# Multiplication

|  |  |
| --- | --- |
| **Problem** | Make a function that return the product of two integer number |
| **Rule** | 1. Use loop for solve the problem |
| **Example** | **Input:**  x = 2  y = 20  **Output:**  40 |

# Leap Year

|  |  |
| --- | --- |
| **Problem** | Make a function that return previous and next leap year from a certain year |
| **Rule** |  |
| **Example** | **Input:**  year = 2015  **Output:**  {prev: 2012, next: 2016} |

# Liner Search

|  |  |
| --- | --- |
| **Problem** | Find how many times a specific number appear in array |
| **Rule** | 1. Array is defined two dimensional / linear array |
| **Example** | **Input:**  source = 5,3,4,2,0,7,2,1,6,8  target = 2  **Output:**  2 |

# Linear Search part 2

|  |  |
| --- | --- |
| **Problem** | Find smallest and largest number from a linear array |
| **Rule** | 1. Array is defined two dimensional / linear array |
| **Example** | **Input:**  source = 5,3,4,2,0,7,2,1,6,8  **Output:**  {min: 0, max: 8} |

# Reverse String

|  |  |
| --- | --- |
| **Problem** | Write a function that return a reverse of a certain string |
| **Rule** | 1. Make your own function |
| **Example** | **Input:**  string = Hello World  **Output:**  dlroW olleH |

# Palindrome

|  |  |  |  |
| --- | --- | --- | --- |
| **Problem** | Write a function that check whether a string is palindrome or not | | |
| **Rule** |  | | |
| **Example** | **Input:**  s = stats  **Output:**  true | **Input:**  s = list  **Output:**  false | **Input:**  s = aibohphobia  **Output:**  true |

# ATM

|  |  |
| --- | --- |
| **Problem** | Make a simple ATM program that has a denomination of Rp 100.000, Rp 50.000, Rp 20.000, Rp 10.000. The program will calculate how many of each denomination to be received by the user according to the amount of money that will be taken. |
| **Rule** | 1. Check that the input is multiple of smallest available denomination 2. To make it simple, assumed that the ATM has unlimited amount of each denomination |
| **Example** | **Input:**  amount = 270000  **Output:**  {100000: 2, 50000: 1, 20000: 1} |

# Fibonacci

|  |  |  |  |
| --- | --- | --- | --- |
| **Problem** | Write a function that return the nth number of Fibonacci Sequence | | |
| **Rule** | 1. First and second number of the sequence defined as 0 and 1 2. Must be a single function 3. No loop are allowed | | |
| **Example** | **Input:**  n = 1  **Output:**  0 | **Input:**  n = 4  **Output:**  2 | **Input:**  n = 8  **Output:**  13 |

# Prime Number

|  |  |  |  |
| --- | --- | --- | --- |
| **Problem** | Write a function that check whether a number is prime number or not | | |
| **Rule** | 1. Must be a single function 2. No loop are allowed | | |
| **Example** | **Input:**  n = 3  **Output:**  true | **Input:**  n = 11  **Output:**  true | **Input:**  n = 15  **Output:**  false |