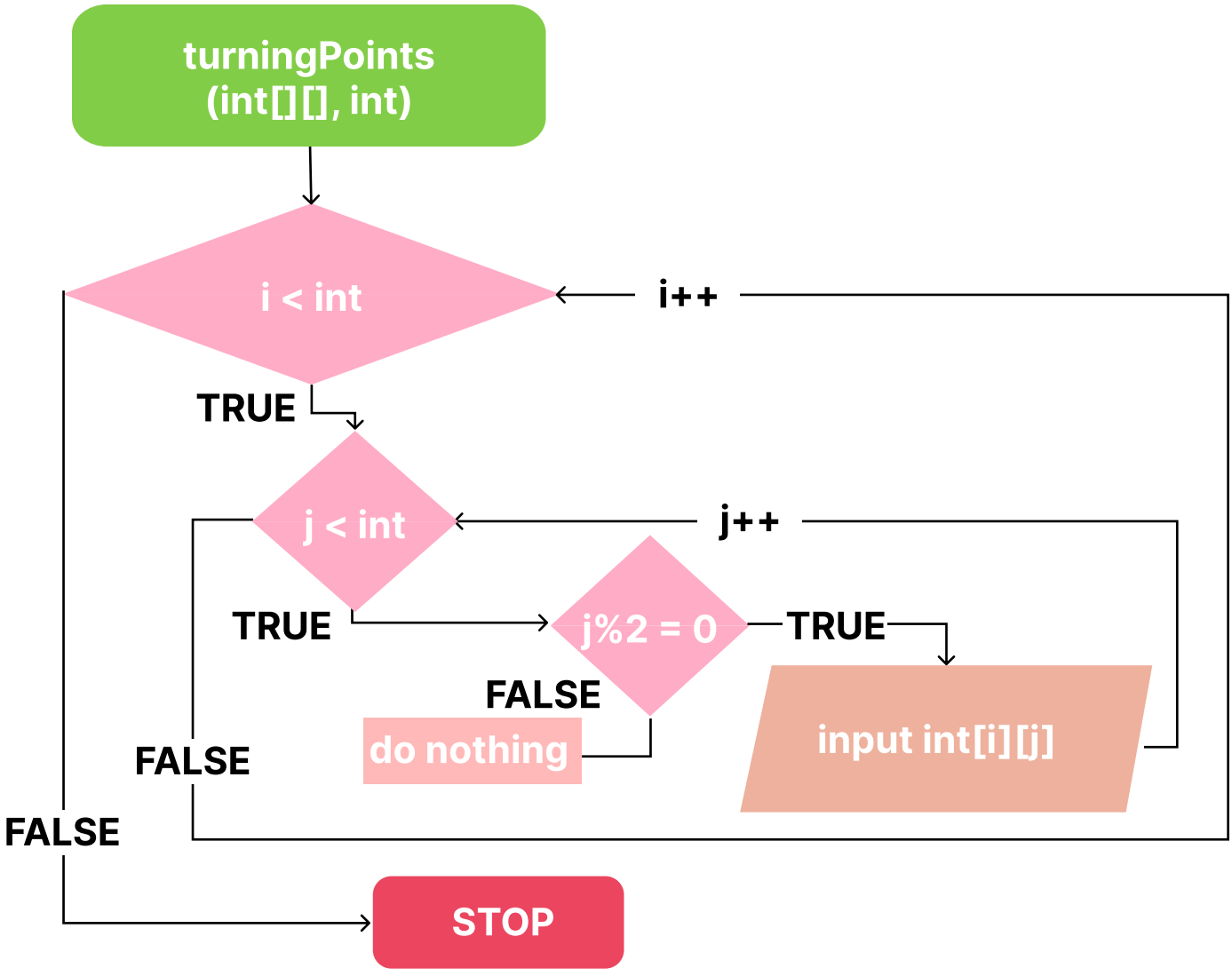
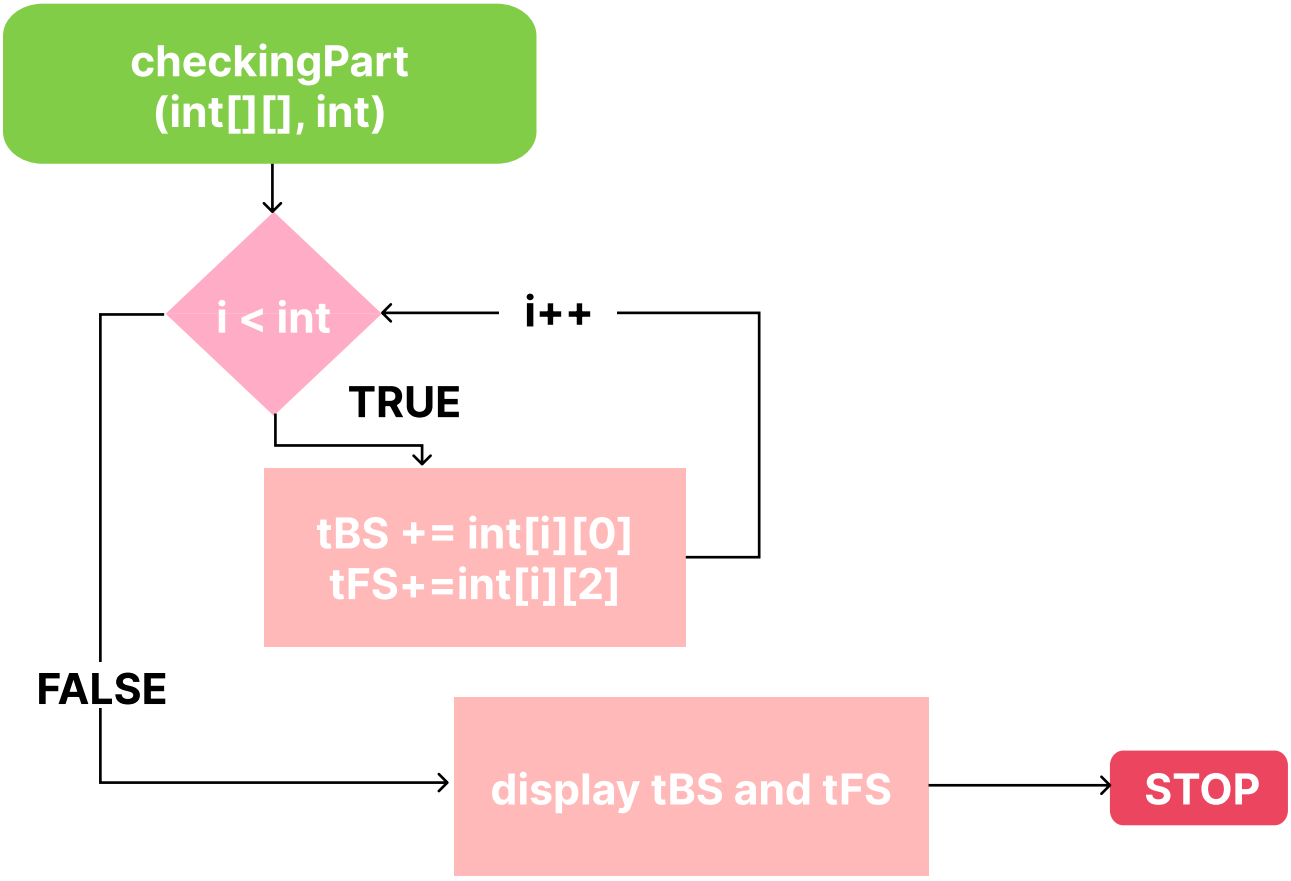

START
welcomePage()
WHILE
A
mainMenu()
FALSE
size = getSize()
int[size][4]
clearArray
(int[][], size)
benchMarks
(int[][], size)
turningPoints
(int[][], size)
compute
(int[][], size)
displayTable
(int[][], size)
BREAK
TRUE
retry != 'y'
FALSE
A
END

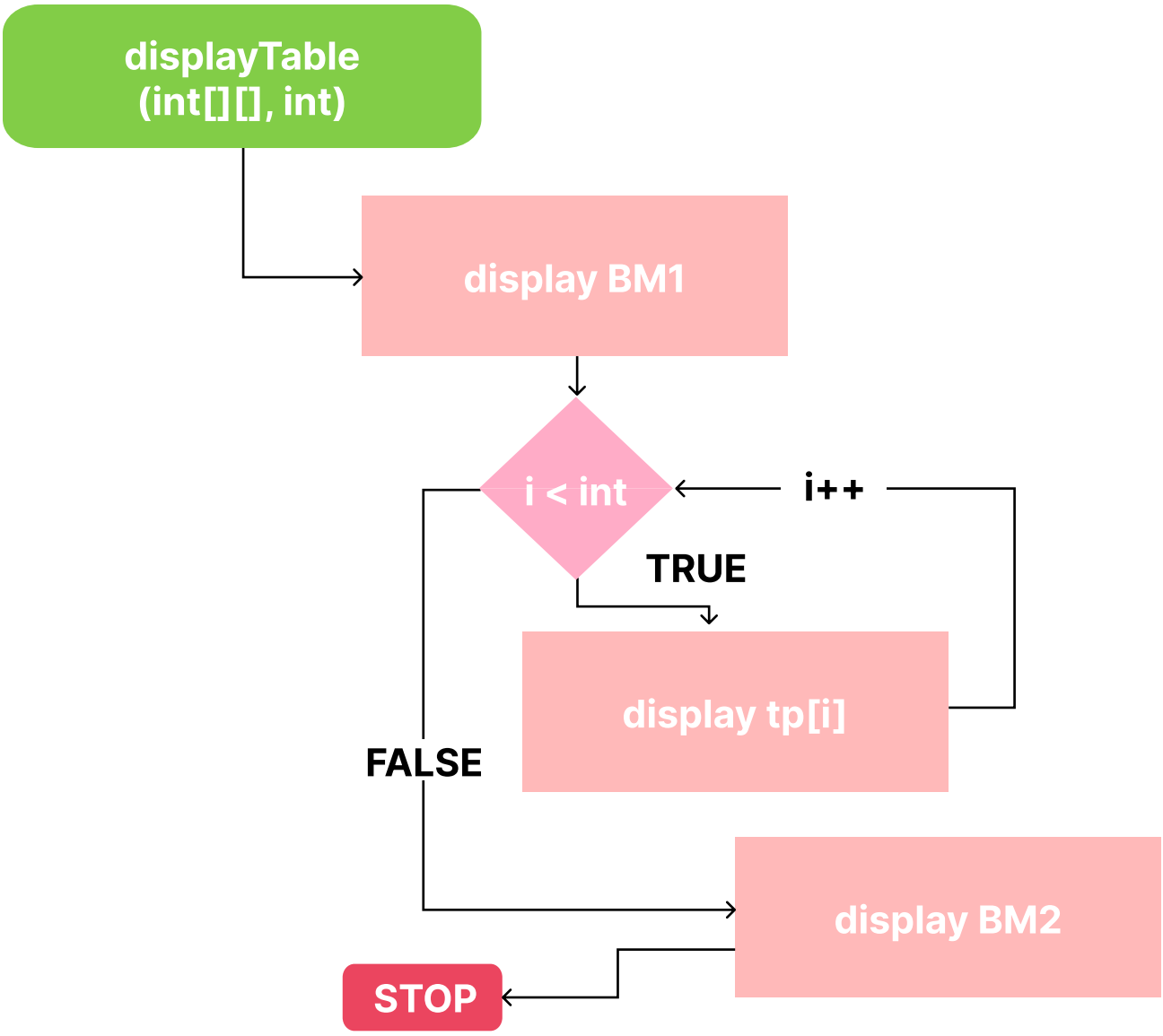

welcomePage()
loadingScreen()
display author
STOP
loadingScreen()
display loading text
anim
(string, int)
STOP
anim
(string, int)
while(string[i])
TRUE
print char
i++
FALSE
STOP
mainMenu()
display intro text
input exit
exit = 'x'
TRUE
FALSE
exitMenu()
TRUE
return 1
return 0


exitMenu()
display exit text
exit = 'k'
FALSE
return 0
TRUE
return 1
getSize()
input size
return size +2
loadingScreen()
input int[0][0]
input int[0][3]
input int[size][2]
STOP
clearArray
(int[][], int)
i < int
i++
TRUE
j < int
j++
TRUE
int [i][j] = 0
FALSE
FALSE
STOP


compute
(int[][], int)
i < int
i++
TRUE
int[i][1] = int[i][0]+int[i][3]
int[i++][3]=int[i][1]-int[i++][2]
FALSE
STOP


turningPoints
(int[][], int)
i < int
i++
TRUE
j < int
j++
TRUE
j%2 = 0
TRUE
FALSE
do nothing
input int[i][j]
FALSE
FALSE
STOP


checkingPart
(int[][], int)
i < int
i++
TRUE
tBS += int[i][0]
tFS+=int[i][2]
FALSE
display tBS and tFS
STOP


displayTable
(int[][], int)
display BM1
i < int
i++
TRUE
display tp[i]
FALSE
display BM2
STOP

Algorithm

main()

1. call function welcomePage()
2. initialize while loop.
3. make the loop call mainMenu() function, the mainMenu() function will then return 1, if true. and return 0, if false.
4. if mainMenu() returns 1,
    a. create variable size, then initialize size, to getSize() function, getSize() function returns integer
    b. create a 2d float array, initialize its rows to size, and columns to 4
    c. call function clearArray().
    d. call function benchMarks().
    e. call function turningPoints().
    f. call function compute().
    g. call function displayTable().
    h. create a condition if the user wants to retry the program, if user chose ‘y’, the loop will excecute once again. else it will break.
5. if mainMenu() returns 0, the loop will terminate immediately and end the program

getSize()

1. ask user to input a number.
2. return inputted number and increment by 2.

clearArray()

1. get parameters (size).
2. create a for loop and run it by how much size the array has.
3. assign each of array 0.

turningPoints()

1. create a for loop and excecute it base of size.
2. for each loop. check if index is divisible by 2,
3. if divisible by 2, ask for user input.
4. else, do nothing.
5. if for loop condition is false, stop the function

compute()

1. create a for loop and excecute it base of size.
2. for each loop. add 0 and 3rd index and assign it to 1st index.
3. also subtract the first index and 2nd index of next row, assign it to the 3rd index of the next row.
4. after the condition turns false, stop the function.

checkingPart()

1. create a for loop base on array size.
2. get the sum of all first columns and assign them to tBS.
3. get the sum all 3rd columns and assign it to tFS.
4. display tBS and tFS