

Universidad nacional amazónica de madre de dios

Carrera profesional: Ingeniería en sistemas e informática

Nombre y apellido: Jin alexander Olmedo Paredes

Curso: Algoritmos

Código: 22221042

Docente: Holgado Apaza Luis Alberto

Guía de laboratorio 5



Actividad 1

Descripción del problema

- 1) Desarrolle una solución que permita generar y visualizar los n primeros términos de la siguiente serie:
2/5, 5/9, 8/13, 11/17, 14/21.

1. Análisis

1.1. Entender el problema

entrada	Proceso		salida	
terminos	numerador	denominador	numerador	denominador
4	2+3	5+4	2	5
	5+3	9+4	5	9
	8+3	13+4	8	13
	11+3	17+4	11	17
3	2+3	5+4	2	5
	5+3	9+4	5	9
	8+3	13+4	8	13

1.2. Modelo

Términos=n Numerador=2 Denominador=5	Términos=n+1 Numerador=numerador+3 Denominador=denominador+4
--	--

1.3. Especificación del algoritmo

1.3.1. Diccionario de variables

Variables de entrada

terminos (como entero)

Variables de proceso

Numerador y denominador (como entero)

Variables de salida

Numerador y denominador (como entero)

1.3.2. Pre condición

{termino>0}

1.3.3. Acción del algoritmo

{Identificar los primeros términos de la sucesión }

1.3.4. Post condición

{Numerador y denominador > 0}

2. Diseño

2.1. Definición de algoritmo en c++

```
1  #include<iostream>
2  using namespace std;
3
4  int main() {
5      int contador;
6      float denominador;
7      int n;
8      float numerador;
9      cout << "Ingrese la cantidad de terminos a generar:" << endl;
10     cin >> n;
11     numerador = 2;
12     denominador = 5;
13     cout << numerador << "/" << denominador << endl;
14     contador = 2;
15     while (contador<=n) {
16         numerador = numerador+3;
17         denominador = denominador+4;
18         cout << numerador << "/" << denominador << endl;
19         contador = contador+1;
20     }
21     return 0;
22 }
23
24
```

Definición del algoritmo en pseint:

```
Algoritmo fracciones
    Definir n, contador Como Entero
    Definir numerador, denominador Como Real

    Escribir "Ingrese la cantidad de términos a generar:"
    Leer n

    numerador ← 2
    denominador ← 5

    Escribir numerador, "/", denominador

    contador ← 2

    Mientras contador ≤ n Hacer
        numerador ← numerador + 3
        denominador ← denominador + 4

        Escribir numerador, "/", denominador

        contador ← contador + 1
    FinMientras

FinAlgoritmo
```

Actividad 2)

Descripción del problema

- 2) Dado un valor inicial y valor final, obtener la suma y la cantidad de los números divisibles por 3 y 5 a la vez.

1. Análisis

1.1. Entender el problema

entrada		proceso		salida	
Valor inicial	Valor final	suma	cantidad	suma	cantidad
6	3	0	0	0	0
15	10	15	1	15	1
20	120	525	7	525	7

1.2. Especificación del algoritmo

1.2.1. Diccionario de variables

Variable de entrada (como entero)

Valor inicial, valor final

Variable de proceso

Suma, cantidad (como entero)

Variable de salida

Suma, cantidad (como entero)

1.2.2. Pre condición

{Valor inicial>0}, {valor final>0}

1.2.3. Acción del algoritmo

Mostrar la suma y la cantidad de números divisibles por 3 y 5 a la vez

1.2.4. Post condición

{Suma>=0}, {cantidad>=0}

2. Diseño

2.1. Definición del algoritmo en pseint

```

1  Algoritmo ejercicio2
2      Definir valor_inicial, valor_final, suma, cantidad Como Entero
3      Escribir "Ingrese el valor inicial: "
4      Leer valor_inicial
5      Escribir "Ingrese el valor final: "
6      Leer valor_final
7      suma ← 0
8      cantidad ← 0
9
10     Para i ← valor_inicial hasta valor_final Hacer
11         Si (i mod 3 = 0) y (i mod 5 = 0) Entonces
12             suma ← suma + i
13             cantidad ← cantidad + 1
14         FinSi
15     FinPara
16
17     Escribir "La suma de los números divisibles por 3 y 5 es: ", suma
18     Escribir "La cantidad de números divisibles por 3 y 5 es: ", cantidad
19
20 FinAlgoritmo

```

Codificación en c++

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int valorI, valorF, suma = 0, cantidad = 0;
8
9      cout << "Ingrese el valor inicial: ";
10     cin >> valorI;
11     cout << "Ingrese el valor final: ";
12     cin >> valorF;
13
14     for (int i = valorI; i <= valorF; i++) {
15         if (i % 3 == 0 && i % 5 == 0) {
16             suma += i;
17             cantidad++;
18         }
19     }
20
21     cout << "La suma de los números divisibles por 3 y 5 es: " << suma << endl;
22     cout << "La cantidad de números divisibles por 3 y 5 es: " << cantidad << endl;
23
24     return 0;
25 }
26

```

Actividad 3)

Descripción del problema

3) Encuentre el **mcm** de dos números a y b sabiendo que **$(a)(b)=(mcm)(mcd)$**

1. Análisis

1.1. Entender el problema

entrada		proceso		salida
a	b	mcm	mcd	mcm
4	6	24	2	24
8	9	72	17	72
12	24	288	12	288

1.2. Especificación del algoritmo

1.2.1. Diccionario de variables

Variable de entrada

A, b (como entero)

Variable de proceso

Mcm, mcd (como entero)

Variable de salida

Mcm (como entero)

1.2.2. Pre condición

{a>0} y {b>0}

1.2.3. Acción del algoritmo

Calcular el mcm de a y b

1.2.4. Post condición

{mcm>0}

2. Diseño

2.1. Definición del algoritmo en pseint

```
Algoritmo MCM_For
    Definir a, b, mcm, mcd Como Entero

    Escribir "Ingrese el primer número:"
    Leer a

    Escribir "Ingrese el segundo número:"
    Leer b

    Para i ← 1 Hasta a Hacer
        Si a mod i = 0 y b mod i = 0 Entonces
            mcd ← i
        FinSi
    FinPara

    mcm ← (a * b)

    Escribir "El MCM de ", a, " y ", b, " es: ", mcm
    Escribir "El MCD de " a, "y", b, "es:", mcd
FinAlgoritmo
```

Codificación en c++

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
6     int a;
7     int b;
8     float i;
9     int mcd;
10    int mcm;
11    cout << "Ingrese el primer número:" << endl;
12    cin >> a;
13    cout << "Ingrese el segundo número:" << endl;
14    cin >> b;
15    for (i=1;i<=a;i++) {
16        if (a%i==0 && b%i==0) {
17            mcd = i;
18        }
19    }
20    mcm = (a*b);
21
22    cout << "El MCM de " << a << " y " << b << " es: " << mcm << endl;
23    cout << "El MCD de " << a << "y" << b << "es:" << mcd << endl;
24    cout << "El Resultados de " << a*b << " es igual a" << mcd*mcm << endl;
25
26    return 0;
27 }
```

Actividad 4)

Descripción del problema

4) Calcule la suma de los cuadrados y cubos de los N primeros números naturales.

1. Análisis

1.1. Entender el problema

entrada	proceso		salida	
numero	Suma de cuadrados	Suma de cubos	Suma de cuadrados	Suma de cubos
8	204	1296	204	1296
13	819	8281	819	8281
16	1496	18496	1496	18496
6	91	441	91	441

1.2. Especificación del algoritmo

1.2.1. Diccionario de variables

variable de entrada

número (como entero)

variable de proceso

Suma de cuadrados y Suma de cubos (como entero)

variable de salida

Suma de cuadrados y Suma de cubos (como entero)

1.2.2. Pre condición

{Numero>0}

1.2.3. Acción del algoritmo

calcular la suma de cuadrados y la suma de cubos del número introducido

1.2.4. Post condición

{Suma de cuadrados==0} y {Suma de cubos==0}

2. Diseño

2.1. codificación del algoritmo en pseint

```
1  Algoritmo ejercicio4
2      Definir N, i, sCuadrados, sCubos como Entero
3
4      Escribir "Ingrese un numero : "
5      Leer N
6
7      suma_cuadrados = 0
8      suma_cubos = 0
9
10     Para i = 1 Hasta N Con Paso 1 Hacer
11         sCuadrados = sCuadrados + i * i
12         sCubos = sCubos + i * i * i
13     FinPara
14
15     Escribir "La suma de los cuadrados de los primeros ", N, " numeros naturales es: ", sCuadrados
16     Escribir "La suma de los cubos de los primeros ", N, " numeros naturales es: ", sCubos
17
18 FinAlgoritmo
19
```

codificación en c++

```
1  #include<iostream>
2
3  using namespace std;
4
5  int main() {
6      int i;
7      int n;
8      int sCuadrados = 0, sCubos = 0;
9      cout << "Ingrese un numero : " << endl;
10     cin >> n;
11     sCuadrados = 0;
12     sCubos = 0;
13     for (i=1; i<=n; i++) {
14         sCuadrados+=i*i;
15         sCubos+=i*i*i;
16     }
17     cout << "La suma de los cuadrados de los primeros " << n << " numeros naturales es: " << sCuadrados << endl;
18     cout << "La suma de los cubos de los primeros " << n << " numeros naturales es: " << sCubos << endl;
19     return 0;
20 }
21
22
```

Actividad 5)

Descripción del problema

- 5) Calcule la n -ésima potencia de un número teniendo en cuenta que el exponente puede ser negativo. Considere el caso que cualquier número elevado a la potencia cero da la unidad.

1. Análisis

1.1. Entender el problema

entrada		proceso	salida
base	exponente	resultado	resultado
5	6	15625	15625
2	3	8	8
6	4	1296	1296
8	4	4096	4096

1.2. Especificación del algoritmo

1.2.1. Diccionario de variables

Variable de entrada

Base y exponente (como real)

Variable de proceso

Resultado (como entero)

Variable de salida

Resultado (como entero)

1.2.2. Pre condición

{Base>0}

1.2.3. Acción del algoritmo

Calcular la n -ésima potencia de un número dado, teniendo en cuenta que el exponente puede ser negativo

1.2.4. Post condición

{resultado>=1}

2. Diseño

2.1. codificación del algoritmo en pseint

```
1  Algoritmo PotenciaN
2  Definir base, exponente Como Real
3  Definir resultado Como Entero
4  Escribir "Ingrese el número base: "
5  Leer base
6
7  Escribir "Ingrese el exponente: "
8  Leer exponente
9
10 si exponente = 0 entonces
11     resultado = 1
12 sino
13     si exponente > 0 entonces
14         resultado = 1
15         para i = 1 hasta exponente hacer
16             resultado = resultado * base
17         finpara
18     sino
19         resultado = 1
20         para i = 1 hasta abs(exponente) hacer
21             resultado = resultado * base
22         finpara
23         resultado = 1/resultado
24     finsi
25 FinSi
26 Escribir "El resultado es: ", resultado
27 FinAlgoritmo
28
```

Codificación en c++

```
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main() {
7      double base, exponente, resultado;
8
9      cout << "Ingrese la base: ";
10     cin >> base;
11
12     cout << "Ingrese el exponente: ";
13     cin >> exponente;
14
15     if (exponente == 0) {
16         resultado = 1;
17     } else if (exponente > 0) {
18         resultado = 1;
19         for (int i = 1; i <= exponente; i++) {
20             resultado *= base;
21         }
22     } else {
23         resultado = 1;
24         exponente *= -1;
25         for (int i = 1; i <= exponente; i++) {
26             resultado *= base;
27         }
28         resultado = 1 / resultado;
29     }
30
31     cout << "El resultado es: " << resultado << endl;
32
33     return 0;
34 }
```