

CHULETA AUTORIZADA PARA EL PRIMER PARCIAL

Normas, temario y ejemplos

Introducción

Este recurso está diseñado como una guía para estudiantes de primer curso en la asignatura de "Fundamentos de Programación" que tienen la oportunidad de llevar una chuleta autorizada al examen. La guía aborda la necesidad de reunir, en un solo documento, información clave que no se encuentra completamente en los recursos previos disponibles. Además, incluye ejemplos reales de chuletas elaboradas por alumnos, facilitando la comprensión de su estructura y contenido. Está orientado a estudiantes con conocimientos básicos en matemáticas, algoritmia y programación, y se recomienda revisar materiales previos relacionados para un mejor aprovechamiento del recurso.

Normas:

Para que la chuleta sea autorizada debe cumplir una serie de normas que se recogen a continuación. El no cumplimiento de estas normas resultará en la retirada inmediata de la chuleta y la prohibición de su uso en el examen. Las normas que debe cumplir son:

- Debe estar escrita a mano
- No se aceptan fotocopias de chuletas hechas a mano.
- Tiene que ser original, es decir, el autor de la chuleta debe ser el que hará uso de ella.
- Solo se permiten chuletas de un folio escrito por ambas caras
- Se permite escribir en ella cualquier cosa que el alumno considere oportuno para aprobar

Temario:

Es importante que lo primero que hagas es enterarte bien de qué temas se van a evaluar en el examen. No pierdas tiempo preparándote cosas que no entran en el primer parcial y pregunta a profesores y compañeros si tienes dudas.

En nuestro caso, se nos examinó de **algoritmia** (sumatorios, productorios, bucles, vectores, etc...) e **interpolación** (por sistema de ecuaciones, por Lagrange y por Newton; sólo la parte matemática).

Te recomendamos que hagas todos los ejercicios propuestos en clase y por Moodle (a resolver por el alumno) relacionados con estos temas.

Si necesitas repasar este temario te recomendamos visitar [esta web](#) y buscar allí todos los recursos relacionados con lo que necesitas. También encontraras enlaces a algunos videos que ayudaron a algún integrante del equipo a prepararse este parcial.

Ejemplos:

A continuación, te compartimos una recopilación de algunos ejemplos de chuleta autorizada hechas por parte de miembros de nuestro equipo.

- Chuleta 1:

Organigramas

Bucles:

- **If / If/else / If/else/if**
 Símbolo para preguntar una condición, si se cumple se ejecuta una cosa, sino otra. No opera.
- **While** → si es verdad lo del nombre, entra en el bucle → realiza las operaciones marcadas y vuelve a preguntar, si se vuelve a cumplir, vuelve a entrar, sino sale. (Mientras que... (hacia lo mismo que de preguntar hasta que (lo queramos)).
- **For** → Sabemos cuantas veces se repite el bucle
 $i = 1, n$, (2) → salt que realiza i.
 valores que toma la variable

Factorial / Productorio

$$Fac = \prod_{i=2}^n x$$

```

    graph TD
      A[1] --> B[fac = 1]
      B --> C{i = 2, n}
      C --> D[fac = fac * x]
      D --> C
      C --> E[Fin]
    
```

Sumatorio

$$S = \sum_{i=1}^m A_i$$

```

    graph TD
      A[m, A, x] --> B[S = 0]
      B --> C{j = 1, m, i + 1}
      C --> D{i != 3}
      D -- si --> E[S = S + A_i]
      E --> C
      D -- no --> F[i = i + 1]
      F --> C
      C --> G[Fin]
    
```

Nº entero a binario

```

    graph TD
      A[Num entero] --> B[Binario = C()]
      B --> C[Num. par = Num entero]
      C --> D{Num. par = 0}
      D --> E[resto = Num. p % 2]
      E --> F[Binario = C(resto, Binario)]
      F --> G[Num. p = floor(Num. p / 2)]
      G --> D
      G --> H[Fin]
    
```

Diagramas de flujo:

- Diagrama 1: Condición "No" / "Si" con caminos divergentes.
- Diagrama 2: Condición "No" / "Si" con caminos que convergen.
- Diagrama 3: Condición "No" / "Si" con un bucle "Mientras que" y un "Fin" al salir.
- Diagrama 4: Condición "No" / "Si" con un bucle "Mientras que" y un "Fin" al salir.

Notas:

- Mayor > Menor <
- Mayor/igual > =
- Menor/igual < =
- Distinto (≠) !=
- Igual ==
- Inversa \neq
- A^{-1} = solve(A)
- % * % → producto matricial

Pasos a seguir:

- 1º Resumir / priorizar fórmula (sumatorios, productorios) (de dentro a fuera).
- 2º Izq. y der. → hacer dentro bucles for como subíndices en letra (y en su orden).
- 3º poner el final de fórmula total.
- 4º Der. y der. → De fuera a dentro ir poniendo los bucles for necesarios para cada productorio/sumatorio y poner el final del bucle de fórmula.
- 5º Si dentro de los fórmulas hubiera alguna matriz, vector, factorial que hay que hacer (porque se dice lo que vale) lo hay que hacer antes de todo.
- 6º poner al principio del todo los valores que necesitamos y que hay que pedir.

⚠️ No hay que especificar que son matrices, o vectores si no hay que calcularlos y si no tiene nada no hay que poner de que valores depende ni calcularlos.

Regla de Heron (músculos cuadrados): $x_{n+1} = \frac{1}{2} (x_n + \frac{S}{x_n})$ (subíndices)

Serie de Taylor para seno: $\sin x = \sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{2k-1}}{(2k-1)!}$

Serie de Maclaurin para e^x: $e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$

Para comprobar/saber que algo es máximo o mínimo se derivaba y se crean dos bucles if (o if/else/if) primero con $i > \text{max} \rightarrow \text{no} < \text{min} \rightarrow \text{no}$ vuelve al for.
 Si para un mismo vector se tienen los límites distintos hacemos un bucle if para coger solo el mayor y crear así el vector. luego cada prod/sum va con su límite correspondiente.

Interpolación

Resolvimos por sistemas de ecuaciones: vector soporte $S = [x_1, x_2, \dots, x_n]$
 su longitud determina el grado del polinomio $(n-1)$.

$$p(x) = a_0 x^0 + a_1 x^1 + a_2 x^2 + a_3 x^3 + \dots + a_n x^{n-1}$$

(dependiendo del soporte que nos den n dar la función (o la que vale S), es decir $f(x)$ den copiamos más o menos)

Obtendremos un matriz M

$$\begin{pmatrix} 1 & \dots & s_1 & \dots & s_1^{(j-1)} & \dots & s_1^{(n-1)} \\ \vdots & & \vdots & & \vdots & & \vdots \\ 1 & \dots & s_2 & \dots & s_2^{(j-1)} & \dots & s_2^{(n-1)} \\ \vdots & & \vdots & & \vdots & & \vdots \\ 1 & \dots & s_i & \dots & s_i^{(j-1)} & \dots & s_i^{(n-1)} \\ \vdots & & \vdots & & \vdots & & \vdots \\ 1 & \dots & s_n & \dots & s_n^{(j-1)} & \dots & s_n^{(n-1)} \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f(s_1) \\ f(s_2) \\ \vdots \\ f(s_n) \end{pmatrix}$$

$A = M^{-1} \cdot B$
Resolver el sistema por Gauss

$$P(I) = \sum_{k=1}^n a(k) * S(I)^{(k-1)}$$

Calculamos el valor que nos piden substituyendolo en $p(x)$.

Problemas: causa grandes errores en los extremos (con polinomios de alto grado) \rightarrow Fenómeno de Runge
 polinomios de alto grado $(>20) \rightarrow$ matriz singular o mal condicionada \rightarrow impide solución matemática estable

Polinomios en base de Lagrange:

$$P(x) = \sum_{i=1}^n f(s_i) \cdot \left(\prod_{j \neq i} \frac{x - s_j}{s_i - s_j} \right)$$

Puntos de soporte $(s_i = j)$ de valores

Necesitamos saber el punto de aproximación

El pol. en base de Lagrange asociado a $S(j)$, es aquel que tiene un valor 1 en $S(j)$ y 0 para el resto

vector $L_j(x) = C_{j,1} x^{n-1} + C_{j,2} x^{n-2} + \dots + C_{j,n-1} x + C_{j,n}$

$$L_1(x) = \left[\frac{x - s_2}{s_1 - s_2} \right] \cdot \left[\frac{x - s_3}{s_1 - s_3} \right] \cdot \dots$$

$$L_2(x) = \left[\frac{x - s_1}{s_2 - s_1} \right] \cdot \left[\frac{x - s_3}{s_2 - s_3} \right] \cdot \dots$$

Substituímos los valores del soporte $S(s_1, s_2, s_3, \dots, s_n)$

Para calcular el valor aproximado multiplicamos los L_i por $f(s_i)$ en cada punto y simplificamos el polinomio y substituímos la x por el valor.

$$\text{Valor aprox.} = f(s_1) \cdot \left[\frac{x - s_2}{s_1 - s_2} \right] \cdot \left[\frac{x - s_3}{s_1 - s_3} \right] + f(s_2) \cdot \left[\frac{x - s_1}{s_2 - s_1} \right] \cdot \left[\frac{x - s_3}{s_2 - s_3} \right] + \dots$$

Problemas: hay que repetir el proceso para cada valor que queremos aproximar.
 Si pide calcular el soporte $S = \{s_0, s_1, s_2\} \Rightarrow L_1 = \dots$ $L_1(s_0) = 0$ $L_1(s_1) = 1$ $L_1(s_2) = 0$
 despreciar valores y los posibles soportes no pueden tener elementos repetidos y de igual orden.

Método de Newton: sirve para calcular el pol. si se añade un nuevo punto de soporte sin tener que repetir el cálculo desde el inicio.

$$P_{k+1}(x) = P_k(x) + C_{k+1} \cdot N_{k+1}(x)$$

$$N_{k+1}(x) = \prod_{j=0}^k (x - s_j)$$

$$C_{k+1} = \frac{f(s_{k+1}) - P_k(s_{k+1})}{s_{k+1} - s_0}$$

Tabla para facilitar cálculos:

$$f[s_0] \rightarrow f[s_0, s_1] = \frac{f[s_1] - f[s_0]}{s_1 - s_0}$$

$$f[s_1] \rightarrow f[s_1, s_2] = \frac{f[s_2] - f[s_1]}{s_2 - s_1}$$

$$f[s_2] \rightarrow f[s_2, s_3] = \frac{f[s_3] - f[s_2]}{s_3 - s_2}$$

sin tabla

$$f[s_0, s_1] = \frac{f[s_1] - f[s_0]}{s_1 - s_0}$$

$$f[s_0, s_1, s_2] = \frac{f[s_2] - f[s_0]}{s_2 - s_0}$$

$$f[s_0, s_1, s_2, s_3] = \frac{f[s_3] - p_2(s_3)}{(s_3 - s_0)(s_3 - s_1)(s_3 - s_2)}$$

$$f[s_0, s_1, s_2, s_3] = \frac{f[s_3] - p_2(s_3)}{(s_3 - s_0)(s_3 - s_1)(s_3 - s_2)}$$

- Chuleta 2:

OBJETIVO INTERMEDIARIO → DETERMINAR COEFICIENTES DE UN POLINOMIO A PARTIR DE VALORES DE LAS VARIAS (SOPORTE) (n+1) VARIAS

1) SISTEMA LINEAL DE ECUACIONES
 DATOS: VALORES VARIAS (X ASIGNA) → $S = c(s_1, s_2, s_3)$
 VALORES DE $f(x)$ EN X → $f(s_i) = f(c(s_1, s_2, s_3))$
 n = LARGO DE ASIGNA = CANTIDAD MIN DE X = 3
 GRADO POLINOMIO = p^{n-1}
 a = COEFICIENTES CANONICOS
 x → VALORES CANONICOS

INVERSIÓN MATRIZ:
 $A^{-1} = \frac{1}{|A|} \cdot \text{adj}(A)$
 MATRIZ DE LOS MENORES ENTERADOS (n x n) VALORES CANONICOS FILAS Y COLUMNAS
 POLINOMIO: $P(x) = a_0 \cdot x^n + a_1 \cdot x^{n-1} + \dots + a_n \cdot x^0$
 $f(s_i) = a_0 \cdot s_i^n + a_1 \cdot s_i^{n-1} + \dots + a_n \cdot s_i^0$
 $f(s_j) = a_0 \cdot s_j^n + a_1 \cdot s_j^{n-1} + \dots + a_n \cdot s_j^0$
 $M_{ij} = s_i^{j-1}$
 $a = M^{-1} \cdot b$

2) BASES/COEFICIENTES DE LAGRANGE
 $P(x) = \sum_{i=0}^n f(s_i) \cdot L_i(x)$
 $L_i(x) = \prod_{j \neq i} \frac{x - s_j}{s_i - s_j}$
 DATOS: S = VALORES DE SOPORTE (s1, s2, ...)
 X = VALOR DE CUALQUIER LAGRANGE
 n = LONGITUD DE X
 z = VALORES CANONICOS ENTEROS DE X (f(s1), f(s2), ...)

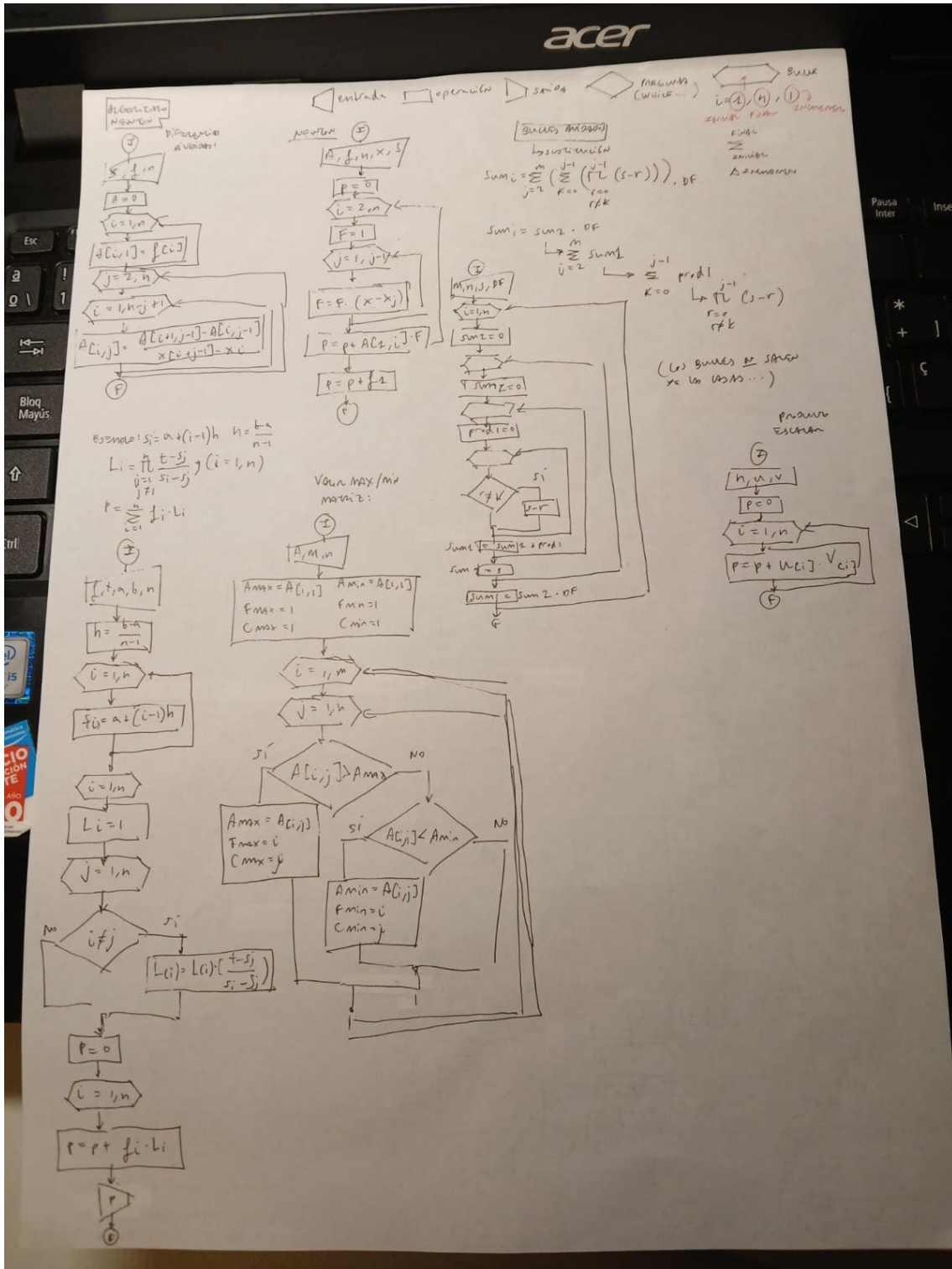
3) NEWTON
 L = MÓDULO DE DIFERENCIAS DIVIDIDAS
 FORMULA DE NEWTON: $P(x) = f(s_1) + f[x_1, x_2] \cdot (x - x_1) + f[x_1, x_2, x_3] \cdot (x - x_1) \cdot (x - x_2) + \dots$
 DATOS: S = c(s1, s2, ... s2)
 f(s) = f(s1), f(s2)
 n = NÚM DE S
 GRADO POLINOMIO: n-1

FORMULA GENERAL: $P(x) = f + \sum_{i=2}^n [f[x_1, x_2, \dots, x_i]] \cdot \prod_{j=1}^{i-1} (x - x_j)$

SOPORTE	VALOR	ORDEN 1	ORDEN 2	ORDEN 3
x1	f1	$f[x_1, x_2] = \frac{f_2 - f_1}{x_2 - x_1}$	$f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1}$	$f[x_1, x_2, x_3, x_4] = \frac{f[x_2, x_3, x_4] - f[x_1, x_2, x_3]}{x_4 - x_1}$
x2	f2	$f[x_2, x_3] = \frac{f_3 - f_2}{x_3 - x_2}$	$f[x_2, x_3, x_4] = \frac{f[x_3, x_4] - f[x_2, x_3]}{x_4 - x_2}$	$f[x_1, x_2, x_3, x_4]$
x3	f3	$f[x_3, x_4] = \frac{f_4 - f_3}{x_4 - x_3}$		
x4	f4			

ALGORITMO INTERMEDIARIO
 1. n = length(S)
 2. i = 1 to n
 3. j = 1 to n
 4. M[i, j] = S[i]^(j-1)
 5. b[i, 1] = f(S[i])
 6. a = M^-1 * b

ALGORITMO
 - SIGNO JUNTOS: CUMPLIR O YA ESTÉN TODAS LAS VARIABLES (CONTINUA) OPERACIÓN A LA DEMANDA
 - SIGUIENTES: TANTO OMS SUGERENCIAS, WHO DEBE? OMS OMS
 - SUMAS: SUM = 0 (ANULAR) SUM = SUM + f(x) (SUMAR)
 - PROD: PROD = 1 (PROD = PROD * f(x))



- Chuleta 3:

Lagrange (n+1 puntos polinomio) = n+1 - 1 = n

Polinomio interpolados Lagrange $\rightarrow P(x) = \sum_{i=0}^n f(s_i) \cdot L_i(x)$

Polinomio base de Lagrange $L_i(x) = \prod_{j \neq i} \frac{x - s_j}{s_i - s_j}$

Puntos, soporte $\rightarrow L_j(x) \begin{cases} L_j(s_j) = 1 \\ L_j(s_i) = 0 \end{cases}$

Repeticiones, combinatoria $L_i(x) = \prod_{j \neq i} \frac{x - s_j}{s_i - s_j}$

- Definir pts de soporte y su $f(s_i)$
- Calcular polinomio de base Lagrange.
- Combinar polinomio interpolados $f(s_i) \cdot L_i(x) + f(s_2) \cdot L_2(x) + \dots$
- Simplificar polinomio
- Evaluar en pts de xado para valores aprox

Sist. lineal de ecuaciones $P(x) = a_0x^0 + a_1x^1 + a_2x^2 + \dots + a_nx^n$

Ej: $f(-1)=3, f(0)=2, f(2)=6$

$P(-1) = a_0 + a_1(-1) + a_2(-1)^2 = 3$

$P(0) = a_0 + 0 + 0 = 2$

$P(2) = a_0 + 2a_1 + 4a_2 = 6$

$$\begin{bmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 2 & 4 \end{bmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 6 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & -1 & 1 & | & 3 \\ 1 & 0 & 0 & | & 2 \\ 1 & 2 & 4 & | & 6 \end{pmatrix}$$

Ej Lagrange $f(-1)=3, f(0)=2, f(2)=6, n=2$

$L_0(x) = \frac{(x-s_2)(x-s_1)}{(s_0-s_2)(s_0-s_1)}$; $L_1(x) = \frac{(x-s_2)(x-s_0)}{(s_1-s_2)(s_1-s_0)}$; $L_2(x) = \frac{(x-s_1)(x-s_0)}{(s_2-s_1)(s_2-s_0)}$

Formula de Newton $P_n(x) = \sum_{i=0}^n f[s_0, s_1, \dots, s_i] \prod_{j=0}^{i-1} (x - s_j)$

$P_n(x) = P_{n-1}(x) + f[s_0, s_1, \dots, s_n] (x - s_0) \dots (x - s_{n-1})$

sin tabla

s_k	1	2	4	7
$f(s_k)$	-7	3	251	2543

Soporte $\{s_0\} P_0(x) = f(s_0) = -7$

Soporte $\{s_0, s_1\} P_1(x) = P_0(x) + f[s_0, s_1](x - s_0)$

$f[s_0, s_1] = \frac{3 - (-7)}{2 - 1} = 10$

$P_1(x) = -7 + 10(x - 1) = 10x - 17$

Soporte $\{s_0, s_1, s_2\} P_2(x) = P_1(x) + f[s_0, s_1, s_2](x - s_0)(x - s_1)$

$f[s_0, s_1, s_2] = \frac{251 - 23}{(4 - 1)(4 - 2)} = 38$

$P_2(x) = 10x - 17 + 38(x - 1)(x - 2) = 10x - 17 + 76x^2 - 176x + 76 = 76x^2 - 166x + 59$

Soporte $\{s_0, s_1, s_2, s_3\} P_3(x) = P_2(x) + f[s_0, s_1, s_2, s_3](x - s_0)(x - s_1)(x - s_2)$

$f[s_0, s_1, s_2, s_3] = \frac{2543 - 1747}{(7 - 1)(7 - 1)(7 - 4)} = 15$

$P_3(x) = 15x^3 - 67x^2 + 106x - 67$

Si queremos hacer calculamos $P_4(x)$.

Con tabla con sumatoria * si sea x

$s_0 = 0 \rightarrow f[s_0, s_1] = \frac{f(s_1) - f(s_0)}{s_1 - s_0}$

$s_1 = \frac{\pi}{4} \rightarrow f[s_1, s_2] = \frac{f(s_2) - f(s_1)}{s_2 - s_1}$

$s_2 = \frac{\pi}{2} \rightarrow f[s_2, s_3] = \frac{f(s_3) - f(s_2)}{s_3 - s_2}$

$s_3 = \frac{\pi}{6} \rightarrow f[s_3, s_4] = \frac{f(s_4) - f(s_3)}{s_4 - s_3}$

$f[s_0, s_1, s_2, s_3, s_4] = \frac{f[s_1, s_2, s_3, s_4] - f[s_0, s_1, s_2, s_3]}{s_4 - s_0}$

recursiva.

```

graph TD
    A[Inicio] --> B[n = n elementos]
    B --> C{i = 1}
    C --> D{i = n}
    D --> E[m(i) = f(i)]
    E --> F{i = i - 1}
    F --> G{f(i, i-1) = f(i)}
    G --> H{f(i, i-1, i-2) = f(i, i-1)}
    H --> I{f(i, i-1, i-2, i-3) = f(i, i-1, i-2)}
    I --> J{f(i, i-1, i-2, i-3, i-4) = f(i, i-1, i-2, i-3)}
    J --> K{f(i, i-1, i-2, i-3, i-4, i-5) = f(i, i-1, i-2, i-3, i-4)}
    K --> L{f(i, i-1, i-2, i-3, i-4, i-5, i-6) = f(i, i-1, i-2, i-3, i-4, i-5)}
    L --> M{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7) = f(i, i-1, i-2, i-3, i-4, i-5, i-6)}
    M --> N{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7)}
    N --> O{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8)}
    O --> P{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9)}
    P --> Q{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10)}
    Q --> R{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11)}
    R --> S{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12)}
    S --> T{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13)}
    T --> U{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14)}
    U --> V{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15)}
    V --> W{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16)}
    W --> X{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17)}
    X --> Y{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18)}
    Y --> Z{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19)}
    Z --> AA{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20)}
    AA --> AB{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21)}
    AB --> AC{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22)}
    AC --> AD{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23)}
    AD --> AE{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24)}
    AE --> AF{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25)}
    AF --> AG{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26)}
    AG --> AH{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27)}
    AH --> AI{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28)}
    AI --> AJ{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29)}
    AJ --> AK{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30)}
    AK --> AL{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31)}
    AL --> AM{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32)}
    AM --> AN{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33)}
    AN --> AO{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34)}
    AO --> AP{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35)}
    AP --> AQ{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36)}
    AQ --> AR{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37, i-38) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37)}
    AR --> AS{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37, i-38, i-39) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37, i-38)}
    AS --> AT{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37, i-38, i-39, i-40) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37, i-38, i-39)}
    AT --> AU{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37, i-38, i-39, i-40, i-41) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37, i-38, i-39, i-40)}
    AU --> AV{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37, i-38, i-39, i-40, i-41, i-42) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37, i-38, i-39, i-40, i-41)}
    AV --> AW{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37, i-38, i-39, i-40, i-41, i-42, i-43) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37, i-38, i-39, i-40, i-41, i-42)}
    AW --> AX{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37, i-38, i-39, i-40, i-41, i-42, i-43, i-44) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37, i-38, i-39, i-40, i-41, i-42, i-43)}
    AX --> AY{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37, i-38, i-39, i-40, i-41, i-42, i-43, i-44, i-45) = f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29, i-30, i-31, i-32, i-33, i-34, i-35, i-36, i-37, i-38, i-39, i-40, i-41, i-42, i-43, i-44)}
    AY --> AZ{f(i, i-1, i-2, i-3, i-4, i-5, i-6, i-7, i-8, i-9, i-10, i-11, i-12, i-13, i-14, i-15, i-16, i-17, i-18, i-19, i-20, i-21, i-22, i-23, i-24, i-25, i-26, i-27, i-28, i-29
```

$n = a, b$ numeric (break line) (print = "ingresó valor de n: ")
 Print(n) print printf("ese número vale de a, b: ")

Intercambio if

```

if (a > b) {
    aux = (b & a);
    print(aux);
} else {
    print(a);
}
    
```

FLAG (marca) (no se que es)

```

FLAG = 0;
if (a > b) {
    FLAG = 1;
}
    
```

while (mientras que, hasta que) (no sabemos cuánto veces se repite) (ver en pro. datos)

```

s = 0;
while (s < 10) {
    print(s);
    s = s + 1;
}
    
```

for (cuando se sabe cuántas veces se repite)

```

for (i = 0; i < 10; i++) {
    print(i);
}
    
```

Matrices y vectores $v = c(i)$

$v(i) = a_i$ numeric... $v(i) = i^2$

Raíz sqrt(x) con $x = a + \frac{1}{a}$

$x = \frac{1}{2} (x + \frac{1}{x})$

$x = \frac{1}{2} (x + \frac{1}{x})$

$x = \frac{1}{2} (x + \frac{1}{x})$

Binario

num = 10
 binario = (10)
 num = 10
 num = 10
 binario = (10)
 num = 10
 binario = (10)

Binario real

num = 10
 binario = (10)
 num = 10
 binario = (10)

while

```

while (n > 0) {
    print(n);
    n = n - 1;
}
    
```

for

```

for (i = 0; i < 10; i++) {
    print(i);
}
    
```

Suma de series

$S_n = \sum_{k=1}^n k$

$S_n = \frac{n(n+1)}{2}$

Factorial

$n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 1$

Flowcharts

Flowchart for swapping two numbers using a temporary variable.

Flowchart for calculating the sum of a series using a while loop.

Flowchart for calculating the sum of a series using a for loop.

Flowchart for calculating the factorial of a number using a while loop.

Flowchart for calculating the factorial of a number using a for loop.

- Chuleta 4:

organigrama = pseudocódigo

\odot - inicio ; \oplus - fin ; \circ - conectores
 \square - entrada ; \square - salida ; \square - operaciones
 \square - bucles ; \square - condiciones ; \downarrow - flujo.

Condiciones:
 If (condición) {
 sentencias
 }
 # no tiene condición
 If (condición) {
 sentencia 1
 } else { # "en cualquier otro caso"
 sentencia 2
 }
 If (condición 1) {
 sentencia 1
 } else if (condic. 2) {
 sentencia 2
 } else {
 sentencia 3
 }

Sumatorio: $sum = 0$; $sum = sum + \dots$
Productorio: $prod = 1$; $prod = prod * \dots$
 $\sum_{i=1}^n \prod_{i=1}^n \rightarrow \langle i=1, n \rangle \leftarrow \prod_{i=1}^n$

Inicializar vectores:
 V ← numeric(n) # número de n ceros
 V ← integer(n) # entero de n ceros
 V ← character(n) # n cadenas vacías
 V ← logical(n) # v. lógico de n False
 V ← c(n) # n elementos
 V ← c(5, 2, 3) # 3 elementos determinados
 V ← 1:10 # con valores del 1 al 10.

Inicializar matriz:
 M ← matrix(0, n, n) # 2 bucles for cantidades, usar dos variables #

Bucle While:
 while (cond.) {
 sentencia
 }

Binario:
 while (cond.) {
 sentencia
 }

== igual
 > mayor que
 < menor que
 != distinto
 >= mayor o igual
 <= menor o igual
 %/2 == 0 verifica si el n = es par (si el resto de dividir el n = entre 2 es 0).
 -Cero: se pone \emptyset para diferenciar de B. extra "0".

Suma de productos:
 $sum(i) = \sum_{j=2}^m \left(\sum_{k=0}^{j-1} \prod_{r=0}^{k-1} (s-r) \right) * (DF(i, m) / \text{Factorial})$
 $i = 1, 2, \dots, n$

Diagrama de flujo:
 INICIO
 In, m, DF, Factorial, s
 i = 1, n
 sum1 = 0
 j = 2, m
 sum2 = 0
 r = 0, j-1
 prod = 1
 r = 0, j-1
 r ≠ k? (si) prod = prod * (s-r)
 sum2 = sum2 + prod
 sum1 = sum1 + sum2
 sum[i] = sum1 * (DF(i, m) / Factorial)
 FIN

Formulas:
 $F(n, a) = \sum_{i=1}^n \sum_{j=2}^m \left[\prod_{k=1}^j (1+1+2k) \right]$
 k = 1, 2, ..., m
 $BEST_k = ECOL_{1,1} + \sum_{i=2}^m (ECOL_{1,i} * \prod_{j=1}^{i-1} (y_k - t_j))$
 k = 1, 2, ..., M

Polinomio interpolador: Datos: $f(-1) = 3$, $f(0) = 2$; $f(2) = 6$ $\left\{ P(x) \text{ de grado } \overbrace{2}^{\text{no ptes}} - 1 \right.$

- Resolviendo un s. lineal de ecuaciones.
 Polinomio interpolador: $P(x) = a_0 \cdot x^0 + a_1 \cdot x^1 + a_2 \cdot x^2 \dots \dots$ (igualamos a $f(x)$)

$S_0 \rightarrow P(-1) = a_0 \cdot (-1)^0 + a_1 \cdot (-1)^1 + a_2 \cdot (-1)^2 = a_0 \cdot 1 + a_1 \cdot (-1) + a_2 \cdot 1 = a_0 - a_1 + a_2 = 3$

$S_1 \rightarrow P(0) = a_0 \cdot (0)^0 + a_1 \cdot (0)^1 + a_2 \cdot (0)^2 = a_0 \cdot 1 + a_1 \cdot 0 + a_2 \cdot 0 = a_0 = 2$

$S_2 \rightarrow P(2) = a_0 \cdot (2)^0 + a_1 \cdot (2)^1 + a_2 \cdot (2)^2 = a_0 \cdot 1 + a_1 \cdot 2 + a_2 \cdot 4 = a_0 + 2a_1 + 4a_2 = 6$

\rightarrow Sistema de ecuaciones:

$$\begin{cases} \textcircled{1} P(-1) = a_0 - a_1 + a_2 = 3 \\ \textcircled{2} P(0) = a_0 = 2 \\ \textcircled{3} P(2) = a_0 + 2a_1 + 4a_2 = 6 \end{cases} \begin{cases} \textcircled{1} a_1 = a_0 + a_2 - 3; \xrightarrow{a_2=1} a_1 = 0 \\ \textcircled{3} 2 + 2 \cdot (a_0 + a_2 - 3) + 4a_2 = 6; a_2 = 1 \end{cases}$$

\rightarrow sustituimos en $P(x)$ es a_n , pero no x , quedando: $P(x) = x^2 + 2$

- Resolviendo por Lagrange: $P(x) = \sum_{i=0}^{n-1} f(s_i) \cdot L_i(x)$ $L_i(x) = \prod_{j \neq i} \frac{x - s_j}{s_i - s_j}$

n ($n = \text{coordenadas } x$) = 3
 i (desde 0 a $n-1$) = 0, 1, 2
 j (desde 0 a $n-1$) = 0, 1, 2

$s_0 = -1$; $f(s_0) = 3$
 $s_1 = 0$; $f(s_1) = 2$
 $s_2 = 2$; $f(s_2) = 6$

Iteración 1:
 $i=0$ $j \neq i$
 $j=1, 2$ $j \neq 0$
 $L_0(x) = \frac{(x-s_1) \cdot (x-s_2)}{(s_0-s_1) \cdot (s_0-s_2)} = \frac{(x-0) \cdot (x-2)}{(-1-0) \cdot (-1-2)} = \frac{x \cdot (x-2)}{(-1) \cdot (-3)} = L_0(x) = \frac{x^2 - 2x}{3}$

Iteración 2:
 $i=1$ $j \neq i$
 $j=0, 2$ $j \neq 1$
 $L_1(x) = \frac{(x-s_0) \cdot (x-s_2)}{(s_1-s_0) \cdot (s_1-s_2)} = \frac{(x-(-1)) \cdot (x-2)}{(0-(-1)) \cdot (0-2)} = \frac{(x+1) \cdot (x-2)}{1 \cdot (-2)} = \frac{x^2 - 2x + x - 2}{-2} = -\frac{x^2 - x - 2}{2}$

Iteración 3:
 $i=2$ $j \neq i$
 $j=0, 1$ $j \neq 2$
 $L_2(x) = \frac{(x-s_0) \cdot (x-s_1)}{(s_2-s_0) \cdot (s_2-s_1)} = \frac{(x-(-1)) \cdot (x-0)}{(2-(-1)) \cdot (2-0)} = \frac{(x+1) \cdot x}{(2+1) \cdot 2} = \frac{x^2 + x}{3 \cdot 2} = L_2(x) = \frac{x^2 + x}{6}$

$P(x) = f(s_0) \cdot L_0(x) + f(s_1) \cdot L_1(x) + f(s_2) \cdot L_2(x) = 3 \cdot \left(\frac{x^2 - 2x}{3}\right) + 2 \cdot \left(-\frac{x^2 - x - 2}{2}\right) + 6 \cdot \left(\frac{x^2 + x}{6}\right) = x^2 + 2$

- Resolviendo por Newton: $P(x) = \sum_{i=0}^{n-1} [f(s_0, s_1, \dots, s_i)] \cdot \prod_{j=0}^{i-1} (x - s_j)$ $n = n = \text{ptos/coordenadas}$

$P(x) = a_0 + a_1 \cdot (x - s_0) + a_2 \cdot (x - s_0) \cdot (x - s_1) + \dots + a_{n-1} \cdot (x - s_0) \cdot (x - s_1) \cdot \dots \cdot (x - s_{n-2})$

i	s_i	$f(s_i)$	$f[s_0, s_1]$	$f[s_0, s_1, s_2]$
$i=0$	$s_0 = -1$	$f[s_0] = f(s_0) = 3 = a_0$	$f[s_0, s_1] = \frac{f[s_1] - f[s_0]}{s_1 - s_0} = \frac{2 - 3}{0 - (-1)} = \frac{-1}{+1} = -1 = a_1$	$f[s_0, s_1, s_2] = \frac{f[s_1, s_2] - f[s_0, s_1]}{s_2 - s_0} = \frac{\frac{6-2}{2-0} - (-1)}{2 - (-1)} = \frac{\frac{4}{2} + 1}{2+1} = \frac{2+1}{2+1} = \frac{3}{3} = 1 = a_2$
$i=1$	$s_1 = 0$	$f[s_1] = f(s_1) = 2$	$f[s_1, s_2] = \frac{f[s_2] - f[s_1]}{s_2 - s_1} = \frac{6-2}{2-0} = \frac{4}{2} = 2$	0
$i=2$	$s_2 = 2$	$f[s_2] = f(s_2) = 6$	0	0

$P(x) = a_0 + a_1 \cdot (x - s_0) + a_2 \cdot (x - s_0) \cdot (x - s_1) = 3 + (-1) \cdot (x - (-1)) + 1 \cdot (x - (-1)) \cdot (x - 0) = 3 - x - 1 + (x+1) \cdot x = 3 - x - 1 + x^2 + x = x^2 + 2 = P(x) \rightarrow \text{Comprobado } \checkmark$

Conclusión:

Aquí van algunas recomendaciones.

Lo importante es que trabajes los ejercicios y hagas la chuleta tuya. De nada sirve copiar los ejemplos de las chuletas de otras personas si no lo entiendes o luego no encuentras lo que necesitas el día del examen.

Hazlo conciso, claro y ordenado. Escribe con letra pequeña para que te quepa todo pero que se entienda.

Y, por último, queremos animarte y decirte que se puede sacar, aunque parezca que no. Es algo nuevo y es normal que te parezca abrumador, pero tú puedes si lo trabajas y practicas mucho.

Videos recomendados:

- [Expresiones con sumatorio y productorios anidados](#)
- [Algoritmo con anidamiento de sumatorios y productorios](#)
- [Interpolación de Lagrange. Ejemplo Resuelto](#)
- [Interpolación de Newton](#)