

Today's class:

- **Second Assignment** (same structure as the first one, read engi231020p1c4.pdf from lmvicente.com/engi2310/engi2310.htm)
 - Cover page.
 - Table of contents.
 - Introduction.
 - Methodology and Results
 - Phase 1:
 - Analysis, inputs, outputs.
 - Pseudocode.
 - FlowChart.
 - Phase 2: Implementation in c++.
 - Phase 3:
 - Testing, using debugger.
 - Show screen output.
 - References.
 - Appendix (show the complete code here).

- Examples of projects (A projects):

COMPONENTES ELECTRÓNICOS

En una empresa de montaje de aparatos electrónicos se desea un programa que calcule ciertas características de los productos de la empresa. La empresa monta 10 tipos diferentes de aparatos electrónicos, cada uno de los cuales incluye un número de componentes diferentes. En total la empresa utiliza 100 componentes electrónicos diferentes. Suponemos que cada aparato sólo tiene **una unidad** de cada uno de los componentes que lo forman.

Debes construir un programa que realice las siguientes funciones:

- 1. En primer lugar se procederá a la lectura de los datos que indican cuáles son los componentes de cada aparato electrónico. El usuario introducirá los datos relativos a los 10 aparatos, indicando primero el número de componentes que tiene.
- 2. A continuación, el programa deberá leer el número de unidades que se montan mensualmente de cada aparato.
- 3. Para realizar los pedidos correspondientes, se deben contabilizar el número de componentes que se necesitan cada mes y escribirlos en la pantalla.
- 4. El programa deberá identificar el componente eléctrico que más se utiliza.

Apartado 1)

```
Introduce los componentes de cada aparato
Número de componentes del aparato 0: 3
  Componente 1: 1
  Componente 2: 3
  Componente 3: 99
Número de componentes del aparato 1: 20
  Componente 1: 1
  Componente 2: 3
  ...
  Componente 20: 99
...
Número de componentes del aparato 9: 1
  Componente 1: 65
```

Apartado 2)

```
Introduce el número de aparatos montados mensualmente:
Aparato 0: 100
Aparato 2: 1000
...
Aparato 9: 300
```

Apartado 3)

Número de unidades de cada componente utilizados mensualmente:

Componente 0: 100

...

Componente 99: 100

Apartado 4)

El componente electrónico 7 es el más utilizado, se necesitan 10000 unidades.

HÁBITOS DE CONSUMO: BEBIDAS

El Gobierno Vasco quiere hacer un estudio de hábitos de consumo de alcohol y bebida entre la juventud vasca. Para ello, ha escogido 100 pueblos donde llevará a cabo la consulta. El ayuntamiento de cada uno de estos pueblos recogerá el número de litros consumidos durante el pasado año de los 8 tipos de bebidas nombrados a continuación:

Tipos De Bebidas

0. Naturales: agua, leche, zumos, mostos, infusiones, ...
1. Gaseosos: Refrescos
2. Vinos
3. Cervezas
4. Whisky
5. Licores
6. Energéticos: Aquarius, Red Bull,...
7. Combinados: Cubatas, GinTonics,..

Mediante un programa en C, se te pide que realices un estudio que estará compuesto por los siguientes pasos:

1. Recoge y almacena la información, para cada pueblo, sobre los litros consumidos de los tipos de bebidas mencionados. Recuerda que 100 pueblos aportarán información a tu estudio. No obstante, la información no te la tienen por qué dar ordenada por tipo de bebida o número de litros y puede que de algún tipo de bebida no se tenga información.

Así, será preciso introducir pares (tipo de bebida, litros) y para indicar que no hay más datos de un pueblo, bastará con introducir como tipo de bebida el valor -1. Ver ejemplo para cada uno de los apartados al final del enunciado.

2. Determina el “tipo de bebida” más consumida entre todos los pueblos; es decir, el tipo de bebida del que más litros se hayan bebido entre todos los pueblos.
3. Recoge y almacena cuáles de las clases de bebidas tienen alcohol. Para ello, te dan únicamente la lista de los “tipos de bebidas” que contienen alcohol acabada con un -1 para indicar que no hay más tipos de bebida que tengan alcohol. Por ejemplo: 7 2 5 4 3 -1

- a. De los “tipos de bebidas” con alcohol, ¿cuál es el más consumido?
- b. ¿En qué pueblo se bebe más alcohol en total?

Ejemplo: (En letra normal lo que el programa imprime y en negrita lo que el usuario teclea)

```
Para comenzar el estudio recogemos los datos por pueblos:
Por cada pueblo: Introduce pares (tipoBebida, litros). Para
acabar, el primer componente del último par tiene que tener
valor -1.
Pueblo 0:
  0 8
  4 7
  2 0
  3 50
  1 10
  -1 0
Pueblo 2:
  0 18
  1 15
  4 0
  2 6
  3 36
  -1 32
Pueblo 3:
  5 16
  ...
Pueblo 99:
  3 5
  1 12
  -1 9
```

El tipo de bebida más consumida es la: 3
(Suponiendo que lo que más se ha bebido en litros sean cervezas)

Ahora necesitamos que enumeres SÓLO los tipos de bebida CON alcohol, -1 para acabar: 7 2 5 4 3 -1

El tipo de bebida con alcohol más consumida es la: 3
(Suponiendo que así sea en litros entre: "combinados", "vinos", "licores", "whiskies" y "cervezas". En general, el "tipo de bebida" y "el tipo de bebida con alcohol" más consumidas no tiene por qué coincidir).

El pueblo con mayor consumo de alcohol: X
(Suponiendo que entre "combinados", "vinos", "licores", "whiskies" y "cervezas", X sea la población que más litros haya vendido).

EMISORA DE RADIO

Una emisora de radio quiere hacer un concurso en el que proponen a sus oyentes los títulos de 10 canciones, a las que ha numerado del 0 al 9. Cada participante debe llamar para dar los números de 3 títulos por orden de preferencia decreciente.

Se le dan 3 puntos a la canción que el oyente nombra en 1er lugar, 2 puntos a la que indica en 2º lugar y 1 punto a la 3ª.

Se pide hacer un programa que realice las siguientes tareas:

7. Leer y almacenar los votos emitidos por cada oyente, suponemos que como máximo pueden concursar 100 oyentes. Y a medida que llaman los concursantes se les asigna un número de forma correlativa.

La entrada de datos se realizará en forma de tríos, con los números de las canciones que vota cada oyente. Para finalizar se introducirá el valor -1 en el primer dato del trio, sin importar el valor de los otros dos datos.

Ejemplo:

Oyente 0: 6 2 1
Oyente 1: 3 7 5
Oyente 2: 3 0 1
Oyente 3: 0 5 2
Oyente 4: -1 0 0

8. Calcular los votos obtenidos por cada canción e indicar cuáles son la 1° y 2° canción más votada.

Según el ejemplo anterior:

Canción 0: 5 votos
Canción 1: 2 votos
Canción 2: 3 votos
Canción 3: 6 votos
Canción 4: 0 votos
Canción 5: 3 votos
Canción 6: 3 votos
Canción 7: 2 votos
Canción 8: 0 votos
Canción 9: 0 votos
1ª canción: 3
2ª canción: 0

TIENDA

Una pequeña tienda guarda información de las ventas de sus artículos a lo largo del año 2009. La tienda dispone de 100 artículos diferentes.

Se pide escribir un programa que realice las siguientes tareas:

1. Leer las ventas de los artículos (en unidades) a lo largo de los 12 meses del año.
2. Leer la lista de precios por artículo y obtener el mes que más ganancias ha tenido.
3. Por cada artículo leer el total de ganancias del periodo comprendido entre los meses de julio y diciembre del año 2008. A continuación comparar estos datos con los del mismo periodo del 2009 (leídos en el apartado 1). Indicar cuántos artículos han superado las ganancias en el año 2009 y cuales son estos artículos.

Ejemplo de ejecución:

☞ Para el apartado 1)

Introduce las ventas de los artículos de la siguiente forma (en negrita los datos tecleados por el usuario):

```
Artículo, mes , ventas : 1 5 800 (son unidades)
Artículo, mes , ventas : 0 4 700
Artículo, mes , ventas : 4 5 100
Artículo, mes , ventas : 6 5 800
Artículo, mes , ventas : 6 11 800
.....
.....
Artículo, mes , ventas : 1 0 800
Artículo, mes , ventas : -1 0 0
```

CENTRAL ELÉCTRICA

Una central eléctrica desea realizar un estudio sobre la cantidad de energía que suministra diariamente. Para ello, el programa informático que se ha de desarrollar deberá permitir recoger los datos de las potencias típicas, en megavatios, suministradas diariamente durante un período máximo de 52 semanas (1 año).

1. La entrada de datos se deberá programar de forma que, para cada semana, se introduzcan las potencias suministradas durante los 7 días.

Cuando se desee finalizar la entrada de datos (por ejemplo, en el caso de realizar el estudio para un periodo inferior a las 52 semanas) se deberá introducir un -1 , como dato de la potencia, en el primer día de la semana.

Ejemplo de entrada de datos:

```
Semana 0
Potencia suministrada del día 0: 207
Potencia suministrada del día 1: 301
Potencia suministrada del día 2: 222
Potencia suministrada del día 3: 302
Potencia suministrada del día 4: 22
Potencia suministrada del día 5: 167
Potencia suministrada del día 6: 125
```

```
Semana 1
Potencia suministrada del día 0: 367
Potencia suministrada del día 1: 60
Potencia suministrada del día 2: 120
Potencia suministrada del día 3: 111
Potencia suministrada del día 4: 301
Potencia suministrada del día 5: 400
Potencia suministrada del día 6: 434
```

```
Semana 4
Potencia suministrada del día 0: -1
```

LADRÓN ELCA COMAYOR

El señor Elca Comayor ha decidido hacer un programa para ver como va su negocio. El negocio de nuestro amigo es un tanto curioso, pues se dedica al mundo del robo y está sumamente interesado en desarrollar un programa que le ayude a llevar sus cuentas.

Él quiere guardar los datos de las últimas 8 semanas. Hay que tener en cuenta que él puede trabajar todos los días de la semana, pues no cierra nunca su negocio. De esta manera, a medida que va haciendo las cuentas introduce tres datos en el programa: la semana (del 0 al 7), el día (del 0 al 6) y la cantidad que ha robado ese día. Cuando introduzca un -1 en el lugar que le corresponde a la semana el programa entenderá que la entrada de datos ha finalizado.

Ejemplo: (en negrita los datos tecleados por el usuario)

```
Introduce los datos (semana, día, cantidad): 3    5    300
Introduce los datos (semana, día, cantidad): 0    1    50
Introduce los datos (semana, día, cantidad): 5    6    179
Introduce los datos (semana, día, cantidad): 1    2    22
Introduce los datos (semana, día, cantidad): -1   0    0
```

Además de estos datos, nuestro amigo Elca Comayor introduce los datos de sus gastos de las últimas 8 semanas, esto es, cuánto ha gastado cada una de las semanas.

Ejemplo:

```
Dato de la semana 0: 77
Dato de la semana 1: 60
.....
.....
Dato de la semana 7: 99
```

Tras introducir todos estos datos, nuestro amigo el empresario quiere conocer los siguientes datos:

1. Cuánto ha robado cada semana y cuál ha sido la semana en la que más ha robado.
2. Cuánto ha robado por cada uno de los días de la semana, esto es, cuánto los lunes, martes, ... y cuál es el día de la semana que más roba, para poder dedicarle más tiempo a su negocio ese día.
3. Cuánto ha ganado cada semana y el total ganado en las 8 semanas, teniendo en cuenta los gastos.
- 4.Cuál ha sido el primer día en el que no ha cometido ningún robo.

Además, él sabe que si la policía le detiene, recibirá el castigo dependiendo de lo que ha robado:

Menos de 5.000 € → Realizará obras en la comunidad
 Menos de 10.000€ → 3 meses en prisión
 Menos de 20.000€ → 6 meses en prisión
 Más de 20.000€ → Un año en prisión

A nuestro amigo Elca Comayor no le agrada nada la idea de tener que ir a prisión, por lo que le interesan mucho las consecuencias de su negocio. Por lo tanto, haz que el programa le diga a nuestro amigo cuál es la sanción que se le aplicará si le detienen.

EJERCICIO 8.11: TABLA DE LOGARITMOS CON FORMATOS ADECUADOS POR COLUMNAS

Se pide realizar un programa que calcule los logaritmos decimal y neperiano de los números comprendidos entre 1 y 100, con intervalos de 5 en 5. Para calcular los logaritmos se empleará la librería matemática *math.h*. La función que calcula el logaritmo decimal es *log10(d)* siendo *d* un número cualquiera, y la que calcula el logaritmo neperiano es *log(d)*.

La salida por pantalla que debe dar este programa **deberá ser exactamente** la siguiente:

```
Este programa calcula el logaritmo decimal
y el logaritmo neperiano de los numeros entre el
uno y el 100, de 5 en 5.
  numero          log          ln
-----
    1          0.000000      0.000000
    5          0.698970      1.609438
   10          1.000000      2.302585
   15          1.176091      2.708050
   20          1.301030      2.995732
   25          1.397940      3.218876
   30          1.477121      3.401197
   35          1.544068      3.555348
   40          1.602060      3.688879
   45          1.653213      3.806662
   50          1.698970      3.912023
   55          1.740363      4.007333
   60          1.778151      4.094345
   65          1.812913      4.174387
   70          1.845098      4.248495
   75          1.875061      4.317488
   80          1.903090      4.382027
   85          1.929419      4.442651
   90          1.954243      4.499810
   95          1.977724      4.553877
  100          2.000000      4.605170
```

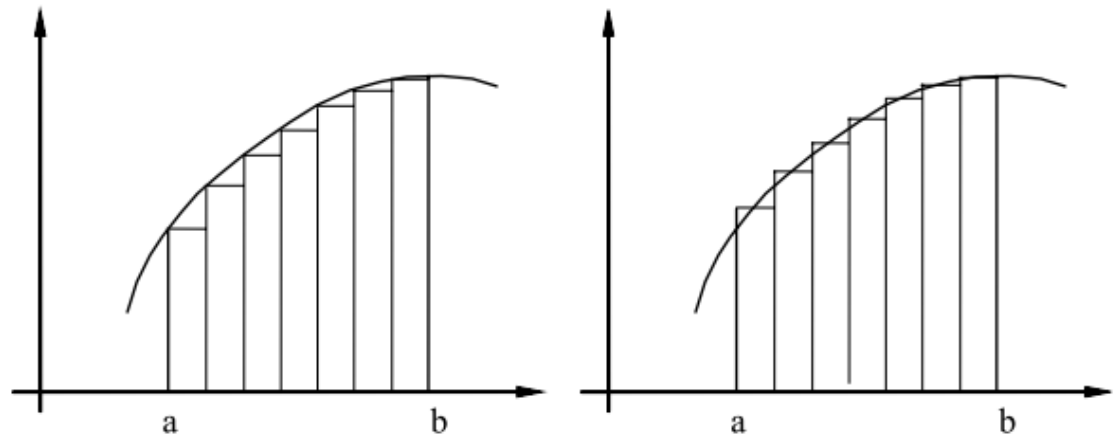
Se pondrá atención especial en la alineación de las columnas, a los encabezamientos con su subrayado y a los puntos decimales. El nombre del programa será *logarit.c*

EJERCICIO 8.7: INTEGRACIÓN NUMÉRICA DE UNA FUNCIÓN

La integral definida entre los puntos a y b de una función continua y acotada $f(x)$ representa el área comprendida debajo de esa función. En ocasiones es necesario calcular integrales (áreas) de modo numérico, es decir, sin conocer la integral de la función $f(x)$. Existen varios posibles métodos para calcular esta área. Quizás el más sencillo sea sustituir el área por un conjunto de n rectángulos elementales de base $h=(b-a)/n$ y altura $f(a+ih)$, $i=0, 1, 2, \dots, n-1$. El área sería:

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1} f(a+ih)h \quad (1)$$

La representación gráfica de esta forma de aproximar la integral se presenta en la figura siguiente (parte izquierda). Resulta que si n se hace muy grande (h muy pequeño) el error será pequeño.



Otro método algo más complicado (representado en la figura de la derecha) consiste en sustituir el área por un conjunto de n trapecios elementales de base h y lados $f(a+ih)$ y $f(a+(i+1)h)$, o lo que es lo mismo, por n rectángulos de altura $(f(a+ih)+f(a+(i+1)h))/2$. Es obvio que con este segundo método los errores van a ser más pequeños. En este caso, si llamamos f_i a $f(a+ih)$, la fórmula resulta ser:

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1} (f_i + f_{i+1})h/2 = \frac{f_0 + f_n}{2}h + \sum_{i=1}^{n-1} f_i h \quad (2)$$

A la vista de estos métodos, se pide realizar un programa que calcule el área de la función $f(x)=x^3+x^2-5x+3$ entre 0 y 3, **por el segundo de los métodos** explicados. El programa deberá pedir el valor de n . Llama a este programa *integra.c*

- Ejemplo de A project, pero haciendo tus propias modificaciones.

Una compañía le paga a sus vendedores una cantidad fija semanal de \$320.00, más un % de comisión que depende de las ventas realizadas por el vendedor esa semana, como sigue:

ventas	% comisión
\$1,000 - < \$3,000	5%
\$3,000 - < \$5,000	10%
\$5,000 - < \$8,000	12.5%
\$8,000 -	15%

Escriba un programa que genere un reporte del vendedor para la semana, que siga el formato que se presenta a continuación:

NOMINA DEL VENDEDOR PARA ESTA SEMANA

Ventas realizadas \$xxx.xx
 % de comisión que le aplica: xx.x %
 Cantidad fija semanal a pagarle: \$xxx.xx
 Cantidad a pagarle por comisión: \$xxx.xx
 Total a pagarle esta semana: \$xxx.xx

- Ejemplos de B projects

EJERCICIO #1

Se quiere solicitar al usuario 10 números cualesquiera, y se quiere determinar la suma y la multiplicación de los 10 números entrados.

Para solucionar este problema se necesita un ciclo que se ejecute 10 veces (para recibir los 10 valores). Dentro de este ciclo se ejecutan las siguientes acciones:

- Se pide un número
- Se recibe (con cin).
- Si hay algo que validar, se valida el dato entrado (en este caso el dato de entrada no se valida porque el problema dice que se soliciten 10 números cualesquiera por lo que se aceptan valores negativos, cero y positivos)
- Se acumula el dato en una variable que mantenga registro de la suma de valores entrados hasta el momento
- Se multiplica el dato por los anteriores en una variable que mantenga un registro de las multiplicaciones de números entrados hasta el momento
- Se repite el ciclo hasta que se hace 10 veces

```
/*MUESTRA DE LA SALIDA EN PANTALLA DEL PROGRAMA ANTERIOR
Entre el numero 1 a considerar en la suma y en la multiplicacion: 2
Entre el numero 2 a considerar en la suma y en la multiplicacion: 3.5
Entre el numero 3 a considerar en la suma y en la multiplicacion: 4
Entre el numero 4 a considerar en la suma y en la multiplicacion: 5.5
Entre el numero 5 a considerar en la suma y en la multiplicacion: 2.8
Entre el numero 6 a considerar en la suma y en la multiplicacion: 3.3
Entre el numero 7 a considerar en la suma y en la multiplicacion: 2
Entre el numero 8 a considerar en la suma y en la multiplicacion: 2
Entre el numero 9 a considerar en la suma y en la multiplicacion: 1
Entre el numero 10 a considerar en la suma y en la multiplicacion: 2

La suma de los 10 valores entrados es 28.1

La multiplicacion de los 10 valores entrados es 11383.7

Press any key to continue . . .
*/
```

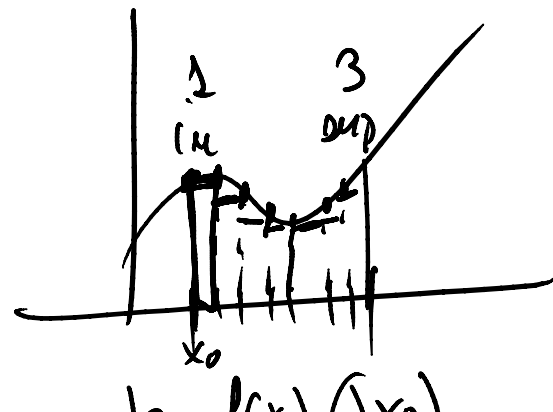
Ejemplo #2

```
// Un profesor quiere sacar un promedio de la clase.  
// Se quiere solicitar al profesor la nota final de  
// cada estudiante e imprimir el promedio de todas.  
// La clase tiene específicamente 10 estudiantes!!!
```

Entre nota del estudiante 1: 90
 Entre nota del estudiante 2: 800
 Nota entrada invalida. Re-entre.
 Entre nota del estudiante 2: 88
 Entre nota del estudiante 3: 78
 Entre nota del estudiante 4: 67
 Entre nota del estudiante 5: 99
 Entre nota del estudiante 6: 77
 Entre nota del estudiante 7: 66
 Entre nota del estudiante 8: 90
 Entre nota del estudiante 9: 56
 Entre nota del estudiante 10: 77

El promedio de la clase es 78.8
 Press any key to continue

- Programa que integra una funcion que entra el usuario.
- **Analisis:**
 - Se supone que la funcion es una polinómica, y el usuario va a entrar los coeficientes y cuando entre -99 se termina.
 - Despues el usuario entra el intervalo a integrar.
 - Despues el usuario entra el numero de divisiones del intervalo (precision de la integral).
 - Hacer los calculos y presentar.
- **Inputs:** coeficientes, intervalo inferior, intervalo superior, numero de divisiones.
- **Output:** la integral.
- **Pseudocodigo**
 - Definir variables, coeficientes, intervalo inf intervalo sup, numero de divisiones, resultado.
 - Mensaje al usuario para que entre los coeficientes y entre -99 para terminar.
 - Capturar los coeficientes.
 - Mensaje para que entre el intervalo.
 - Capturar intervalo.



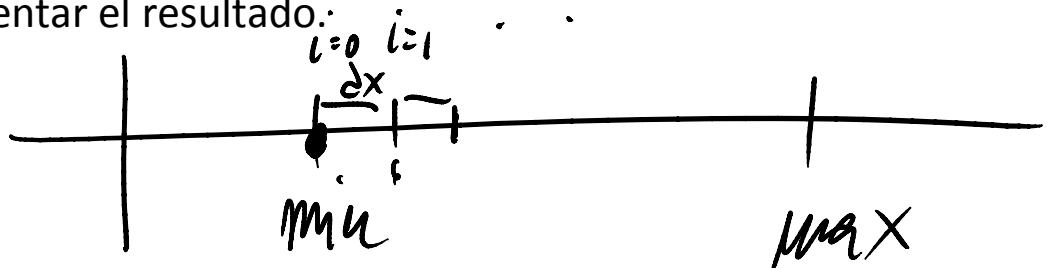
- Mensaje para que entre el intervalo.
- Capturar intervalo.
- Mensaje para que entre el numero de divisiones.
- Capturar el numero de divisiones.
- Hacer un bucle (for) y calcular el valor de la funcion para cada uno de los intervalos en que hemos dividido.
- Hacer un running sum dentro de for.
- Despues del for multiplicar por la anchura de los intervalitos.
- Presentar el resultado:

$$da = f(x_0) \cdot dx_0$$

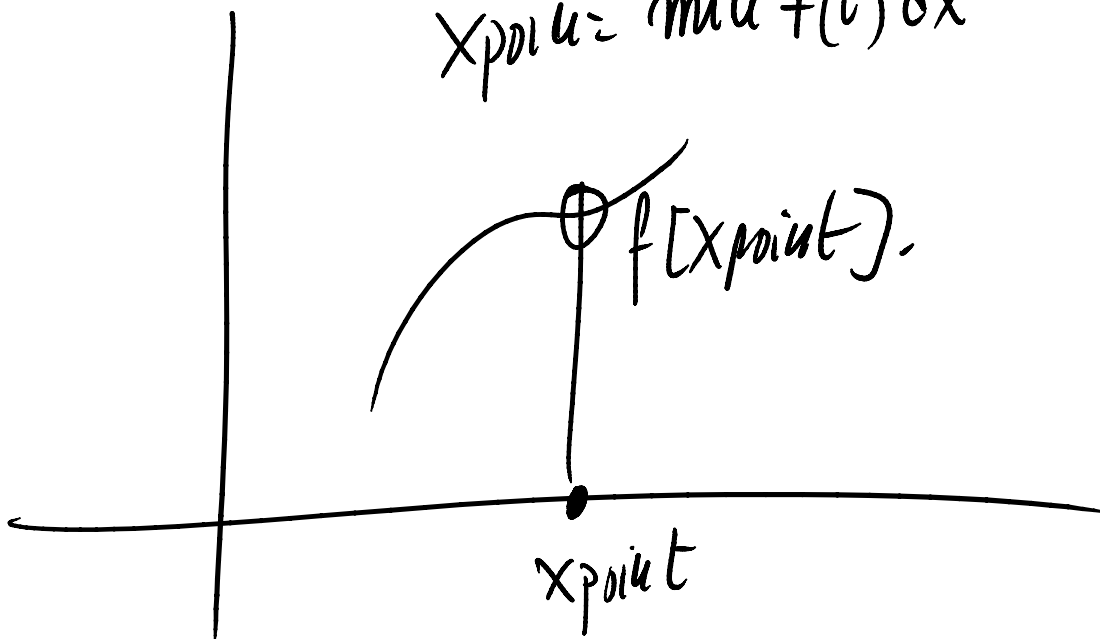
$$dx_0 = \frac{sup - inf}{n}$$

$$I = da_0 + da_1 + da_2 + \dots$$

$$= (f(x_0) + \dots + f(x_n)) dx$$

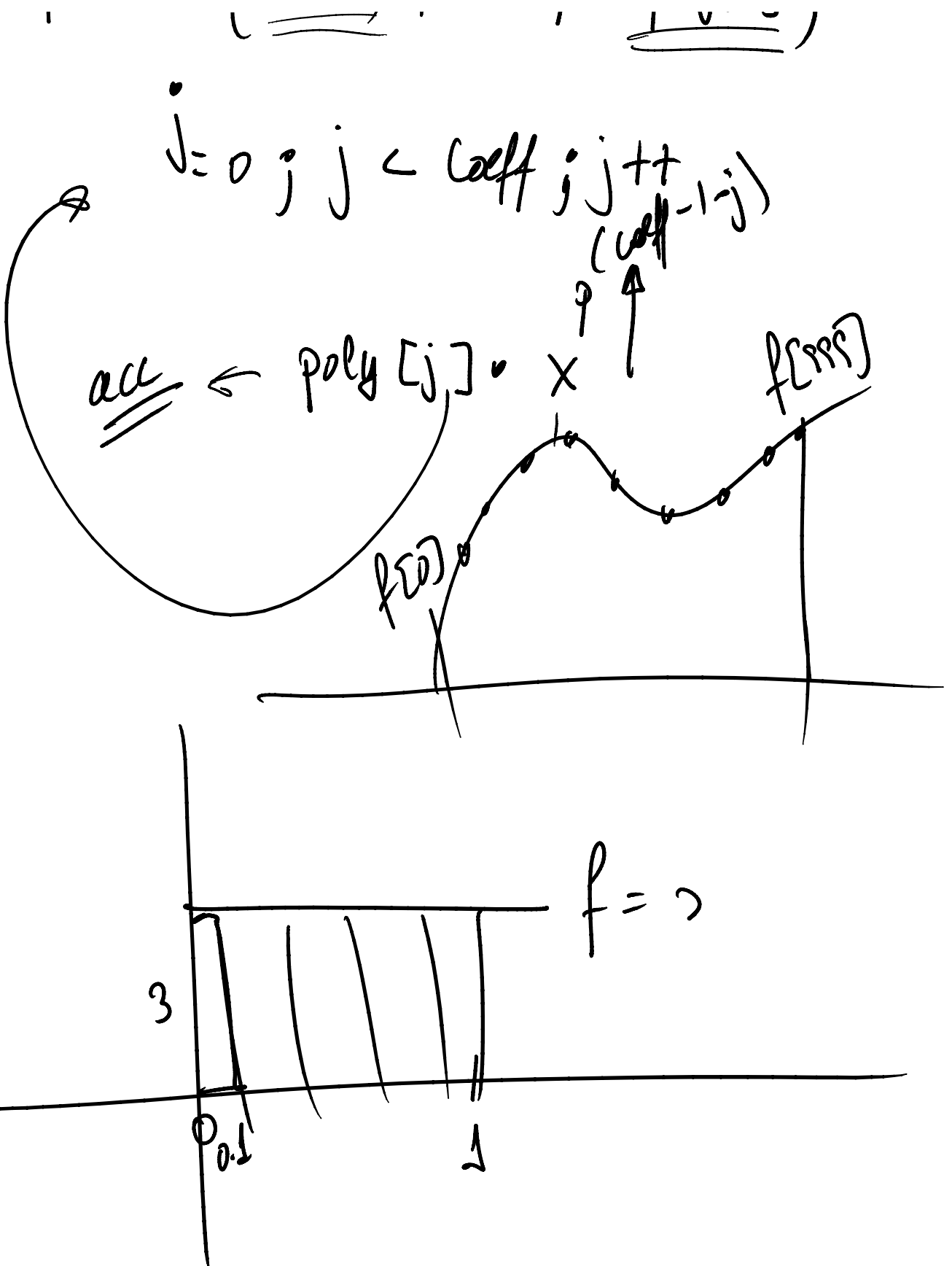


$$x_{point} = min + (i) dx$$



$$f(x_p) = \underbrace{poly[0]} x_p^2 + poly[1] x_p^1 + \underbrace{poly[2]} x_p^0$$

coeff - 1



El código es el siguiente:

```

1  #include <iostream>
2  #include <math.h>
3  using namespace std;
4
5  const int NUMCOEFS = 10;
6  const int MAXN = 1000;
7
8  int main() {
9      //Definition of variables used
10     double poly[NUMCOEFS];
11     double f[MAXN],xpoint,dx;
12     double min,max,n,result=0;
13     int i=0,coeffNum;
14     //while(1){
15
16     //user Message input data
17     cout << "Por favor entre los coeficientes de la funcion a ";
18     cout << "integrar de mayor a menor, entre -99 para acabar:";
19     i=0;
20     do{
21         cin >> poly[i];
22         i++;
23     }while((poly[i-1] != -99) && (i < NUMCOEFS) );
24
25     coeffNum = i-1;
26
27     //User Message input interval
28     cout << "Por favor entre el intervalo de integracion, desde: ";
29     cin >> min;
30     cout << ", hasta: ";
31     cin >> max;
32
33     //User message input number of division
34     cout << "Entre el numero de divisiones para el intervalo: ";
35     cin >> n;
36
37     dx = (max-min)/n;
38
39     //Loop to calculate function points
40     for(i=0;i<n;i++){
41         xpoint = min + i*dx;
42         f[i]=0;
43         //Now we calculate the function at xpoint
44         for(int j=0;j<(coeffNum);j++){
45             f[i]= f[i] + poly[j] * pow(xpoint,(coeffNum-1-j));
46         }
47         result = result + f[i]*dx;
48     }
49     cout << "El area es: " << result << endl;
50     //}
51     return 0;
52 }

```

- Este ejemplo es bastante difícil, ya que como el número de coeficientes no se sabe a priori hubo que crear un bucle "for"

- (línea 44) anidado dentro del primer for (línea 40) para calcular el valor de la función en cada punto (xpoint).
- El primer bucle for (línea 40) va calculando el valor de la función en cada punto (xpoint o la altura del mini rectángulo). Lo multiplica por dx (o la base del mini rectángulo) para calcular el área y acumularlo en result (línea 47).
 - El segundo bucle, calcula el valor de la función en cada punto. Como sabemos el número de coeficientes que tiene la función polinómica, podemos calcular cada término según la fórmula de la línea 45, que es bastante complicado.

Cuando se compila este es un ejemplo de salida:

```

C:\!Vicente\MyDest7\Desktop\Project1\Project1.exe
Por favor entre los coeficientes de la funcion a integrar de mayor a menor, entr
e -99 para acabar:1 2 3 4 5 6 -99
Por favor entre el intervalo de integracion, desde: 0
 hasta: 1
Entre el numero de divisiones para el intervalo: 1000
El area es: 11.1425

-----
Process exited with return value 0
Press any key to continue . . .

```

En este caso la función es: $f(x)=1*x^5+2x^4+3x^3+4x^2+5x+6$.

El intervalo de integración es $[0,1]$

El número de divisiones es 1000.

El resultado del área es 11.1425.

Podemos verificar ese resultado integrando la función analíticamente y dándole valores a la integral en el intervalo:

$\text{Int}f(x)=1/6x^6+2/5x^5+3/4x^4+4/3x^3+5/2x^2+6x$ en el intervalo $0,$

1 tenemos

$$\text{Intf}(1) - \text{intf}(0) = (1/6 + 2/5 + 3/4 + 4/3 + 5/2 + 6) - (0 + 0 + 0 + 0 + 0 + 0) = 11.5$$

El resultado del programa se acerca bastante!!!!