Зад. 1

```
1. Alg1(n)
2.      s←1
3.      for i←1 to n
4.              s←s*2
5.              i←i+1
6.      return s
```

Alg1(n) = 2^n
Инварианта: При всяко k-то достигане на ред 3. имаме, че $s = 2^{k-1}$ (k броим от 1), i=k

База: при к=1-во влизане
$s = 1 = 2^0 = 2^{1-1} = 2^{k-1}$
check for i

Поддръжка:
Нека е вярно за някое k-то влизане, което не е последното... т.е имаме, че $s = 2^{k-1}$.
Сега се изпълнява тялото на цикъла.. т.е ред 4. и ред 5. Единствено ред 4. ни променя
$s_{new} = s_{old} * 2 = 2^{k-1} * 2 = 2^k$
check for i

Терминация:
При k=(n+1)-то достигане на ред 3. тялото на цикъла няма да се изпълни.. от поддръжката имаме, че $s = 2^{n+1-1} = 2^n$.
Директно след for-а имаме return s = return 2^n т.е на изход получихме точно 2^n

Зад. 2

```
1. Alg2(A[1..n])
2.      s←0
3.      for i←1 to n
4.              s←s+A[i]
5.              i←i+1
6.      return s
```

Alg2(A) събира всички числа в масива
Инварианта: На всяко k-то достигане на ред 3. в s=$\sum_{j=1}^{k-1} A[j]$ и i=k

База: При k=1-во влизане
0=s=$\sum_{j=1}^{0} A[j]$=0
i=1=k

Поддръжка:
Нека е вярно за някое k-то влизане, което не е последното. Ще докажем, че е изпълнено за (k+1)-то влизане.
От предположението имаме, че $s_{old} = \sum_{j=1}^{k-1} A[j]$ и i=k. В тялото на цикъла само 4. ред изменя s.

Тоест $s_{new} = s_{old} + A[i] = s_{old} + A[k] = \sum_{j=1}^{k-1} A[j] + A[k] = \sum_{j=1}^{k+1-1} A[j]$. В тялото на цикъла само 5. ред изменя i. Тоест $i_{new} = i_{old} + 1 = k + 1$

Терминация:
От инвариантата знаем, че при достигане на ред 3. за k=(n+1)-ви път имаме s=$\sum_{j=1}^{n+1-1} A[j]$. Директно след for-а имаме return s = return $\sum_{j=1}^{n+1-1} A[j]$. Т.е програмата ни връща сумата на всички елементи на масива.

Зад. 3

```
1. Alg3(a, n)
2.      if n=0 then
3.              return 1
4.      if n is even then
5.              return Alg3(a*a, n/2)
6.      return a*Alg3(a, n-1)
```

Alg3(a, n) = a^n, $n \in \mathbb{N}_0$ & $a \in \mathbb{R}$ & $a^2 + n^2 \neq 0$ //$\theta(\log(n))$

База: n=0
Тогава при изпълнението на Alg3, if-а на ред 2., условието му ще е TRUE т.е ще се изпълни тялото на if-а и по-конкретно ред 3. Той връща директно $1 = a^0$.

Инд. предположение:
Нека ∀m≤n е изпълнено, че Alg3(a, m) = a^m

Инд. стъпка:
Ще док., че е вярно за n+1.
Имаме 2 случая:
1. (n+1)≡0 (mod 2)
    1.1 n=0 ...
    1.2 n>0
        Тогава ще влезем в тялото на if-а от ред 4. и по-конкретно return Alg3(a*a, n/2).
        Имаме Alg3(a, n+1) = Alg3(a*a, (n+1)/2) =ИП= $(a*a)^{(n+1)/2} = (a^2)^{(n+1)/2} = a^{\frac{2(n+1)}{2}} = a^{n+1}$
2. (n+1)≡1 (mod 2)
    Тогава няма да влезем нито в тялото на if-а от ред 2. нито в тялото на if-а от ред 4. и значи се изпълнява ред 6.
    Имаме Alg3(a, n+1) = a*Alg3(a, n+1-1) =ИП= $a*a^n = a^{n+1}$

Зад. 4
Дадена е кутия с 53 сини топки и 42 червени топки. Извън кутията имаме неограничен брой сини и червени топки.

```
1. Alg4()
2.      while “не остане 1 топка в кутията” do
3.              “извади 2 случайни топки от кутията”
4.              “ако 2-те топки са еднакъв цвят, добави 1 червена, иначе добави 1 синя”
```

Какъв е цвета на топката, която е останала самичка в кутия в края?

Инварианта: На k-тото достигане на ред 2. имаме: броя топки в кутията е 95-k+1 и имаме нечетен брой сини топки в кутията.

Зад. 5

```
1. Alg5(A[1..n]) // Kadane's Algorithm
2.      c←A[1]
3.      m←A[1]
4.      for i←2 to n
5.              if A[i]+c > A[i] then
6.                      c←c+A[i]
7.              else
8.                      c←A[i]
9.              if c > m then
10.                     m←c
11.             i←i+1
12.     return m
```

Намира най-голята сума на непразен подмасив на масива A[1..n].
Инварианта: На k-тото достигане на ред 4. имаме: i=k+1, m е най-голямата сума на подмасив на A[1..i-1], c е най-голямата сума на подмасив на A[1..i-1], завършваща в индекс i.

База: На k=1-во влизане имаме:
i=2=k+1, c, m - изпълняват усл. на инвариантата

Поддръжка:
Нека е изпълнено за някое k-то (непоследно) достигане на ред 4. т.е имаме: i=k+1, c, m - ...
Ще докажем за (k+1)-то достигане на ред 4.
От доп. зн., че c_{old} е най-голямата сума на подмасив, завършващ в A[k]. Твърдим, че c_{new} е най-голямата сума на подмасив, завършващ в A[k+1]. От ред 5-8 имаме
$c_{new} = \max(c_{old} + A[k+1], A[k+1])$.. нека допуснем, че има по-голяма сума t+A[k+1], $t \in \mathbb{N}^+$. Т.е допуснахме, че $t + A[k+1] > c_{new}$ т.е $t + A[k+1] > \max(c_{old} + A[k+1], A[k+1])$ т.е
$t + A[k+1] > c_{old} + A[k+1]$ т.е $t > c_{old}$, но тогава t има семантика на най-голяма сума на подмасив на A[1..k], завършващ в индекс k, но ние имахме, че c_{old} е максималната сума.. противоречие.
Т.е c_{new} е ок. От ред 9 и 10 и допускането може да видим, че и изискването за m е изпълнено.

Терминация:
Когато k=n е 1вия път в който не се изпълнява тялото на for-а, т.е се изпълнява ред 12. return m, което е точно каквото искахме да док.