CSE 307 – Assignment # 2

Assignment problem # 1 (marks: UG: 5, PG: 4)

Write a function:

```
def solution(A)
```

that, given an array A of N integers, returns the smallest positive integer (greater than 0) that does not occur in A.

For example, given A = [1, 3, 6, 4, 1, 2], the function should return 5.

Given A = [1, 2, 3], the function should return 4.

Given A = [−1, −3], the function should return 1.

Write an efficient algorithm for the following assumptions:

- N is an integer within the range [1..100,000];
- each element of array A is an integer within the range [−1,000,000..1,000,000].

Complexity:

- Expected worst-case time complexity is O(N)
- Expected worst-case space complexity is O(N), beyond input storage (not counting the storage required for input arguments).
- Elements of input arrays can be modified.

Assignment problem # 2 (marks: UG: 5, PG: 4)

Consider the following available coin denominations of ₩:

10, 50, 100, 500.

Find the minimum number of coins with which a given amount of money can be paid.

E.g. 470 ₩ can be paid using 4 coins of 100 ₩ each, 1 coin of 50 ₩ and 2 coins of 20 ₩.

Assignment problem # 3 (marks: UG: 5, PG: 4)

Given two sets, write a Python program to create an intersection of sets, union of sets, set difference and symmetric difference.

Assignment problem # 4 (marks: UG: 10, PG: 9)

Write a lexical analyzer in Python that accepts the following language and generates a report:

Tokens	Tokens
SEMICOLON	POWER

PRINT	ASSIGN
ID	INT_CONST
PLUS	REAL_CONST
MINUS	STRING
TIMES	LPAREN
DIV	

A Sample program:

```
a% = 5.5;
b# = 3;
c% = a% + b#;
PRINT "the result is";
PRINT c%;
```

Sample Report:

```
ID a REAL
ASSIGN
REAL_CONST 5.5
SEMICOLON
ID b INTEGER
ASSIGN
INT_CONST 3
SEMICOLON
ID c REAL
ASSIGN
ID a REAL
PLUS
ID b INTEGER
SEMICOLON
PRINT
STRING the result is
SEMICOLON
PRINT
ID c REAL
SEMICOLON
```

Assignment problem # 5 (Mandatory for graduate students) (Marks: PG 4)

A zero-indexed array A consisting of N integers is given. It contains daily prices of a stock share for a period of N consecutive days. If a single share was bought on day P and sold on day Q, where $0 \leq P \leq Q < N$, then the profit of such transaction

is equal to A[Q] − A[P], provided that A[Q] ≥ A[P]. Otherwise, the transaction brings loss of A[P] − A[Q].

For example, consider the following array A consisting of six elements such that:

A[0] = 23171

A[1] = 21011

A[2] = 21123

A[3] = 21366

A[4] = 21013

A[5] = 21367

If a share was bought on day 0 and sold on day 2, a loss of 2048 would occur because A[2] − A[0] = 21123 − 23171 = −2048. If a share was bought on day 4 and sold on day 5, a profit of 354 would occur because A[5] − A[4] = 21367 − 21013 = 354. Maximum possible profit was 356. It would occur if a share was bought on day 1 and sold on day 5.

Write a function,

```
def solution(A);
```

that, given a zero-indexed array A consisting of N integers containing daily prices of a stock share for a period of N consecutive days, returns the maximum possible profit from one transaction during this period. The function should return 0 if it was impossible to gain any profit.

Assume that:

- N is an integer within the range [0..400,000];
- each element of array A is an integer within the range [0..200,000].

Complexity:

- Expected worst-case time complexity is O(N);
- Expected worst-case space complexity is O(1), beyond input storage (not counting the storage required for input arguments).
- Elements of input arrays can be modified.