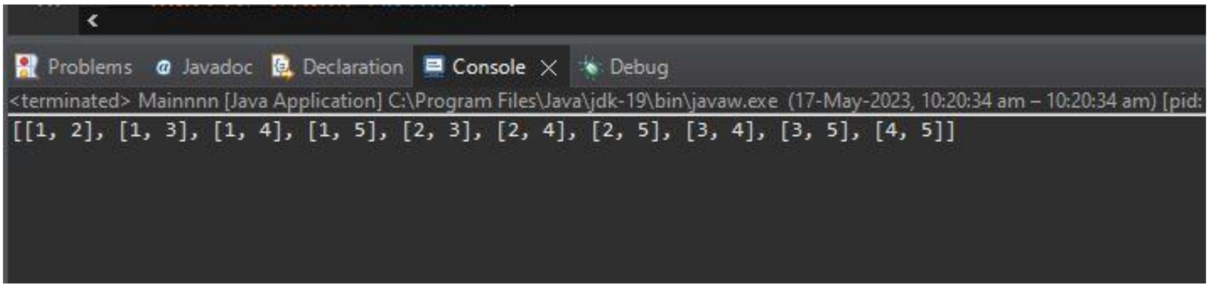
Worksheet - 18

77. Combinations

```
import java.util.ArrayList;
import java.util.List;

// Creating a class for the solution
class Solution {
// Creating a combine method with List return type to pass values
public List<List<Integer>> combine(int n, int k) {
List<List<Integer>> subsets = new ArrayList();
generateSubsets(1,n,new ArrayList(),subsets,k);
return subsets;
}
// First to get the subset values
//Created a method generateSubsets which contains the required input values
void generateSubsets(int start, int n,List<Integer> current,List<List<Integer>> subsets,int k){

// Filtering only value of required size
if(current.size()==k){
//Adding the values to the subset list
subsets.add(new ArrayList(current));


}


for(int i=start;i<=n;i++){
// Adding values to the current list
current.add(i);
// Traversing back and updating the changes in the following parameters
generateSubsets(i+1,n,current,subsets,k);
//Removing the value inside current list since it acts as temporary storage of values
current.remove(current.size()-1);
}
}
}


public class Main {


public static void main(String[] args) {
//Declaring the values for n and k
int n=5;
int k=2;
// Creating the object for class solution
Solution obj = new Solution ();
// Passing the values inside the method call parameter and Printing
System.out.println(obj.combine(n,k));
}
}
```

Output:

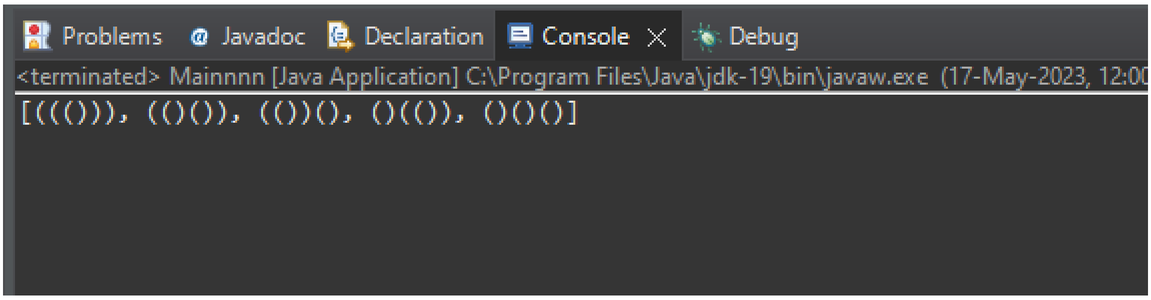


```
<terminated> Mainnnn [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe  (17-May-2023, 10:20:34 am – 10:20:34 am) [pid:
[[1, 2], [1, 3], [1, 4], [1, 5], [2, 3], [2, 4], [2, 5], [3, 4], [3, 5], [4, 5]]
```

22. Generate Parentheses

```
import java.util.ArrayList;
import java.util.List;

// Creating a class for the solution
class Solution {
// Creating a method generateParenthesis with list return type and receiving the value of n
public List<String> generateParenthesis(int n) {
//Creating a new list to store the values
List<String> result = new ArrayList<>();
//Calling generateParentheses method to get the values
generateParentheses(result, "", 0, 0, n);
return result;
}

//Created a private method generateParentheses which has the required parameters
private void generateParentheses(List<String> result, String current, int open, int close, int n) {

//If length of current list is equals to 2*n then return it
if (current.length() == 2 * n) {
//Add the current value to result set
result.add(current);
return;
}
//if open is less than n
if (open < n) {
//calling the generateParentheses with the left bracket
generateParentheses(result, current + '(', open + 1, close, n);
}
//if close is less than n
if (close < open) {

//calling the generateParentheses with the right bracket

generateParentheses(result, current + ')', open, close + 1, n);

}

}

}


public class Main {


public static void main(String[] args) {

//Declaring the values for n

int n=3;

// Creating the object for class solution

Solution obj = new Solution ();

// Passing the values inside the method call parameter

System.out.println(obj.generateParenthesis(n));

}

}
```

Output:



```
<terminated> Mainnnn [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe  (17-May-2023, 12:00
[((())), (()()), (())(), ()(()), ()()()]
```