

Kompetensi: Functions and Conditionals

1. Number Processing

Disediakan sebuah function `numberProcessing` yang akan memproses angka dalam array. Function tersebut akan menerima parameter berupa array yang berisi angka. Function `numberProcessing` membutuhkan 3 function terpisah bernama `findMax`, `findMin`, dan `findMean`.

Tanpa mengubah function `numberProcessing`, selesaikan function `findMax`, `findMin`, `findMean`, `findOdds`, dan `findEvens` dengan benar!

Function `findMax`: menerima parameter berupa array angka dan mengembalikan salah satu angka yang memiliki nilai paling besar dari array tersebut.

Ex: `[2, 1, 4, 3]` => 4

Function `findMin`: menerima parameter berupa array angka dan mengembalikan salah satu angka yang memiliki nilai paling kecil dari array tersebut.

Ex: `[2, 1, 4, 3]` => 1

Function `findMean`: menerima parameter berupa array angka dan mengembalikan rata-rata dari seluruh angka dalam array. Apabila angka koma, lakukan pembulatan kebawah.

Ex: `[2, 1, 4, 3]` => 2

Function `findOdds`: menerima parameter berupa array angka dan mengembalikan angka-angka yang ganjil sebagai string dan dipisahkan dengan tanda '-'.

Ex: `[2, 1, 4, 3]` => "1-3"

Function `findEvens`: menerima parameter berupa array angka dan mengembalikan angka-angka yang genap sebagai string dan dipisahkan dengan tanda '-'.

Ex: `[2, 1, 4, 3]` => "2-4"

Apabila seluruh function tersebut selesai di buat, function `numberProcessing` akan mengembalikan nilai berikut:

Ex: `[2, 1, 4, 3]` => "Min: 1, Max: 4, Mean: 2, Odds: 1-3, Evens: 2-4"

2. String Manipulator

Disediakan sebuah function `stringManipulator` yang akan memproses sebuah string. Function tersebut akan melalui beberapa function kecil di dalamnya dan mengembalikan sebuah string baru yang telah berubah. Adapun function ini membutuhkan 3 function terpisah yaitu: `removeDash`, `capitalizeString`, dan `transformTextToNumber`.

Tanpa mengubah function `stringManipulator`, selesaikan function `removeDash`, `capitalizeString`, dan `transformTextToNumber` dengan benar!

Function `removeDash`: menerima parameter berupa string dan mengembalikan string baru dengan menghapus semua simbol '-' dan menggantinya dengan ' ' (spasi).

Ex: "hello-world" => "hello world"

Function `capitalizeString`: menerima parameter berupa string dan mengembalikan string baru yang merupakan kata yang telah dikapital. Kata yang kapital: huruf depan setiap kata besar / uppercase, dan huruf sisanya kecil / lowercase.

Ex: "hello world" => "Hello World"

Function `transformTextToNumber`: menerima parameter berupa string dan mengembalikan string baru dengan perubahan dimana untuk beberapa huruf, diubah menjadi bentuk angka dengan pola berikut:

Huruf i diganti menjadi 1

Huruf e diganti menjadi 3

Huruf s diganti menjadi 5

Huruf o diganti menjadi 0

Huruf a diganti menjadi 4

Ex: "helicopter fly away" => "h3l1c0pter fly 4w4y"

Apabila seluruh function tersebut selesai di buat, function **`stringManipulator`** akan mengembalikan nilai berikut:

Ex: "hello-world" => "H3ll0 W0rld"

Kompetensi: Arrays and Loopings

1. Number Cruncher

Disediakan sebuah function `numberCruncher` yang bertugas untuk menerima sebuah parameter berupa angka, dan mengembalikan array baru yang merupakan hasil penambahan angka di index array sebelumnya dengan dua angka yang berada di kiri dan kanannya. Apabila tidak ada angka di kiri atau kanan, maka dianggap sebagai 0.

Ex:

Input: [**2**, **5**, **1**, **3**]

Proses: [0 + **2** + 5 , 2 + **5** + 1, 5 + **1** + 3, 1 + **3** + 0]

Output: [7, 8, 9, 4]

2. Initial Grouping

Disediakan sebuah function `initialGrouping` yang bertugas untuk menerima sebuah parameter berupa array satu dimensi berisikan nama-nama. Function `initialGrouping` akan mengelompokkan nama-nama tersebut berdasarkan huruf depannya dan dipisahkan ke dalam array dua dimensi.

Ex:

Input: ["Budi", "Badu", "Joni", "Jono"]

Proses:

Huruf depan yang ditemukan: B dan J

Pisahkan nama yang depannya B, dan nama yang depannya J. Masukkan ke dalam array baru
Susun array baru sebagai berikut:

```
[  
  [ "Budi", "Badu" ],  
  [ "Joni", "Jono" ],  
]
```

Setiap array dimensi kedua diawali dengan huruf depan yang mengelompokkan nama tersebut.

Kompetensi: Objects

1. Fox Activities

Buatlah sebuah class Fox yang menerima parameter awal berupa species dan favouriteFood. Fox memiliki property/attribute yaitu species dan favouriteFood, dan mempunyai method atau function sebagai berikut:

howl(): function ini akan mereturn sebuah string yang berisi: "Howl! I'm a/an <species> Fox !!!"
Pastikan jika nama species huruf depannya merupakan huruf hidup/vokal, gunakan an. Jika nama species huruf depannya merupakan konsonan/huruf mati, gunakan a.

meet(foxObj): function yang menerima object fox lain ini akan mereturn string yang berisi:

"hello nice to meet you" apabila nama spesies fox berbeda dengan nama species fox yang menjadi parameter.

"Hey brother!" apabila nama spesies fox sama dengan nama species fox yang menjadi parameter.

changeGenetics(newSpecies): function yang menerima parameter string yang berisi nama spesies baru, dan akan mengubah nama spesies Fox tersebut.

Kompetensi: Recursive

1. TambahTerus

Function tambahTerus akan menerima 1 parameter berupa angka.

Secara rekursif, function tambahTerus akan menambahkan setiap digit di angka tersebut.

Function tambahTerus akan berhenti saat angka sisa satu digit.

Ex:

Contoh 1 =====

tambahTerus(123) = 6

123 => $1 + 2 + 3 = 6$, dan 6 memiliki 1 digit maka return 6.

Contoh 2 =====

tambahTerus(1234) = 1

1234 => $1 + 2 + 3 + 4 = 10$, 10 masih 2 digit, maka teruskan fungsi.

10 => $1 + 0 = 1$, dan 1 memiliki 1 digit maka return 1.

Contoh 3 =====

tambahTerus(88888888) = 1

88888888 => $8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 = 64$, 64 masih 2 digit, maka teruskan fungsi.

64 => $6 + 4 = 10$, 10 masih 2 digit, maka teruskan fungsi.

10 => $1 + 0 = 1$, dan 1 memiliki 1 digit maka return 1.

Contoh 4 =====

tambahTerus(54) = 9

54 => $5 + 4 = 9$, dan 9 memiliki 1 digit maka return 9.

Kompetensi: Rocket!

1. British National Crisis

Tiga narapidana kabur dari penjara London. Tiga narapidana ini memiliki nomor sel 1, 5, dan 9. Tiga narapidana ini bergabung dan menyatukan kekuatan dan pikiran untuk mengacau di Inggris dengan melakukan pengeboman di tempat-tempat yang berbeda. Setiap kali mereka mau melakukan pengeboman, mereka mengirimkan sebuah kode aneh ke Scotland Yard, markas kepolisian Inggris. Seorang detektif pemula tengah mencoba menyelesaikan krisis tersebut.

Ternyata, kode tersebut merupakan sebuah pola. Kode tersebut merupakan angka 2,3,4,6,7, atau 8

yang ditulis dengan kata dalam bahasa Inggris, dan diacak. Seluruh karakter di kode tersebut apabila disusun dengan baik, akan menghasilkan beberapa kata berupa angka dalam bahasa Inggris.

District yang akan diserang selanjutnya adalah total dari angka yang ditemukan di kode tersebut!

Karena tiga narapidana ini benci dengan sel mereka, mereka tidak memasukkan angka 1, 5, dan 9.

Berikut ini adalah bahasa Inggris untuk angka yang digunakan:

TWO -> 2

THREE -> 3

FOUR -> 4

SIX -> 6

SEVEN -> 7

EIGHT -> 8

Contoh Kode 1:

WTWTHROETEO. Apabila diacak mendapatkan kombinasi TWO TWO THREE => 2, 2, dan 3. $2 + 2 + 3 = 7$, maka targetnya District 7!

Contoh Kode 2:

HSEVTEEING. Apabila diacak mendapatkan kombinasi SEVEN EIGHT => 7 dan 8. $7 + 8 = 15$, maka targetnya District 15!

Contoh Kode 3:

FNEXSIVUSEOR. Apabila diacak mendapatkan kombinasi FOUR SIX SEVEN => 4, 6, dan 7. $4 + 6 + 7 = 17$, maka targetnya District 17!

Contoh Kode 4:

WTO. Apabila diacak mendapatkan kombinasi TWO => 2. Maka targetnya District 2!

HINT:

Urutan angka mana yang keluar lebih dulu tidak penting!

Yang penting adalah kita mendapatkan district berapa yang menjadi target!