

Se cere: a) elaborarea CFG

- I. Metoda isPrime (int): boolean;

```

graph TD
    1((1)) --> 2{2}
    2 -- T --> 3((3))
    2 -- F --> 4{4}
    3 --> 5((5))
    4 --> 5
    5 --> 6{6}
    6 -- T --> 7((7))
    6 -- F --> 8{8}
    7 --> 9((9))
    8 --> 9
    9 --> 10((10))
    10 --> 11((11))
    11 --> 12((12))
    12 --> 13((13))
    13 --> 14((14))
    14 --> 15((15))
    15 --> 16((16))
    16 --> 17((17))
    17 --> 18((18))
    18 --> 19((19))
    19 --> 20((20))
    20 --> 21((21))
    21 --> 22((22))
    22 --> 23((23))
    23 --> 24((24))
    24 --> 25((25))
    25 --> 26((26))
    26 --> 27((27))
    27 --> 28((28))
    28 --> 29((29))
    29 --> 30((30))
    30 --> 31((31))
    31 --> 32((32))
    32 --> 33((33))
    33 --> 34((34))
    34 --> 35((35))
    35 --> 36((36))
    36 --> 37((37))
    37 --> 38((38))
    38 --> 39((39))
    39 --> 40((40))
    40 --> 41((41))
    41 --> 42((42))
    42 --> 43((43))
    43 --> 44((44))
    44 --> 45((45))
    45 --> 46((46))
    46 --> 47((47))
    47 --> 48((48))
    48 --> 49((49))
    49 --> 50((50))
    50 --> 51((51))
    51 --> 52((52))
    52 --> 53((53))
    53 --> 54((54))
    54 --> 55((55))
    55 --> 56((56))
    56 --> 57((57))
    57 --> 58((58))
    58 --> 59((59))
    59 --> 60((60))
    60 --> 61((61))
    61 --> 62((62))
    62 --> 63((63))
    63 --> 64((64))
    64 --> 65((65))
    65 --> 66((66))
    66 --> 67((67))
    67 --> 68((68))
    68 --> 69((69))
    69 --> 70((70))
    70 --> 71((71))
    71 --> 72((72))
    72 --> 73((73))
    73 --> 74((74))
    74 --> 75((75))
    75 --> 76((76))
    76 --> 77((77))
    77 --> 78((78))
    78 --> 79((79))
    79 --> 80((80))
    80 --> 81((81))
    81 --> 82((82))
    82 --> 83((83))
    83 --> 84((84))
    84 --> 85((85))
    85 --> 86((86))
    86 --> 87((87))
    87 --> 88((88))
    88 --> 89((89))
    89 --> 90((90))
    90 --> 91((91))
    91 --> 92((92))
    92 --> 93((93))
    93 --> 94((94))
    94 --> 95((95))
    95 --> 96((96))
    96 --> 97((97))
    97 --> 98((98))
    98 --> 99((99))
    99 --> 100((100))
    100 --> 101((101))
    101 --> 102((102))
    102 --> 103((103))
    103 --> 104((104))
    104 --> 105((105))
    105 --> 106((106))
    106 --> 107((107))
    107 --> 108((108))
    108 --> 109((109))
    109 --> 110((110))
    110 --> 111((111))
    111 --> 112((112))
    112 --> 113((113))
    113 --> 114((114))
    114 --> 115((115))
    115 --> 116((116))
    116 --> 117((117))
    117 --> 118((118))
    118 --> 119((119))
    119 --> 120((120))
    120 --> 121((121))
    121 --> 122((122))
    122 --> 123((123))
    123 --> 124((124))
    124 --> 125((125))
    125 --> 126((126))
    126 --> 127((127))
    127 --> 128((128))
    128 --> 129((129))
    129 --> 130((130))
    130 --> 131((131))
    131 --> 132((132))
    132 --> 133((133))
    133 --> 134((134))
    134 --> 135((135))
    135 --> 136((136))
    136 --> 137((137))
    137 --> 138((138))
    138 --> 139((139))
    139 --> 140((140))
    140 --> 141((141))
    141 --> 142((142))
    142 --> 143((143))
    143 --> 144((144))
    144 --> 145((145))
    145 --> 146((146))
    146 --> 147((147))
    147 --> 148((148))
    148 --> 149((149))
    149 --> 150((150))
    150 --> 151((151))
    151 --> 152((152))
    152 --> 153((153))
    153 --> 154((154))
    154 --> 155((155))
    155 --> 156((156))
    156 --> 157((157))
    157 --> 158((158))
    158 --> 159((159))
    159 --> 160((160))
    160 --> 161((161))
    161 --> 162((162))
    162 --> 163((163))
    163 --> 164((164))
    164 --> 165((165))
    165 --> 166((166))
    166 --> 167((167))
    167 --> 168((168))
    168 --> 169((169))
    169 --> 170((170))
    170 --> 171((171))
    171 --> 172((172))
    172 --> 173((173))
    173 --> 174((174))
    174 --> 175((175))
    175 --> 176((176))
    176 --> 177((177))
    177 --> 178((178))
    178 --> 179((179))
    179 --> 180((180))
    180 --> 181((181))
    181 --> 182((182))
    182 --> 183((183))
    183 --> 184((184))
    184 --> 185((185))
    185 --> 186((186))
    186 --> 187((187))
    187 --> 188((188))
    188 --> 189((189))
    189 --> 190((190))
    190 --> 191((191))
    191 --> 192((192))
    192 --> 193((193))
    193 --> 194((194))
    194 --> 195((195))
    195 --> 196((196))
    196 --> 197((197))
    197 --> 198((198))
    198 --> 199((199))
    199 --> 200((200))
    200 --> 201((201))
    201 --> 202((202))
    202 --> 203((203))
    203 --> 204((204))
    204 --> 205((205))
    205 --> 206((206))
    206 --> 207((207))
    207 --> 208((208))
    208 --> 209((209))
    209 --> 210((210))
    210 --> 211((211))
    211 --> 212((212))
    212 --> 213((213))
    213 --> 214((214))
    214 --> 215((215))
    215 --> 216((216))
    216 --> 217((217))
    217 --> 218((218))
    218 --> 219((219))
    219 --> 220((220))
    220 --> 221((221))
    221 --> 222((222))
    222 --> 223((223))
    223 --> 224((224))
    224 --> 225((225))
    225
```

b) drumuri independente

D2. $1 \rightarrow 2F \rightarrow 4T \rightarrow 5 \rightarrow 11 \rightarrow 12$

D4. $1 \rightarrow 2F \rightarrow 4F \rightarrow 6 \rightarrow 7T \rightarrow 8T \rightarrow 9 \rightarrow 10 \rightarrow 7F \rightarrow 11 \rightarrow 12$

D5. $1 \rightarrow 2F \rightarrow 4F \rightarrow 6 \rightarrow 7T \rightarrow 8F \rightarrow 10 \rightarrow 7F \rightarrow 11 \rightarrow 12$

DB. $1 \rightarrow 2i \rightarrow 4F \rightarrow 6 \rightarrow 7F \rightarrow 11 \rightarrow 12$

c) CC = nr. regiuni = 5

$$CC = E - N + 2 = 15 - 12 + 2 = 5$$

$$CC = \text{nr. conditii} + 1 = 4 + 1 = 5$$

②

- 2°) acoperirea instructorilor (sc)

- 3°) acoperirea deciziilor, condițiilor, deciziilor și condițiilor, condițiilor multiple (dc, cc, dcc, mcc)

e.g.: if $(a > 0)$
then I_1
else I_2

- condition: $a > 0 \rightarrow$ evaluate $\log \frac{T}{T_0}$

- if $((a > 0) \parallel (b < 0))$
 then I_1
 else I_2

-decide: $((a > 0) \vee (b < 0)) \Rightarrow \langle \top, \perp \rangle$

- Conditio: - $a > 0 \Rightarrow \begin{matrix} \top \\ \neg \\ \top \end{matrix}$
- $b < 0 \Rightarrow \begin{matrix} \top \\ \neg \\ \neg \\ \top \end{matrix}$

- decizia $\diamond 2$ $n < 0 \Rightarrow$ 1 decizie $\begin{matrix} \swarrow \\ T \\ \downarrow \\ F \end{matrix}$ | $\Rightarrow 2TCs$
 • 1 conditie $\begin{matrix} \swarrow \\ T \\ \downarrow \\ F \end{matrix}$
 $de \equiv cc \equiv dcc \equiv mcc$

- decizia $\Diamond (M \Rightarrow 0) \vee (M \Rightarrow 1)$

- 2 condiții, fiecare $\langle T, F \rangle$
- 1 decizie $\langle T, F \rangle$

	$n=0$	$n=1$	$(n=0) \vee (n=1)$
1° $n=0 \Rightarrow$	T	F	T
2° $n=1 \Rightarrow$	F	T	T
3° $n=3 \Rightarrow$	F	F	F
4° —	T	T	—

$\left. \begin{matrix} 1^\circ + 2^\circ \\ 1^\circ + 3^\circ \end{matrix} \right\} \text{dec}$

 $\underbrace{\quad}_{\text{must be true simultan}}$

 $\left. \begin{matrix} \text{cc} \\ \text{dec} \end{matrix} \right\} \text{mcc, dcc=mcc}$

 $\Rightarrow 3 \text{ TCs}$

- decizia $\diamond 7$ $d \leq \lfloor \frac{n}{2} \rfloor$ • 1 decizie
• 1 condiție | $\Rightarrow 2 \text{ TCs}$

- decizia $\diamond 8$ $n \% 2 == 0$ • 1 decizie
• 1 condiție | $\Rightarrow 2 \text{ TCs}$

4°) acoperirea buclelor (lc)

- buclă simplă \Rightarrow se proiectează TCs pentru 7 situații diferite:

- 0 iterații - nu intra în buclă
- 1 iterație
- 2 iterații
- $n-1$ iterații
- n iterații
- $n+1$ iterații
- m iterații, $1 < m < n$

- $n = ?$ pentru metoda isPrime(int): boolean

- decizia $\diamond 7$ $d \leq \lfloor \frac{n}{2} \rfloor$ • $d = 2 \Rightarrow$ nr. de iterații = $\lfloor \frac{n}{2} \rfloor - 1$

\Rightarrow se proiectează TCs pentru:

- 0 iterații - 1 TC
- 1 iterație - 1 TC
- 2 iterații - 1 TC
- $(\lfloor \frac{n}{2} \rfloor - 1) - 1 = \lfloor \frac{n}{2} \rfloor - 2$ iterații - 1 TC
- $\lfloor \frac{n}{2} \rfloor - 1$ iterații - 1 TC
- $(\lfloor \frac{n}{2} \rfloor - 1) + 1 = \lfloor \frac{n}{2} \rfloor$ iterații - 1 TC
- m iterații, $1 < m < \lfloor \frac{n}{2} \rfloor - 1$ - inclusiv în $\lfloor \frac{n}{2} \rfloor - 2$ iterații

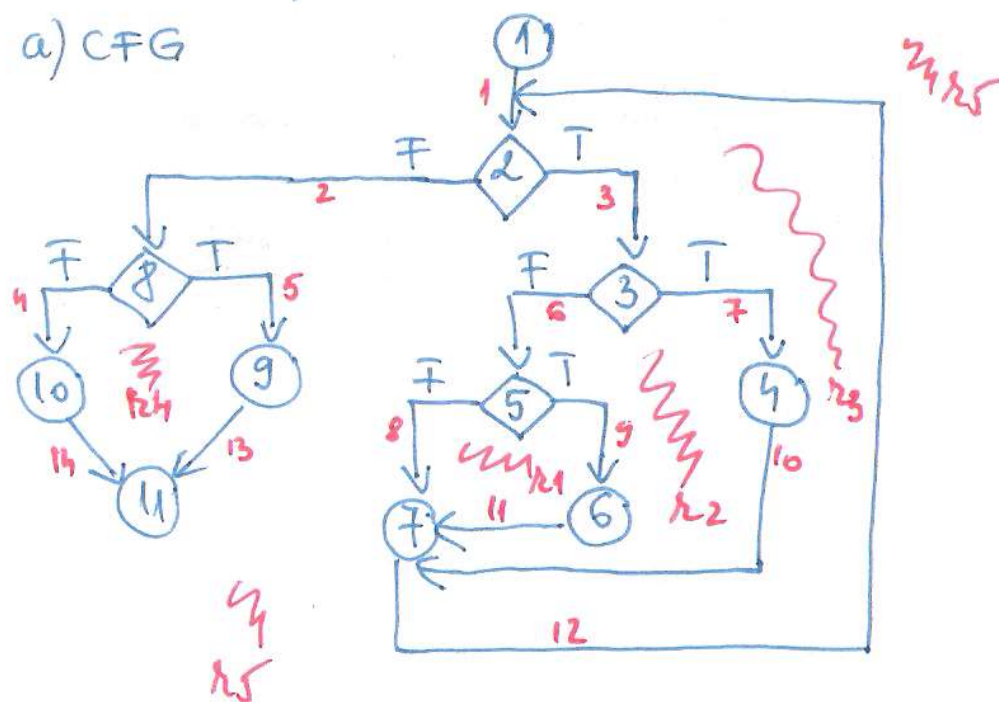
6 TCs

nTC	Input	Output	Statement (sc)	Condition/Decision (dc, cc, dec, mcc)										Path (apc)					Bucle (lc)					(4)				
				$n < d$		$n == 0$		$n == 1$		$n == 0 n == 1$		$d \leq \lfloor \frac{n}{2} \rfloor$		$n \% d == 0$		1	2	4	5	3	0	1	2		$\lfloor \frac{n}{2} \rfloor - 2$	$\lfloor \frac{n}{2} \rfloor - 1$	$\lfloor \frac{n}{2} \rfloor$	m
				T	F	T	F	T	F	T	F	T	F	T	F													
TC01	2	true	1,2,4,6,7,11,12	*		*		*		*		*					*		*					(*)				
TC02	0	false	1,2,4,5,11,12	*	*			*	*					*				*										
TC03	1	false	1,2,4,5,11,12	*		*	*	*		*				*				*										
TC04	-1	exceptio	1,2,3,11,12	*										*				*										
TC05	4	false	1,2,4,6,7,8,9 10,11,12	*		*		*		*	*	*	*			*	*	*	*	*			*					
TC06	7	true	1,2,4,6,7,8 10,12	*		*	*	*		*	*	*	*			*	*	*	*	*	*		*	*				
TC07	5	true	1,2,4,6,7,8 10,12	*		*	*	*		*	*	*	*			*	*	*	*	*	*		*	*				
																						?			?			

ii Metoda computeMaxCounter(): int, :

5

a) CFG



b) drumuri independente

D1. 1 → 2F → 8T → 9 → 11 (size = 0)

D2. 1 → 2F → 8F → 10 → 11 (size > 0)

D3. 1 → 2T → 3F → 5F → 7 → 2F → 8F → 10 → 11 (size > 0)

D4. 1 → 2T → 3F → 5T → 6 → 7 → 2F → 8F → 10 → 11 (size > 0)

D5. 1 → 2T → 3T → 4 → 7 → 2F → 8F → 10 → 11 (size > 0)

- dacă s-a parcurs ramura 2T (size > 0) ⇒ se ajunge în vârful 11 doar prin ramura 8F (size > 0)

c) CC = nr. regiuni = 5

$$CC = E - N + 2 = 14 - 11 + 2 = 5$$

$$CC = \text{nr. condiții} + 1 = 4 + 1 = 5$$

d) 1°) acoperirea drumurilor (apc)

- 1 TC pentru fiecare drum

D2 inclus în D3, D4, D5 ⇒ 4 TCs pentru 100% apc

2°) acoperirea instrucțiunilor (sc)

- instrucțiunile se află pe D1...D5 ⇒ 4 TCs pentru 100% sc

3) acoperirea deciziilor, condițiilor, deciziilor și condițiilor, condițiilor multiple (dc, cc, dcc, mcc)

⑥

- decizia ② $\text{index} < \text{list.size}()$ • 1 condiție $< \frac{T}{F}$ | $\Rightarrow 2 \text{TCs}$
• 1 decizie $< \frac{T}{F}$

- decizia ③ $\text{list.get}(\text{index}) > \text{list.get}(\text{posMax})$
• 1 condiție | $\Rightarrow 2 \text{TCs}$
• 1 decizie

- decizia ⑤ $\text{list.get}(\text{posMax}) == \text{list.get}(\text{index})$
• 1 condiție | $\Rightarrow 2 \text{TCs}$
• 1 decizie

- decizia ⑧ $\text{list.size}() == 0$ • 1 condiție | $\Rightarrow 2 \text{TCs}$
• 1 decizie

4) acoperirea buclelor (lc)

- decizia ② $\text{index} < \text{list.size}()$

a) • 0 iterații (nu se intră în buclă) - 1TC

b) • 1 iterație - 1TC

c) • 2 iterații - 1TC

d) • $\text{size} - 1$ iterații - nu se poate simula

e) • size iterații - 1TC

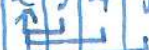
f) • $\text{size} + 1$ iterații - nu se poate simula

g) • m iterații, $1 < m < \text{size}$ - inclus în $(\text{size} - 1)$

} 4TCs

- condiția inclusă în decizia ② asigură parcurgerea listei de la $\text{index} = 0$ până la $\text{index} = \text{size} - 1$

- pentru a analiza cazurile d) și g) ar fi necesar ca în cadrul buclei while să apară 4 instrucțiuni / condiții neplănuite

Nr TC	Input	Output	Statement (sc)	Condition / Decision (dc, cc, dcc, mcc)								Path (apc)					Bucle (lc)							⑦
				index < size		$l(\text{index}) > l(\text{posMax})$		$l(\text{posMax}) == l(\text{index})$		size == 0		1	2	3	4	5	0	1	2	n	n-1	n+1	m	
				T	F	T	F	T	F	T	F													
TC01	0, []	0	1, 2, 8, 9, 10, 11		*					*		*				*				3	3	3		
TC02	2, [1, 1]	2	1, 2, 3, 5, 6, 7, 8, 10, 11	*	*		*	*		*		*	*	*			*	*						
TC03	3, [1, 3, 2]	1	1, 2, 3, 5, 6, 7, 8, 10, 11	*	*	*	*	*	*	*		*	*	*	*			*						
TC04	1, [2]	1	1, 2, 3, 5, 7, 8, 10, 11	*	*		*	*		*		*	*	*			*							

D1 = size == 0

D2 = acoperit de fiecare dată când size > 0

D3 = acoperit dacă elementul de pe indexul curent < maxCounter

D4 = acoperit când se crește frecvența de apariție a maximumului

D5 = acoperit când se schimbă maximumul