

1 留学生 2 年 JavaScript 後期中間試験

【問題 1】 次の問題を読み、正しいものには○を、誤りには×を記入しなさい。(20 点 @2 点)

1. 引数は「いんすう」と読む (×)
2. 変数の宣言は、var を使う (○)
3. 文字列は、[と] を使って囲む。 (×)
4. 変数 aaa を宣言 (var aaa;) したとき、aaa の初期値は、0 である。 (×)
5. 変数の初期化を行わない時、変数の値は不定となる。 (○)
6. 繰り返しを行うときには、while 文か、for 文を使う。 (○)
7. 多分岐を実現する命令に、switch 文がある。 (○)
8. 条件式の結果で処理を二つに分ける命令に、switch 文がある。 (×)
9. 繰り返しで break を使用すると、以降の処理を実行せず先頭に戻る。 (×)
10. 繰り返しでは continue 命令を使うと、残りの部分は実行せず繰り返しを終了する。 (×)

【問題 2】 次のプログラムを読み、ループの処理 (A) が何回実行されるか答えなさい。(20 点 @4 点)

| 問題 | プログラム | 解答 |
|----|--|-----|
| 1 | <pre>function foo(j){ for (var x = 1 ; x<10 ; x++){ document.write("hello "); ← (A) } } foo(3);</pre> | 9 回 |
| 2 | <pre>function alice() { for (var x = 1; x < 10; x++) { if (x == 4){ break; } document.write("hello "); ← (A) } } alice();</pre> | 3 回 |

| | | |
|---|--|---|
| 3 | <pre> function alice() { for (var x = 1; x < 10; x++) { if (x <= 6){ continue; } document.write("hello "); } } alice(); </pre> | <p>← (A)</p> <p>3 回</p> |
| 4 | <pre> function Bob(No) { var end = No % 10; while (end > 0){ document.write("hello "); end--; } } Bob(学績番号); </pre> | <p>← (A)</p> <p>学績番号の末尾の 1 桁の回数</p> |
| 5 | <pre> function Green(param) { var max; switch (param) { case 1: case 2: case 3: max = 2; case 4: case 5: max = 3; default: max = 4; } for(var i = 0 ; i<=max; i++){ document.write("hello world "); } } Green(学績番号 % 5); </pre> | <p>← (A)</p> <p>5 回</p> |

【問題3】 次の関数の結果（変数 RET）がどのようなになるか答えなさい。問題文中の【学籍番号】には自分の学籍番号が入るものとして答えなさい。

(20 点 @4 点)

| 問題 | プログラム | 解答 |
|----|---|------|
| 1 | <pre>function doAction(p1, p2){ var ans; ans = p1 - p2 ; return ans; } RET = doAction(100, 2);</pre> | 98 |
| 2 | <pre>function doAction(p1 , p2){ var ans ; ans = p1 + p2; return ans; } RET = doAction("1" , 2);</pre> | "12" |
| 3 | <pre>function doAction() { var ary = new Array(5); for (var j = 0; j < 5 ; j++){ ary[j] = 2; } for (var k = 1 ; k < 5 ; k++){ ary[k] = ary[k] + ary[k-1]; } var ans ; ans = ary[3] + ary[4]; return ans; } RET = doAction();</pre> | 18 |
| 4 | <pre>function nPr(n, r) { var seki = 1; for (var s = 1; s <= r; s++) { seki *= s; } var seki2 = 1; for (var ss = 1; ss <= r; ss++) { seki2 *= n; n--; } return seki2 / seki; } var RET = nPr(8, 3);</pre> | |

| | | |
|---|---|-----|
| 5 | <pre>function binPower(n){ var ans = 1; for(var i = 1; i <= n ; i++){ ans *= 2; } return ans; } RET = binPower(10);</pre> | 512 |
|---|---|-----|

【問題 4】 次の関数の仕様を読み、関数を作成しなさい。

(40 点 @5 点)

| 問題 | 仕様 | |
|--|-------|--|
| 1 | 関数名 | foo |
| | パラメータ | p |
| | 処理 | もし、パラメータ p が 1 のとき、変数 ans に「おはよう」をセットする パラメータ p が 1 ではない時、変数 ans に「こんにちは」をセットする |
| | 戻り値 | ans |
| 解答欄 | | |
| <pre>function foo(p){ var ans; if (p == 1) { ans = "おはよう"; }else { ans = "こんにちは"; } return ans; }</pre> | | |

| 問題 | 仕様 | |
|---|-------|--|
| 2 | 関数名 | bar |
| | パラメータ | p , q |
| | 処理 | ans = 1; ans = p * (p-1) * (p-2) * (p-3)*...*(q) ※ p から q までの掛け算を行う。 |
| | 戻り値 | ans |
| 解答欄 | | |
| <pre>function bar(p , q){ var ans = 1; while (p >= q){ ans = ans * p ; p = p -1 ; } return ans; }</pre> | | |

| 問題 | 仕様 | |
|---|-------|--|
| 3 | 関数名 | kaijo |
| | パラメータ | p |
| | 処理 | p > 1 のとき、ans ←p * kaijo(p - 1) p > 1 以外のとき、ans ← 1 |
| | 戻り値 | ans |
| 解答欄 | | |
| <pre>function kaijo (p){ var p; if (p > 1){ ans = p * kaijo (p - 1); }else { ans = 1; } return ans ; }</pre> | | |

| 問題 | 仕様 | |
|---|-------|---|
| 4 | 関数名 | foo |
| | パラメータ | x |
| | 処理 | ans ← 1 + 2 + 3 + ... + x (1 から x までの合計を行う) 繰り返しには、for 文を使用すること |
| | 戻り値 | ans |
| 解答欄 | | |
| <pre>function foo(x){ var ans = 0 ; for (var i = 1 ; i<= x ; i ++){ ans += i ; } return ans; }</pre> | | |

| 問題 | 仕様 | |
|--|-------|---|
| 5 | 関数名 | compareNumbers |
| | パラメータ | param1 , param2 |
| | 処理 | param1 < maram2 のとき、ans を 1 に それ以外の時、ans を 0 にする |
| | 戻り値 | ans |
| 解答欄 | | |
| <pre>function compareNumber(param1 , param2){ if (param1 < param2){ ans = 1; } else { ans = 0; } return ans; }</pre> | | |

| 問題 | 仕様 | |
|--|-------|--|
| 6 | 関数名 | repeatHelloMessage |
| | パラメータ | kai |
| | 処理 | 文字列” hello” を改行（” ” ）しながら、パラメータで指定された回数、同一ウィンドウに出力する。 繰り返しには、for を使用すること。 |
| | 戻り値 | なし |
| 解答欄 | | |
| <pre>function repeatHelloMessage(kai){ for(var l = 1 ; l <= kai ; kai ++){ document.write(“hello ”); } }</pre> | | |

| 問題 | 仕様 | |
|--|-------|---|
| 7 | 関数名 | kaijo |
| | パラメータ | n |
| | 処理 | 変数 n を宣言する もし、n>1 なら ans ← n*kaijo(n-1) それ以外なら、 ans ← 1 |
| | 戻り値 | ans |
| 解答欄 | | |
| <pre>function kaijo(n){ var ans; if (n > 1){ ans = n * kaijo(n -1); } else { ans = 1; } return ans; }</pre> | | |

| 問題 | 仕様 | |
|--|-------|--|
| 8 | 関数名 | countDownNumber |
| | パラメータ | n |
| | 処理 | 1 行ずつ数字を HTML の中に表示する。最初の行は n、次の行は n-1、...、最後の行は 1 とする 数字 x を出力して改行するには、document.write(x + " ")を使用する。 for による繰り返しを使用すること |
| | 戻り値 | なし |
| 解答欄 | | |
| <pre>function countDownNumber(n){ for (; n > 0; n--){ document.write(n + " "); } }</pre> | | |

| 問題 | 仕様 | |
|--|-------|--|
| 9 | 関数名 | showSectionForName |
| | パラメータ | sectionIdName |
| | 処理 | var item = document.getElementById(sectionIdName); item.style.display = "block"; |
| | 戻り値 | なし |
| 解答欄 | | |
| <pre>function showSectionfForName(sectionIdName){ var item = document.getElementById(sectionIdName); item.style.display = "block" ; }</pre> | | |

| 問題 | 仕様 | |
|--|-------|--|
| 10 | 関数名 | initializeNumber |
| | パラメータ | numberOfItem, initializeParameter |
| | 処理 | 配列の宣言により名前が aaa の配列で、要素数が numberOfItem 個の配列を作る。 while による繰り返しを使い、すべての要素を initializeParameter の値で初期化する |
| | 戻り値 | 配列名 |
| 解答欄 | | |
| <pre>function initializeNumber(numberOfItem, initializeParameter){ var aaa = new Array(numberOfItem); var index = 0; while (index < numberOfItem){ aaa[index] = initializeParameter; } }</pre> | | |


```
}  
  return aaa;  
}
```