▲ Následující program `power` počítá pro přijatá čísla $z$ a $n$ mocninu $z^n$. Napište specifikaci tohoto programu, a pak ověřte jeho totální korektnost vzhledem k zapsané specifikaci.

```
float power(float z, int n) {
  float r = 1;
  for (int i = 1; i <= n; i = i + 1)
    r = r * z;
  return r;
}
```

▲ Program `linearSearch`, jehož kód je uveden níže, přijímá pole celých čísel a a snaží se zjistit, jestli v intervalu $[\mathtt{l}, \mathtt{u}]$ pole a leží hodnota e nebo ne. Zapište specifikaci tohoto programu, a pak ověřte jeho totální korektnost vzhledem ke specifikaci.

```
bool linearSearch(int[] a, int l, int u, int e) {
  for (int i = l; i <= u; i = i + 1)
    if (a[i] == e) return true;
  return false;
}
```

▲ Mějme dám program `binarySearch`, který přijme pole celých čísel a a snaží se zjistit, jestli v intervalu $[\mathtt{l}, \mathtt{u}]$ pole a leží hodnota e nebo ne. Zapište specifikaci tohoto programu, a pak ověřte jeho totální korektnost vzhledem ke specifikaci. Zde je kód programu.

```
bool binarySearch(int[] a, int l, int u, int e) {
  if (l > u) return false;
  int m = (l + u) div 2;
  if (a[m] == e) return true;
  if (a[m] < e)
    return binarySearch(a,m+1,u,e);
  else
    return binarySearch(a,l,m-1,e);
}
```

▲ Následující program bubbleSort přijímá pole celých čísel a a vrací jeho seřazenou kopii. Zapište specifikaci tohoto programu, a pak ověřte jeho totální korektnost vzhledem ke specifikaci.

```
int[] bubbleSort(int[] a0) {
  int[] a = a0;
  for (int i = |a| - 1; i > 0; i = i - 1)
    for (int j = 0; j < i; j = j + 1)
      if (a[j] > a[j+1]) {
        int m = a[j];
        a[j] = a[j+1];
        a[j+1] = m;
      }
  return a;
}
```

▲ Mějme následující algoritmus se vstupní a výstupní doménou $\mathbb{R}$:

```
input x
i ← x;
z ← 0;
while ⌊i⌋ ≠ 0 do
     z ← z + x;
     i ← i - 1;
endwhile;
output z
```

Vzhledem ke kterým z následujících vstupních podmínek je algoritmus konvergentní?

a) $\varphi(x) \equiv x \in \mathbb{R}$
b) $\varphi(x) \equiv x \in \mathbb{R}^+$
c) $\varphi(x) \equiv x \in \mathbb{Z}$
d) $\varphi(x) \equiv x \in \mathbb{N}$

Vzhledem ke kterým z uvedených vstupních podmínek a výstupní podmínce

$$\psi(x, z) \equiv z = x^2$$

je algoritmus parciálně korektní?

Najděte vstupní podmínku, pro kterou je algoritmus parciálně korektní vzhledem k libovolné výstupní podmínce.

▲ Napište definici funkce `minimum` pro nalezení nejmenšího prvku v neprázdné konečné posloupnosti čísel. Posloupnost je reprezentovaná neprázdným seznamem. Funkce bude mít jednu bázovou (nerekursivní) větev pro jednoprvkovou posloupnost a jednu rekursivní větev pro aspoň dvouprvkovou posloupnost s prvním prvkem $x$, druhým prvkem $y$ a zbytkem posloupnosti $s$.

▲ Dokažte parciální korektnost funkce `minimum` vzhledem ke vstupní podmínce
$\varphi(s) \equiv s$ *je neprázdný seznam celých čísel* a výstupní podmínce
$\psi(s, n) \equiv\ n$ *leží v* $s$ *a pro všechna* $m$ *ze seznamu* $s$ *platí* $m \geq n$.

▲ Dokažte parciální korektnost algoritmu pro seřazení prvků v dvouprvkovém poli $A$ indexovaném od jedničky vzhledem k podmínkám:
$\varphi(A) \equiv A$ *je dvouprvkové pole celých čísel*
$\psi([x, y], [p, q]) \equiv p \leq q \wedge (p, q)$ *je permutací* $(x, y)$

```
input A
    if (A[1] > A[2]) then
        z ← A[1];
        A[1] ← A[2];
        A[2] ← z;
    endif
output A
```

Jiná varianta funkce `power`:

```
power z 0  =  1
power z n  =  if  odd n  then  z * t  else  t
              where  t = power (z*z) (n ‘div‘ 2)
```

▲ Dokažte konvergenci funkce `power` vzhledem ke vstupní podmínce $\varphi((z,n)) \equiv z \in \mathbb{R} - \{0\} \wedge n \in \mathbb{N}$. Která hodnota v definici ostře klesá? Proč nemůže být záporná?

▲ Funkce `soucet` vypočte pro danou posloupnost $A_1, A_2, \ldots$ celých čísel součet $A_1 + \cdots + A_k$, v němž jsou všechny sčítance nenulové a $A_{k+1} = 0$ je prvním výskytem nuly v posloupnosti.

```
function soucet (A:posl): integer;
 var k: integer; s: integer;
 begin k := 1;  s := 0;
       while A[k] ≠ 0       /* inv */
        do begin s := s + A[k];
                 k := k + 1
           end;
       return s
 end
```

Určete invariant cyklu ve vyznačeném místě tak, aby s jeho pomocí bylo možno ze vstupní podmínky

$$\varphi(A) \;\equiv\; \forall i.\, A_i \in \mathbb{Z} \;\wedge\; \exists k \geq 1.\, A_k = 0$$

odvodit výstupní podmínku

$$\psi(A, s) \;\equiv\; \exists k > 0. \left( A_k = 0 \;\wedge\; (\forall j.\, 1 \leq j < k \Rightarrow A_j \neq 0) \;\wedge\; s = \sum_{i=1}^{k} A_i \right)$$

□ Následující algoritmus seřadí číselnou posloupnost $a = (a_1, \ldots, a_n)$ uloženou v poli $A$ vzestupně. Tedy na konci výpočtu bude v poli $A$ posloupnost $a' = (a'_1, \ldots, a'_n)$, která je permutací posloupnosti $a$ a platí $a'_1 \leq \cdots \leq a'_n$

```
for i ← [2 .. n] do
begin
   x := A_i;   j := i − 1;
   while (j > 0) && (A_j > x) do
   begin
       A_{j+1} := A_j;
       j := j − 1
   end;
   A_{j+1} := x
end
```

Formulujte vstupní a výstupní podmínky a nalezněte invarianty pro vnější a vnitřní cyklus.

☐ Nenulový reálný polynom je zadán neprázdnou posloupností koeficientů $p = [a_0, a_1, a_2, \ldots, a_n]$. Napište funkci $h$ se dvěma parametry, $p, x$, která vypočte funkční hodnotu polynomu $p$ v bodě $x$.

☐ Stanovte vstupní a výstupní podmínky a dokažte konvergenci a správnost funkce $h$.