# 留学生2年　JavaScript　後期中間試験

## 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。 （　〇　）
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、残りの部分は実行さず繰り返しを終了する。 （　×　）

## 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 | function foo( j){  for ( var x = 1 ; x<10 ; x++ ){  document.write( "hello <br>");　　　　←（A)  }  }  foo( 3); | 9回 |
| 2 | function alice() {  for (var x = 1; x < 10; x++) {  if ( x == 4){  break;  }  document.write("hello<br>");　　　　　←（A）  }  }  alice( ); | 3回 |

|  |  |  |
| --- | --- | --- |
| 3 | function alice() {  for (var x = 1; x < 10; x++) {  if ( x <= 6){  continue;  }  document.write("hello<br>");　　　　　←（A)  }  }  alice( ); | 3回 |
| 4 | function Bob( No) {  var end = No % 10;  while ( end > 0 ){  document.write("hello<BR>");　　　　　　　←（A)  end--;  }  }  Bob( 学績番号); | 学績番号の末尾の1桁の回数 |
| 5 | 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<br>");　　　　←（A)  }  }  Green( 学績番号 % 5 ); | 5回 |

## 次の関数の結果（変数RET）がどのよう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 | function doAction( p1, p2 ){  var ans;  ans = p1 - p2 ;  return ans;  }  RET = doAction( 100, 2); | 98 |
| 2 | function doAction( p1 , p2 ){  var ans ;  ans = p1 + p2;  return ans;  }  RET = doAction( "1" , 2 ); | "12"  ※文字列でない-2 |
| 3 | 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(); | 18 |
| 4 | 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;  }  RET = nPr(8, 3); | 56 |

|  |  |  |
| --- | --- | --- |
| 5 | function binPower( n){  var ans = 1;  for( var i = 1; i <= n ; i++){  ans \*= 2;  }  return ans;  }  RET = binPower( 10); | 1024 |

## 次の関数の仕様を読み、関数を作成しなさい。 （40点　@4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「こんにちわ」をセットする |
| 戻り値 | ans |
| 解答欄 | | |
| function foo( p){  var ans;  if ( p == 1 ) {  ans = "おはよう";  }else {  ans = "こんにちわ";  }  return ans;  } | | |

|  |  |  |
| --- | --- | --- |
| 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 | | |
| function bar( p , q ){  var ans = 1;  while ( p >= q ){  ans = ans \* p ;  p = p -1 ;  }  return ans;  } | | |

|  |  |  |
| --- | --- | --- |
| 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 | | |
| function kaijo ( p ){  var ans;  if ( p > 1 ){  ans = p \* kaijo ( p - 1 );  }else {  ans = 1;  }  return ans ;  } | | |

|  |  |  |
| --- | --- | --- |
| 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 | | |
| function　foo( x ){  var ans = 0 ;  for ( var i = 1 ; i<= x ; i ++ ){  ans += i ;  }  return ans;  } | | |

|  |  |  |
| --- | --- | --- |
| 問題 | 仕様 | |
| 5 | 関数名 | compareNumbers |
| パラメータ | param1 , param2 |
| 処理 | parame1 < 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 | | |
| function compareNumber( param1 , param2 ){  var ans;  if ( param1 < param2 ){  　ans = 1;  } else {  　ans = 0;  }  return ans;  } | | |

|  |  |  |
| --- | --- | --- |
| 問題 | 仕様 | |
| 6 | 関数名 | repeatHelloMessage |
| パラメータ | kai |
| 処理 | 文字列”hello”を改行（”<BR>”）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 | | |
| function repeatHelloMessage( kai ){  　　for( var I = 1 ; I <= kai ; i ++ ){  　　　　document.write( “hello <br>”);  　　}  } | | |

|  |  |  |
| --- | --- | --- |
| 問題 | 仕様 | |
| 7 | 関数名 | kaijo |
| パラメータ | n |
| 処理 | 変数ans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 | | |
| function kaijo( n){  　　var ans;  　　if ( n > 1 ) {  　　　　ans = n \* kaijo( n -1 );  　　} else {  　　　　ans = 1;  　　}  　　return ans;  } | | |

|  |  |  |
| --- | --- | --- |
| 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nを出力して改行するには、document.write( n + "<br>")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 | | |
| function countDownNumber( n ){  　　for ( ; n > 0 ; n-- ){  　　　　document.write( n + "<br>");  　　}  } | | |

|  |  |  |
| --- | --- | --- |
| 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 | | |
| function showSectionfForName( sectionIdName ){  　　var item = document.getElementById( sectionIdName);  　　item.style.display = “block”;  } | | |

|  |  |  |
| --- | --- | --- |
| 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 | | |
| function initializeNumber( numberOfItem, initializeParameter){  　　var aaa = new Array( numberOfItem);  　　var index = 0;  　　while ( index < numberOfItem){  　　　　aaa[index] = initializeParameter;  　　　　index ++;  　　}  　　return aaa;  } | | |