Studiaţi parţial corectitudinea, terminarea şi total corectitudinea pentru următoarele metode:

[1]. Cel mai mare divizor comun a două numere naturale:

```
/**
 * φ(X): n1 ϵ N, n2 ϵ N
 * Ψ(X, Z): d=cmmdc(n1,n2)
 * @param n1 - numar natural
 * @param n2 - numar natural
 * @return d=cmmdc(n1, n2)
 */
int cmmdc(int n1, int n2){
    •
    int d = n1, i = n2;
    while ((d!=i) && (i>0)) {
        •
        if (d>i)
            d = d - i;
        else i = i - d;
    }
    return d;
    •
}
```

[2]. Căutare binară:

```
/**
 * φ(X): L[0]<=a<=L[n-1], L[0]<= L[1]<=…<=L[n-1]
 * Ψ(X, Z): 0<=p<=n-1, L[p]<=a<L[p+1]
 * @param l - lista de valori intregi ordonata crescator
 * @param a - valoarea cautata
 * @return poz
 *        daca (a este in l)
 *          atunci se returneaza pozitia (0<=poz<=size-1),
 *          altfel se returneaza pozitia de inserare
 */
int search(List<Integer> l, int a) {
    •
    int s=0, d=l.size()-1;
    while (s<d) {
        •
        int m=(s+d)/2;
        if (a<=l.get(m))
            d=m;
        else s=m+1;
    }
    return s;
    •
}
```

Metoda lui Floyd. **Metoda aserțiunilor inductive**

Aplicabilitate:

- pentru a demonstra:
  1. parțial corectitudinea programului;
  2. terminarea programului;
  3. **total corectitudinea = parțial corectitudinea programului + terminarea programului.**

Folosește:

- **precondiție** – condiția satisfăcută de datele de intrare ale programului;
- **postcondiție** – condiția care trebuie satisfăcută de rezultatele programului;
- **algoritmul** – descrierea programului (codul sursă);

Etape de aplicare:

1. identificarea unui punct de tăietură în fiecare buclă;
2. identificarea unei mulțimi de aserțiuni inductive;
3. construirea și demonstrarea condițiilor de verificare/terminare.

Parțial corectitudine. **Etape de realizare**

1. se aleg puncte de tăietură în cadrul algoritmului:
   - două puncte de tăietură particulare: un punct de tăietură la începutul algoritmului, un punct de tăietură la sfârșitul algoritmului;
   - cel puțin un punct de tăietură în fiecare instrucțiune repetitivă;
2. pentru fiecare punct de tăietură se alege câte un predicat invariant (aserțiune):
   - punctul de intrare - φ(X);
   - punctul de ieșire - ψ(X,Z);
3. se construiesc și se demonstrează condițiile de verificare:
   1. $\forall X\ \forall Y\ (P_i(X,Y) \land R_{\alpha_{i,j}}(X,Y)\ \rightarrow\ P_j(X,\ r_{\alpha_{i,j}}(X,Y)))$;
   2. Y – vector de variabile cu rezultate intermediare;
   3. $\alpha_{i,j}$ – drumul de la punctul de tăietură i la punctul de tăietură j;
   4. P_i și P_j – predicate invariante în punctele de tăietură i și j asociate;
   5. $R_{\alpha_{i,j}}(X,Y)$ – predicat care dă condiția de parcurgere a drumului α;
   6. $r_{\alpha_{i,j}}(X,Y)$ – funcție care indică transformările variabilelor Y de pe drumul α;

Theorem

1. Dacă toate condițiile de verificare sunt adevărate atunci programul P este parțial corect în raport cu specificația $(\varphi(X), \psi(X,Z))$. *[Fre10]*

Terminarea algoritmului. **Etape de realizare.**

1. se aleg punctele de tăietură în cadrul algoritmului;
2. pentru fiecare punct de tăietură se alege câte un predicat invariant;
3. se alege o **mulțime convenabilă** M (i.e., o mulțime parțial ordonată, care nu conține nici un șir descrescător infinit) și o funcție descrescătoare u_i;
   - în punctul de tăietură i funcția aleasă este $u_i : D_X \times D_Y \rightarrow M$;
4. se scriu condițiile de terminare:
   - condiția de terminare pe drumul $\alpha_{i,j}$ este:
     $\forall X\ \forall Y\ (\varphi(X) \land R_{\alpha_{i,j}}(X,Y) \rightarrow (u_i(X,Y)\ >\ u_j(X,\ r_{\alpha_{i,j}}(X,Y))))$;
   - dacă s-a demonstrat parțial corectitudinea, atunci condiția de terminare poate fi:
     $\forall X\ \forall Y\ (P_i(X) \land R_{\alpha_{i,j}}(X,Y) \rightarrow (u_i(X,Y)\ >\ u_j(X,\ r_{\alpha_{i,j}}(X,Y))))$;
5. se demonstrează condițiile de terminare:
   - la trecerea de la un punctul de tăietură i la j valorile funcției u descresc, i.e., $u_i\ > u_j$.

Theorem

2. Dacă toate condițiile de terminare sunt adevărate atunci programul P se termină în raport cu predicatul $\varphi(X)$. *[Fre10]*