# Evolutionary Computation, 2018/2019
# Programming assignment 2

## Important information

- Deadline: 15/Oct/2018, 23:59.
- All problems must be submitted through Mooshak.
- Please go to `http://mooshak.deei.fct.ualg.pt/~mooshak/` and register in 'EC1819_P2' from the 'contest' list box.
  - You can submit programs in C, C++, Java, or Python.

## About this assignment

The purpose of this programming assignment is to get you involved in programming more components of a simple genetic algorithm.

Similarly to programming assignment 1, you will be given pseudo-random numbers uniformly generated in [0..1), so that the output of your programs can be checked for correctness.

# Problem A: Roulette wheel selection

Implement *roulette wheel* as explained in class. Given a population of $N$ strings with the corresponding fitness values, you are supposed to spin the roulette $N$ times. In order to do so, you will be given $N$ random numbers.

## Input

The input has $2 + 2N$ lines. The first line contains an integer $N$ that specifies the size of the population. The second line contains an integer $\ell$ that specifies the string length. Each of the next $N$ lines contain a binary string of length $\ell$, followed by a white space, followed by the string's fitness value (a real number). Then, each of the next $N$ lines contains a pseudo random number uniformly generated in [0..1).

For the purpose if this exercise, you can assume that both $N$ and $\ell$ are less or equal than 100.

## Output

The output should be the population of selected strings. You should send them to the output in the same order as they were selected.

## Example

### Sample Input

```
4
3
110 1.0
001 3.0
101 4.0
111 2.0
0.16758
0.54902
0.97234
0.25456
```

### Sample Output

```
001
101
111
001
```

# Problem B: Stochastic universal sampling (SUS)

Implement *stochastic universal sampling* (SUS) as explained in class. Given a population of $N$ strings with the corresponding fitness values, you are supposed to spin the roulette only once. Therefore, you will only be given a single random number.

## Input

The input has $2 + N + 1$ lines. The first line contains an integer $N$ that specifies the size of the population. The second line contains an integer $\ell$ that specifies the string length. Each of the next $N$ lines contain a binary string of length $\ell$, followed by a white space, followed by the string's fitness value (a real number). Then, the next line contains a pseudo random number uniformly generated in [0..1). Important: you should map this number into the interval $[0..1/N)$.

For the purpose if this exercise, you can assume that both $N$ and $\ell$ are less or equal than 100.

## Output

The output should be the population of selected strings. You should send them to the output in the same order as they appear when scanning the roulette from left to right, i.e., from the 1st to the last individual of the population.

## Example

### Sample Input

```
4
3
110 1.0
001 3.0
101 4.0
111 2.0
0.66758
```

### Sample Output

```
001
101
101
111
```

# Problem C: one-point crossover

Implement one-point crossover.

## Input

The input has 4 lines. The first line contains an integer $\ell$ that specifies the string length. The second line contains parent1, a string of length $\ell$. The third line contains parent2, another string of length $\ell$. The fourth line contains a pseudo random number uniformly generated in [0..1). You should map this number into one of the possible $\ell - 1$ cross points, and do so from left to right along the string positions.

You can assume $\ell \leq 1000$.

## Output

The output consists of 2 lines: child1 and child2, each followed by a newline character.

## Example

### Sample Input

```
8
11111111
00000000
0.732
```

### Sample Output

```
11111100
00000011
```

# Problem D: uniform crossover

Implement uniform crossover.

## Input

The input has several lines. The first line contains an integer $\ell$ that specifies the string length. The second line contains parent1, a string of length $\ell$. The third line contains parent2, another string of length $\ell$. Each of the next $\ell$ lines contains a pseudo random number uniformly generated in [0..1). You should map each of these numbers into a fair coin toss. Let $r$ be one such number. For the purpose of this exercise, assume that $r < 0.5$ corresponds to 'head' and $r \geq 0.5$ corresponds to 'tail'. Whenever the outcome of the coin toss is 'head' you should exchange the corresponding gene values of the parents.

You can assume $\ell \leq 1000$.

## Output

The output consists of 2 lines: child1 and child2, each followed by a newline character.

## Example

### Sample Input

```
8
11111111
00000000
0.732
0.343
0.111
0.345
0.568
0.783
0.983
0.004
```

### Sample Output

```
10001110
01110001
```

# Problem E: bit-flip mutation

Implement bit-flip mutation on a binary string of length $\ell$.

## Input

The input has several lines. The first line contains the mutation probability. The second line contains an integer $\ell$ that specifies the string length. The third line contains an individual, a binary string of length $\ell$. Each of the next $\ell$ lines contains a pseudo random number uniformly generated in [0..1). You should map each of these numbers into a biased coin toss. Let $r$ be one such number and $p_m$ be the mutation probability. If $r < p_m$ then the outcome of the coin toss is 'head', otherwise the outcome is 'tail'.

You can assume $\ell \leq 1000$.

## Output

The output consists of the mutated string followed by a newline character.

## Example

### Sample Input

```
0.2
8
11111111
0.732
0.343
0.111
0.345
0.568
0.783
0.983
0.004
```

### Sample Output

```
11011110
```