PRACTICA N O 4 FUNCIONES

Materia: Estructura de Datos – (SIS-312)

Docente: Ing. Limber Ruiz Molina

Auxiliar: Univ. Jhasmani Choqueticlla

Nombre: Univ. Rodriguez Condori Kevin Alexis

REALICE LAS PRUEBAS DE ESCRITORIO DE LAS SIGUIENTES FUNCIONES

|  |  |
| --- | --- |
| int **iguales** (int x, int y, int z)  {  if((x==y) && (y==z))  return 1;  else  if((x==y) || (x==z) || (y==z))  return 2;  else  return 3;  } | Prueba de escritorio con:  **iguales**(7,9,7)  Variables:  x y z return  7 9 7 2 |
| **iguales**(8,8,8)  Variables:  x y z return  8 8 8 1 |

1.-

2.-

|  |  |
| --- | --- |
| int **funcion** (int a, int b, int c, int d)  {  int m = a;  If (b>m)  m = b;  if (c > m)  m = c;  if (d > m)  m = d;  return m;  } | Prueba de escritorio con:  **función**(1,2,3,4)  Variables:  a b c d m return  1 2 3 4 1 4  2  3  4 |
| Prueba de escritorio con:  **función**(1,6,3,4)  a b c d m return  1 6 3 4 1 6  6  6  6 |
| Prueba de escritorio con:  **función**(6,2,7,4)  a b c d m return  6 2 7 4 6 7  6  7  7 |

3.-

|  |  |
| --- | --- |
| int **funcion** (int a)  {  int sum = 0;  for (int i = 0; i < a; i++) {  if (a % i == 0)  sum = sum + i;  }  return sum;  } | Prueba de escritorio con:  **función**(5)  Variables:  a sum i return  5 0 1 1  1 2  1 3  1 4  5 |
| Prueba de escritorio con:  **función**(6)  Variables:  a sum i return  6 0 1 6  1 2  3 3  6 4  6 5  6 |

4.-

|  |  |
| --- | --- |
| int **digito**(int A)  {  int m =0;  while (A > 0) {  if (A % 10 > m)  m = A%10;  A = A/10;  }  return m;  } | Prueba de escritorio con:  **digito**(234)  Variables:  A m return  234 0 4  23 4  2 4  0 |
| Prueba de escritorio con:  **digito**(2931)  Variables:  A m return  2931 0 9  293 1  29 3  2 9  0 |

REALICE LAS SIGUIENTES FUNCIONES:

5.- Realice una función que reciba como parámetro un número y retorne la suma de sus dígitos

Ejemplos: parámetro retorno

21 🡪 3

321 🡪 6

567 🡪 18

7784 🡪 26

|  |  |
| --- | --- |
| Código de la función | Prueba de escritorio |
| int **sumaDigitos** (int A)  {  int modulo = 0;  int suma = 0;  while(A > 0){  modulo = A%10;  suma += modulo;  A = A/10;  }  return suma;  } | **sumaDigitos**(21)  variables:  A modulo suma return  21 0 0 3  21 1 1  2 2 3  0 |
| **sumaDigitos**(321)  variables:  A modulo suma return  **321 0 0 6**  **32 1 1**  **3 2 3**  **0 3 6** |
| **sumaDigitos**(567)  variables:  A modulo suma return  **567 0 0 18**  **56 7 7**  **5 6 13**  **0 5 18** |
| **sumaDigitos**(7784)  variables:  A modulo suma return  **7784 0 0 26**  **778 4 4**  **77 8 12**  **7 7 19**  **0 7 26** |

6.- Realice una función que reciba como parámetro un número y retorne la cantidad de dígitos pares que

tiene

Ejemplos: parámetros retorno

317 🡪 0

248 🡪 3

3581 🡪 1

|  |  |
| --- | --- |
| Código de la función | Prueba de escritorio |
| int contarPares (int A)  {  int modulo = 0;  int contador = 0;  while (A > 0) {  modulo = A%10;  if (modulo % 2 == 0) {  Contador++;  }  A = A/10;  }  return contador;  } | **contarPares**(317)  Variables:  A modulo contador return  317 0 0 0  31 7 0  3 1 0  0 3 0 |
| **contarPares**(248)  Variables:  A modulo contador return  248 0 0 3  24 8 1  2 4 2  0 2 3 |
| **contarPares**(3581)  Variables:  A modulo contador return  3581 0 0 1  358 1 0  35 8 1  3 5 1  0 3 1 |

7.- La serie Fibonacci es: 1 1 2 3 5 8 13 21 34 55 89 . . . . . .

Realice una función que reciba como parámetro un número y retorne su respectivo de la serie Fibonacci.

Ejemplos: parámetro retorno

2 🡪 1

3 🡪 2

5 🡪 5

7 🡪 13

|  |  |
| --- | --- |
| Código de la función | Prueba de escritorio |
| int fibonacci (int A)  {  int number\_1 = 0;  int number\_2 = 1;  int suma = 0;  for (int i = 1; i <= A; i++) {  number\_1 = number\_2;  number\_2 = suma;  suma = number\_1 + number\_2;  }  return suma;  } | **fibonacci**(2)  Variables:  A number\_1 number\_2 suma i return  2 0 1 0 1 1  1 0 1 2  0 1 1 |
| **fibonacci**(3)  Variables:  A number\_1 number\_2 suma i return  3 0 1 0 1 2  1 0 1 2  0 1 1 3  1 1 2 |
| **fibonacci**(5)  Variables:  A number\_1 number\_2 suma i return  5 0 1 0 1 5  1 0 1 2  0 1 1 3  1 1 2 4  1 2 3 5  2 3 5 |
| **fibonacci**(7)  Variables:  A number\_1 number\_2 suma i return  7 0 1 0 1 13  1 0 1 2  0 1 1 3  1 1 2 4  1 2 3 5  2 3 5 6  3 5 8 7  5 8 13 |

8.- Realice una función que reciba como parámetro un número y retorne 1 si el número es un número

primo, caso contrario retorne 0

Un número primo es aquel solo es divisible entre 1 y entre si mismo.

Ejemplos: parámetro retorno

1 🡪 1

3 🡪 1

4 🡪 0

5 🡪 1

6 🡪 0

7 🡪 1

9 🡪 0

10 🡪 0

|  |  |
| --- | --- |
| Código de la funcion | Prueba de escritorio |
| int **esPrimo**(int N)  {  int a = 0;  int contador = 0;  for(int i = 2; i < N; i++){  if(N%i == 0){  Contador++;  }  }  if(contador == 0){  a = 1;  }  return a;  } | **esPrimo**(1)  Variables  N a contador i return  1 0 0 2 1  1 |
| **esPrimo**(3)  Variables  N a contador i return  3 0 0 2 1   1. 0 |
| **esPrimo**(4)  Variables  N a contador i return  4 0 0 2 0  **0 1 3**  **0 1** |
| **esPrimo**(5)  Variables  N a contador i return  5 0 0 2 1  0 0 3  0 0 4  0 0 |
| **esPrimo**(6)  Variables  N a contador i return  6 0 0 2 0  0 1 3  0 2 4  0 2 5  0 2 |
| **esPrimo**(7)  Variables  N a contador i return  7 0 0 2 1  0 0 3  0 0 4  0 0 5  0 0 6  0 0 7  1 |
| **esPrimo**(9)  Variables  N a contador i return  9 0 0 2 0  0 0 3  0 1 4  0 1 5  0 1 6  0 1 7  0 1 8  0 1 9  0 |
| **esPrimo**(10)  Variables  N a contador i return  10 0 0 2 0  0 1 3  0 1 4  0 1 5  0 2 6  0 2 7  0 2 8  0 2 9  0 2 10  0 |