

Capítulo 7

Administración de la calidad de software

Alumno: Juan Duarte
C.I. N°: 4.653.346

EJERCICIOS

Tema I: Modelo De Amplificación De Defectos

Ejercicio 1

Utilizando el modelo de amplificación de defectos, resolver el siguiente ejercicio:

- a-) Se generan 16 errores en la 1era etapa de desarrollo de una aplicación. Se detecta el 10% de los errores.
- b-) De los errores de la etapa anterior, el 30% pasa inadvertido, y los demás errores se amplifican de acuerdo a un factor $x=2,5$. El porcentaje de detección es del 30%.
- c-) El 80% de los errores de la etapa anterior pasan inadvertidos, y el 20% es amplificado de acuerdo a un factor $x=3$. Se generan 30 nuevos errores, y el porcentaje de detección de esta etapa es del 30%.
- d-) En las 3 últimas etapas no se amplifican, ni se generan nuevos errores, teniendo un % de detección del 50,60 y 70 %.

0	10%	15	5	30%	21	17	30%	42
0			$10 \times 2,5 = 25$			$4 \times 3 = 12$		
16			0			30		

42	50%	21	21	60%	9	9	70%	3
0			0			0		
0			0			0		

Finalmente, se tienen 3 errores al terminar el proceso.

Ejercicio 2

Utilizando el modelo de amplificación de defectos, resolver el siguiente. Ejercicio:

- a-) Se generan 86 errores en la 1era etapa de desarrollo de un paquete de software.
- b-) De los errores de la etapa anterior, el 22% pasa inadvertido, y los demás errores se amplifican de acuerdo a un factor $x=1,3$. El porcentaje de detección es del 40%.
- c-) El 60% de los errores de la etapa anterior pasan inadvertidos, y el 40% es amplificado de acuerdo a un factor $x=2$. Se generan 50 nuevos errores, y el porcentaje de detección de esta etapa es 50% mayor que en la etapa anterior.
- d-) En las 3 últimas etapas no se amplifican, pero si se generan 10, 25 y 50 errores en cada etapa, teniendo un % de detección del 30,60 y 80 %.

¿De no utilizar esta técnica de revisión, qué cantidad de errores tendría el proyecto al finalizar?

0	0%	86	19	40%	65	39	60%	57
0			$67 \cdot 1,3 = 88$			$26 \cdot 2 = 52$		
86			0			50		

57	30%	47	47	60%	29	29	80%	16
0			0			0		
10			25			50		

Al finalizar el proceso, se tienen 16 errores. De no haber utilizado el modelo de amplificación de defectos, el resultado hubiera sido el siguiente:

0	0%	86	19	0%	107	65	0%	199
0			$67 \cdot 1,3 = 88$			$42 \cdot 2 = 84$		
86			0			50		

199	0%	209	209	0%	234	234	0%	284
0			0			0		
10			25			50		

Al final, habríamos tenido 284 errores (casi 18 veces más errores).

Ejercicio 3

Utilizando el modelo de amplificación de defectos, resolver el siguiente ejercicio:

- a-) Se generan 40 errores en la 1era etapa de desarrollo de una aplicación. Se detecta el 20 % de los errores.
- b-) De los errores de la etapa anterior, el 50% pasa inadvertido, y los demás errores se amplifican de acuerdo a un factor $x=3$. El porcentaje de detección es del 10%.
- c-) El 70% de los errores de la etapa anterior pasan inadvertidos, y el 30% es amplificado de acuerdo a un factor $x=2$. Se generan 50 nuevos errores, y el porcentaje de detección de esta etapa es del 50%.
- d-) En las 3 últimas etapas no se amplifican, ni se generan nuevos errores, teniendo un % de detección del 50,60 y 30 %.

0	20%	32	16	10%	58	41	50%	63
0			$16 \cdot 3 = 48$			$17 \cdot 2 = 34$		
40			0			50		

63	50%	32	32	60%	13	13	30%	10
0			0			0		
0			0			0		

Ejercicio 4

Utilizando el modelo de amplificación de defectos, resolver el siguiente. Ejercicio:

- a-) Se generan 120 errores en la 1era etapa de desarrollo de un paquete de software.
- b-) De los errores de la etapa anterior, el 32% pasa inadvertido, y los demás errores se amplifican de acuerdo a un factor $x=2,3$. El porcentaje de detección es del 30%.
- c-) El 40% de los errores de la etapa anterior pasan inadvertidos, y el 60% es amplificado de acuerdo a un factor $x=2$. Se generan 150 nuevos errores, y el porcentaje de detección de esta etapa es 40% mayor que en la etapa anterior.
- d-) En las 3 últimas etapas no se amplifican, pero si se generan 30, 25 y 60 errores en cada etapa, teniendo un % de detección del 40,60 y 80 %.

¿De no utilizar esta técnica de revisión, en qué porcentaje sería mayor la cantidad de errores al finalizar el proyecto?

0	0%	120	39	30%	159	64	42%	235
0			$81 * 2,3 = 187$			$95 * 2 = 190$		
120			0			150		
235	40%	159	159	60%	74	74	80%	27
0			0			0		
30			25			60		

Al finalizar el proceso, se tienen 27 errores. De no haber utilizado el modelo de amplificación de defectos, el resultado hubiera sido el siguiente:

0	0%	120	39	0%	226	91	0%	517
0			$81 * 2,3 = 187$			$138 * 2 = 276$		
120			0			150		
517	0%	527	527	0%	552	552	0%	602
0			0			0		
10			25			50		

Al final, hubiéramos tenido 602 errores (más de 22 veces más errores).

Tema II: Índice De Madurez De Software

Ejercicio 1

En cierto Departamento de Sistemas se realiza el cálculo de los sistemas implementados en la empresa:

En el sistema A:

- Fueron 40 los módulos iniciales
- Fueron adicionados 10 módulos, modificados 5 y borrados 7.

En el sistema B:

- Los módulos iniciales fueron 131
- Fueron modificados 25 y borrados 7.

Calcular el IMS de ambos sistemas.

Para el sistema A, los datos son como sigue:

- $MT = 40 + 10 - 7 = 43$
- $Fa = 10$
- $Fm = 5$
- $Fe = 7$

Así, el IMS para el sistema A es:

$$IMS = (MT - (Fa + Fm + Fe)) / MT = (43 - (10 + 5 + 7)) / 43$$

IMS = 0,4884 (se recomienda una reingeniería del sistema)

De manera análoga para el sistema B se tiene:

- $MT = 131 - 7 = 124$
- $Fa = 25$
- $Fm = 0$
- $Fe = 7$

Por tanto el IMS será:

$$IMS = (MT - (Fa + Fm + Fe)) / MT = (124 - (25 + 0 + 7)) / 124$$

IMS = 0,7419

PROCEDURE COMPLEJIDAD1

```

Do while .not. eof()
  saldo(inicio) = (existe + cuarente) - (comprome)
  for k=(inicio) to (inicio+12)
    i=circ(k)
    if i <> inicio
      if i=1
        saldo(i) = (saldo(12) + fabri(12)) - estima(i)
      else
        saldo(i) = (saldo(i-1) + fabri(i-1)) - estima
      (i)
    Endif
  endif
  store 0 to con,reponer
  j = i
  for con = i to (i+(a->in_pun_reo)-1)
    j=ind(con)
    reponer = reponer + estima(j)
  endfor

  if saldo(i) <= (reponer + (estima(ind(j))*5))
    store 0 to con,afa
    for con = i to i+(a->in_lot_com+a->in_pun_reo)-1
      j=ind(con)
      afa = afa + estima(j)
    endfor
    afa=afa-saldo(i)
  endif
endfor
Do grabar

```

**Tema III: Complejidad
Ciclomática De Mc Cabe****Ejercicio 1**

Considérese el grafo construido en la siguiente página tomando como base el código proveído.

La complejidad ciclomática (M) viene dada por:

$$M = E - N + 2 \text{ (aristas - nodos + 2)}$$

$$M = 37 - 31 + 2 = 8.$$

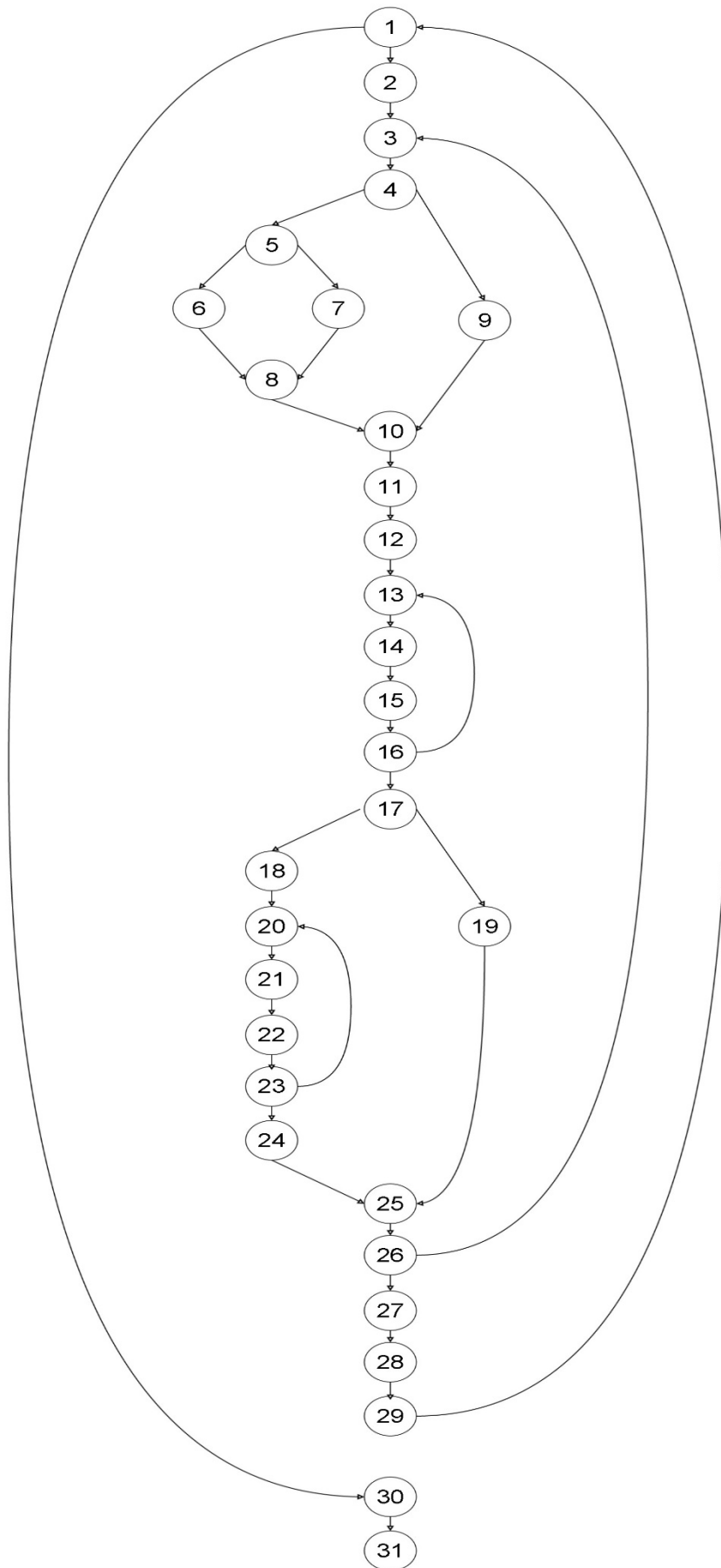
O bien;

$$M = P + 1 \text{ (P es la cantidad de nodos de los cuales salen dos o más aristas)}$$

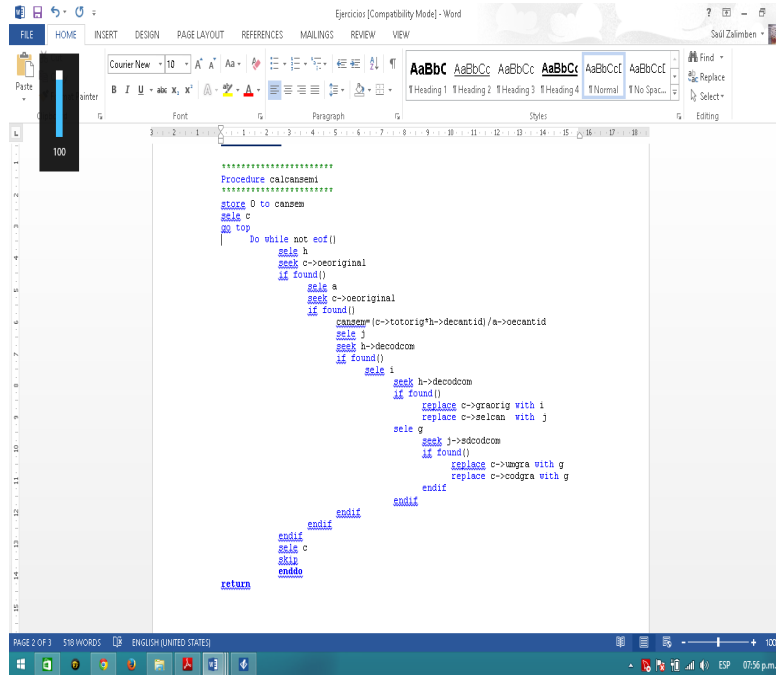
$$M = 7 + 1 = 8.$$

Una última forma de calcularla seria sumando la cantidad de regiones cerradas internas y externas que se forman en el grafo, en cuyo caso la complejidad ciclomática vendría dada por:

$$M = 7 \text{ (regiones internas)} + 1 \text{ (región externa)} = 8.$$



Ejercicio 2



Considerando ahora el grafo construido teniendo en cuenta este código, de manera análoga al ejercicio anterior calculamos el valor de complejidad ciclomática (M) del mismo de las tres maneras diferentes antes definidas.

$$M = E - N + 2$$

$$M = 42 - 37 + 2 = 7.$$

$$M = P + 1 \text{ (P es la cantidad de nodos de los cuales salen dos o más aristas)}$$

$$M = 6 + 1 = 7.$$

$$M = 6 \text{ (regiones internas)} + 1 \text{ (región externa)} = 7.$$

The screenshot shows a Microsoft Word document with a VBA macro named "PROCEDIMIENTO PRACTICO". The macro is written in Spanish and calculates the time taken for a person to travel a certain distance, considering different speeds and a break. The document is in the "Normal" style, and the VBA editor is open, showing the code in the "VBA Project - This Document" pane.

```

PROCEDIMIENTO PRACTICO
DO WHILE 1/100 < 0
    a = .f
    b = 0
    c = 0
    SELECT Asistenc
    CASE 1: b = b + 1
    CASE 2: c = c + 1
    CASE 3: c = c + 1
    CASE 4: c = c + 1
    CASE 5: c = c + 1
    CASE 6: c = c + 1
    CASE 7: c = c + 1
    CASE 8: c = c + 1
    CASE 9: c = c + 1
    CASE 10: c = c + 1
    CASE 11: c = c + 1
    CASE 12: c = c + 1
    CASE 13: c = c + 1
    CASE 14: c = c + 1
    CASE 15: c = c + 1
    CASE 16: c = c + 1
    CASE 17: c = c + 1
    CASE 18: c = c + 1
    CASE 19: c = c + 1
    CASE 20: c = c + 1
    CASE 21: c = c + 1
    CASE 22: c = c + 1
    CASE 23: c = c + 1
    CASE 24: c = c + 1
    CASE 25: c = c + 1
    CASE 26: c = c + 1
    CASE 27: c = c + 1
    CASE 28: c = c + 1
    CASE 29: c = c + 1
    CASE 30: c = c + 1
    CASE 31: c = c + 1
    CASE 32: c = c + 1
    CASE 33: c = c + 1
    CASE 34: c = c + 1
    CASE 35: c = c + 1
    CASE 36: c = c + 1
    CASE 37: c = c + 1
    CASE 38: c = c + 1
    CASE 39: c = c + 1
    CASE 40: c = c + 1
    CASE 41: c = c + 1
    CASE 42: c = c + 1
    CASE 43: c = c + 1
    CASE 44: c = c + 1
    CASE 45: c = c + 1
    CASE 46: c = c + 1
    CASE 47: c = c + 1
    CASE 48: c = c + 1
    CASE 49: c = c + 1
    CASE 50: c = c + 1
    CASE 51: c = c + 1
    CASE 52: c = c + 1
    CASE 53: c = c + 1
    CASE 54: c = c + 1
    CASE 55: c = c + 1
    CASE 56: c = c + 1
    CASE 57: c = c + 1
    CASE 58: c = c + 1
    CASE 59: c = c + 1
    CASE 60: c = c + 1
    CASE 61: c = c + 1
    CASE 62: c = c + 1
    CASE 63: c = c + 1
    CASE 64: c = c + 1
    CASE 65: c = c + 1
    CASE 66: c = c + 1
    CASE 67: c = c + 1
    CASE 68: c = c + 1
    CASE 69: c = c + 1
    CASE 70: c = c + 1
    CASE 71: c = c + 1
    CASE 72: c = c + 1
    CASE 73: c = c + 1
    CASE 74: c = c + 1
    CASE 75: c = c + 1
    CASE 76: c = c + 1
    CASE 77: c = c + 1
    CASE 78: c = c + 1
    CASE 79: c = c + 1
    CASE 80: c = c + 1
    CASE 81: c = c + 1
    CASE 82: c = c + 1
    CASE 83: c = c + 1
    CASE 84: c = c + 1
    CASE 85: c = c + 1
    CASE 86: c = c + 1
    CASE 87: c = c + 1
    CASE 88: c = c + 1
    CASE 89: c = c + 1
    CASE 90: c = c + 1
    CASE 91: c = c + 1
    CASE 92: c = c + 1
    CASE 93: c = c + 1
    CASE 94: c = c + 1
    CASE 95: c = c + 1
    CASE 96: c = c + 1
    CASE 97: c = c + 1
    CASE 98: c = c + 1
    CASE 99: c = c + 1
    CASE 100: c = c + 1
    CASE 101: c = c + 1
    CASE 102: c = c + 1
    CASE 103: c = c + 1
    CASE 104: c = c + 1
    CASE 105: c = c + 1
    CASE 106: c = c + 1
    CASE 107: c = c + 1
    CASE 108: c = c + 1
    CASE 109: c = c + 1
    CASE 110: c = c + 1
    CASE 111: c = c + 1
    CASE 112: c = c + 1
    CASE 113: c = c + 1
    CASE 114: c = c + 1
    CASE 115: c = c + 1
    CASE 116: c = c + 1
    CASE 117: c = c + 1
    CASE 118: c = c + 1
    CASE 119: c = c + 1
    CASE 120: c = c + 1
    CASE 121: c = c + 1
    CASE 122: c = c + 1
    CASE 123: c = c + 1
    CASE 124: c = c + 1
    CASE 125: c = c + 1
    CASE 126: c = c + 1
    CASE 127: c = c + 1
    CASE 128: c = c + 1
    CASE 129: c = c + 1
    CASE 130: c = c + 1
    CASE 131: c = c + 1
    CASE 132: c = c + 1
    CASE 133: c = c + 1
    CASE 134: c = c + 1
    CASE 135: c = c + 1
    CASE 136: c = c + 1
    CASE 137: c = c + 1
    CASE 138: c = c + 1
    CASE 139: c = c + 1
    CASE 140: c = c + 1
    CASE 141: c = c + 1
    CASE 142: c = c + 1
    CASE 143: c = c + 1
    CASE 144: c = c + 1
    CASE 145: c = c + 1
    CASE 146: c = c + 1
    CASE 147: c = c + 1
    CASE 148: c = c + 1
    CASE 149: c = c + 1
    CASE 150: c = c + 1
    CASE 151: c = c + 1
    CASE 152: c = c + 1
    CASE 153: c = c + 1
    CASE 154: c = c + 1
    CASE 155: c = c + 1
    CASE 156: c = c + 1
    CASE 157: c = c + 1
    CASE 158: c = c + 1
    CASE 159: c = c + 1
    CASE 160: c = c + 1
    CASE 161: c = c + 1
    CASE 162: c = c + 1
    CASE 163: c = c + 1
    CASE 164: c = c + 1
    CASE 165: c = c + 1
    CASE 166: c = c + 1
    CASE 167: c = c + 1
    CASE 168: c = c + 1
    CASE 169: c = c + 1
    CASE 170: c = c + 1
    CASE 171: c = c + 1
    CASE 172: c = c + 1
    CASE 173: c = c + 1
    CASE 174: c = c + 1
    CASE 175: c = c + 1
    CASE 176: c = c + 1
    CASE 177: c = c + 1
    CASE 178: c = c + 1
    CASE 179: c = c + 1
    CASE 180: c = c + 1
    CASE 181: c = c + 1
    CASE 182: c = c + 1
    CASE 183: c = c + 1
    CASE 184: c = c + 1
    CASE 185: c = c + 1
    CASE 186: c = c + 1
    CASE 187: c = c + 1
    CASE 188: c = c + 1
    CASE 189: c = c + 1
    CASE 190: c = c + 1
    CASE 191: c = c + 1
    CASE 192: c = c + 1
    CASE 193: c = c + 1
    CASE 194: c = c + 1
    CASE 195: c = c + 1
    CASE 196: c = c + 1
    CASE 197: c = c + 1
    CASE 198: c = c + 1
    CASE 199: c = c + 1
    CASE 200: c = c + 1
    CASE 201: c = c + 1
    CASE 202: c = c + 1
    CASE 203: c = c + 1
    CASE 204: c = c + 1
    CASE 205: c = c + 1
    CASE 206: c = c + 1
    CASE 207: c = c + 1
    CASE 208: c = c + 1
    CASE 209: c = c + 1
    CASE 210: c = c + 1
    CASE 211: c = c + 1
    CASE 212: c = c + 1
    CASE 213: c = c + 1
    CASE 214: c = c + 1
    CASE 215: c = c + 1
    CASE 216: c = c + 1
    CASE 217: c = c + 1
    CASE 218: c = c + 1
    CASE 219: c = c + 1
    CASE 220: c = c + 1
    CASE 221: c = c + 1
    CASE 222: c = c + 1
    CASE 223: c = c + 1
    CASE 224: c = c + 1
    CASE 225: c = c + 1
    CASE 226: c = c + 1
    CASE 227: c = c + 1
    CASE 228: c = c + 1
    CASE 229: c = c + 1
    CASE 230: c = c + 1
    CASE 231: c = c + 1
    CASE 232: c = c + 1
    CASE 233: c = c + 1
    CASE 234: c = c + 1
    CASE 235: c = c + 1
    CASE 236: c = c + 1
    CASE 237: c = c + 1
    CASE 238: c = c + 1
    CASE 239: c = c + 1
    CASE 240: c = c + 1
    CASE 241: c = c + 1
    CASE 242: c = c + 1
    CASE 243: c = c + 1
    CASE 244: c = c + 1
    CASE 245: c = c + 1
    CASE 246: c = c + 1
    CASE 247: c = c + 1
    CASE 248: c = c + 1
    CASE 249: c = c + 1
    CASE 250: c = c + 1
    CASE 251: c = c + 1
    CASE 252: c = c + 1
    CASE 253: c = c + 1
    CASE 254: c = c + 1
    CASE 255: c = c + 1
    CASE 256: c = c + 1
    CASE 257: c = c + 1
    CASE 258: c = c + 1
    CASE 259: c = c + 1
    CASE 260: c = c + 1
    CASE 261: c = c + 1
    CASE 262: c = c + 1
    CASE 263: c = c + 1
    CASE 264: c = c + 1
    CASE 265: c = c + 1
    CASE 266: c = c + 1
    CASE 267: c = c + 1
    CASE 268: c = c + 1
    CASE 269: c = c + 1
    CASE 270: c = c + 1
    CASE 271: c = c + 1
    CASE 272: c = c + 1
    CASE 273: c = c + 1
    CASE 274: c = c + 1
    CASE 275: c = c + 1
    CASE 276: c = c + 1
    CASE 277: c = c + 1
    CASE 278: c = c + 1
    CASE 279: c = c + 1
    CASE 280: c = c + 1
    CASE 281: c = c + 1
    CASE 282: c = c + 1
    CASE 283: c = c +
```

