Explicar detalladamente el método de inserción binaria utilizando para ello el siguiente arreglo inicial

46	*57*	*14*	*44*	*96*	*20*	*8*	*70*

Completar en Modula-2 el código del siguiente programa para que realice el algoritmo de inserción binaria y realizar el análisis del programa, calculando el número de comparaciones y movimientos.

```
PROCEDURE insercion_binaria;
  VAR I,j,m,L,R :   INTEGER;
  x :           INTEGER
BEGIN
  FOR i:=2 TO n DO
  x:=a[I];
  L:= … ; R:= … ;
  WHILE … DO
    m:= … ;
    IF  …  THEN  L:=m+1  ELSE  R:=m
END;
  END
  FOR j:=i TO R+1 BY -1 DO … END;
    a[R]:=x;
  END;
END;
```

Fórmulas:

$$\int \ln x\,dx = x(\ln x - 1),\quad H_n = 1 + 1/2 + 1/3 + 1/4 + \cdots + 1/n = \ln n + g + \frac{1}{2n} + \cdots$$

Consideremos que tenemos una secuencia ordenada, y se le añade al final un nuevo elemento. Para ordenar la nueva secuencia se busca la posición final de dicho elemento dentro de la secuencia. Una vez localizada, se desplazan todos los elementos comprendidos entre esa posición y la penúltima hacia su derecha, colocando el nuevo elemento en el hueco abierto. Hasta ahora se ha actuado exactamente igual que en inserción directa. Lo que diferencia a la inserción binaria de la inserción directa es el método que sigue para encontrar la posición final. En la inserción directa, la posición del nuevo elemento se encontraba comparándolo con los elementos anteriores a él, que iban siendo desplazados de forma automática. En la inserción binaria primero se encuentra la posición, mediante una técnica de bipartición, esto es, dada la secuencia *1…N*, comparamos el nuevo elemento con el elemento medio *m* de la secuencia, si es mayor se toma como nueva secuencia de comparación la *m+1…N*, y si es menor, con la secuencia *1…m*. La ventaja de este método respecto al de inserción directa es que se disminuye el número de comparaciones, aunque el número de movimientos se mantiene del mismo orden.

2º Elemento	Posición	46	57	14	44	96	20	8	70
	Desplazamiento	46	57	14	44	96	20	8	70
3º Elemento	Posición	46	57	14	44	96	20	8	70
	Desplazamiento	14	46	57	44	96	20	8	70
4º Elemento	Posición	14	46	57	44	96	20	8	70
	Desplazamiento	14	44	46	57	96	20	8	70
5º Elemento	Posición	14	44	46	57	96	20	8	70
	Desplazamiento	14	44	46	57	96	20	8	70
6º Elemento	Posición	14	44	46	57	96	20	8	70
	Desplazamiento	14	20	44	46	57	96	8	70
7º Elemento	Posición	14	20	44	46	57	96	8	70
	Desplazamiento	8	14	20	44	46	57	96	70
8º Elemento	Posición	8	14	20	44	46	57	96	70
	Desplazamiento	8	14	20	44	46	57	70	97

El algoritmo queda escrito de la siguiente forma:

```
PROCEDURE Insercion_Binaria;
VAR i,j,m,L,R:INTEGER;
BEGIN
    FOR i:=2 TO n DO
        x:=a[i]; L:=1; R:=i;
        WHILE L < R DO
            m:=(L + R) DIV 2;
            IF a[m] <= x THEN L:=m + 1 ELSE R:=m END;
        END;
        FOR j:=i TO R + 1 BY -1 DO a[j] := a[j - 1] END;
        a[R] := x;
    END;
END Insercion_Binaria;
```

Comparaciones:

El número de comparaciones es igual en todos los casos, independientemente de que la secuencia esté en orden correcto o inverso. Para un elemento *k* cualesquiera, el número de comparaciones necesarias es $log_2\, k$, por tanto el número total de comparaciones vendrá dado como

$$C = \sum_{i=2}^{n} \log_2 i = \sum_{i=1}^{n} \log_2 i$$

Si aproximamos el sumatorio por la integral, se tiene

$$C=\int_1^n \log_2 x dx=\int_1^n \frac{\ln x}{\ln 2}dx=\left[\frac{x(\ln x-1)}{\ln 2}\right]_1^n=\frac{n(\ln n-1)+1}{\ln 2}\approx\frac{n(\ln n)}{\ln 2}$$

Movimientos:

El número de movimientos se calcula de forma análoga al método de inserción directa

Índice	Mejor Caso (ya ordenada)	Peor Caso (orden invertido)
2	2	3
3	2	4
4	2	5
…	…	…
k	2	k+1

Mínimo

$$\sum_{i=2}^{n} 2=2(n-1)$$

Máximo

$$\sum_{i=2}^{n}(i+1)=\frac{3+(n+1)}{2}(n-1)=\frac{n^2+3n-4}{2}$$

Promedio

$$\sum_{i=2}^{n}\left(\frac{i}{2}+1\right)=\frac{4+(n+2)}{4}(n-1)=\frac{n^2+5n-6}{4}$$