COMP 2313 DATA STRUCTURES CODE ASSIGNMENT

|  |  |
| --- | --- |
| ***Assignment Name:*** *EXTRA CREDIT QUESTION* | ***Student Name :****Mustafa Senih Topsakal* |
| ***Assignment Date:****11/30/2020* | ***Student id :****18122014* |

# Problem

Write a recursive method to return all possible k permutations of the given String non-zeros number

Sample  input : "123" , 2

output : "1-2", "1-3", "2-3", "2-1", "3-1", "3-2"

# Challenges

* Write the code please.

# Code

class Main {

static void printAllKLength(char[] set, int k)

{

int n = set.length;

printAllKLengthRec(set, "", n, k);

}

static void printAllKLengthRec(char[] set,

String prefix,

int n, int k)

{

if (k == 0)

{

System.out.println(prefix);

return;

}

for (int i = 0; i < n; ++i)

{

String newPrefix = prefix + set[i];

printAllKLengthRec(set, newPrefix,

n, k - 1);

}

}

public static void main(String[] args)

{

System.out.println("Output");

char[] set = {'1', '2', '3'};

int k = 2;

printAllKLength(set, k);

}

}

# Pseudo Code

Initialize int array with random int numbers ‘1’, ‘2’, ‘3’

public static void main(String[] args)

{

System.out.println("Output");

char[] set = {'1', '2', '3'};

int k = 2;

printAllKLength(set, k);

}

}

# UML Diagram

(-)

# Outputs

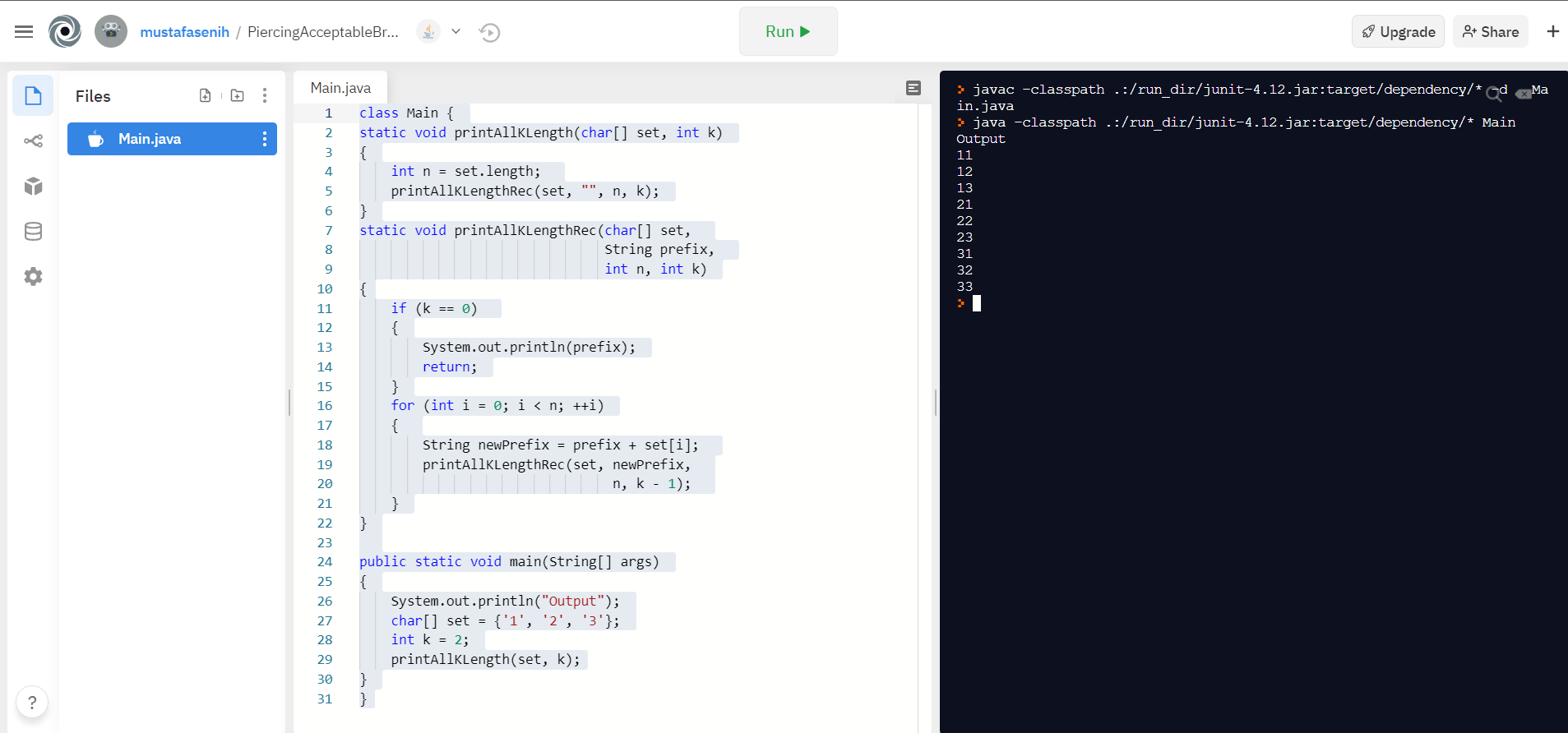


Figure Output Screen of the Odd Numbers Question

# REPL.IT LINK

https://repl.it/@mustafasenih/PiercingAcceptableBruteforceprogramming#Main.java