

Se consideră secvențele de cod de mai jos.**I. Metoda isPrime verifică dacă un număr natural n dat este prim.**

	/** * @param n natural number * @return true, false * @exception InvalidValueException if n < 0 */
	private boolean isPrime(int n) throws InvalidValueException{
	boolean result = true ;
	if (n<0)
	throw new InvalidValueException("negative value was provided!");
	else if ((n == 0) (n == 1))
	result = false ;
	else {
	int d = 2;
	while (d<=Math.ceil(n/2)){
	if (n%d == 0)
	result = false ;
	d++;
	}
	}
	return result;
	}

II. Metoda computeMaxCounter determină frecvența de apariție a valorii maxime dintr-o listă de elemente date.

	/** * @param list - lista de valori intregi * @return frecventa de aparitie a valorii maxime */
	public int computeMaxCounter(List<Integer> list){
	int countMax, index, posMax;
	index=0;countMax=0;posMax=0;
	while (index<list.size()){
	if (list.get(index) > list.get(posMax)){
	posMax=index;
	countMax=1;
	}
	else
	if (list.get(posMax)== list.get(index))
	countMax++;
	index++;
	}
	if (list.size()==0)
	return -1;
	else return countMax;
	}

I. Se cere:

- a. elaborarea CFG;
- b. identificarea drumurilor independente;
- c. calculul CC folosind cele 3 formule.
- d. proiectarea de cazuri de testare pentru acoperirea tuturor:
 - drumurilor;
 - instrucțiunilor;
 - deciziilor, condițiilor, deciziilor și condițiilor, condițiilor multiple;
 - buclelor.

CFG. Construire

- pași de elaborare a unui CFG:
 1. se numerează unic fiecare element de structură secvențială (calcul) și condițională (decizie);
 2. se începe pornind de la vârful de intrare, care are (de obicei) numărul 1;
 3. se adaugă celelalte vârfuri corespunzătoare structurilor numerotate și se unesc prin arce, evidențiind transmiterea controlului în cadrul programului;
 4. la final, toate ieșirile posibile din program se unesc în vârful de ieșire;

Complexitatea ciclomatică. Definiție

- **complexitatea ciclomatică** (*engl.* **McCabe's cyclomatic complexity, CC**):
 - *metrică software* aplicată pentru măsurarea cantitativă a complexității logice a unui program;
 - *permite determinarea numărului de drumuri independente din mulțimea de bază a unui CFG*;
- modalități de calcul a CC la nivelul CFG:
 - $CC = \text{numărul de regiuni din CFG}$;
 - $CC = E - N + 2$, unde E - #arce, N - #vârfuri ;
 - $CC = P + 1$, unde P - #vârfuri condiție.
- **regiune**:
 - zonă a CFG marginită parțial sau în totalitate de arce și vârfuri;