0   panamabananas
1   anamabananas
2   namabananas
3   amabananas
4   mabananas
5   abananas
6   bananas
7   ananas
8   nanas
9   anas
10   nas
11   as
12   s

Q2   Binary search algorithm

5 abananas   Min   0

3   1

1   2

7   3

9   4

11   5

6   step2   mid   6

4   step4   Min   7 .

2   namabananas step2 mid   8

8   nanas   step step2 mid min mid   9

10   step5   max   10 .

0

12   Max   12

S.nana

$\frac{19}{2} = 9.5$

$\frac{17}{2} = 8.5$

5 abananas	Min	0
3		1
1		2
7		3
9		4
11		5
6		6 qoz mid
4		7 . min
2 nanabananas		8
8 nanas		9 step2 mid
10		10 .
0		
12	Max	12

Part 1: Probability of Collision

```cpp
#ifndef    BLOOMFILTER_H
#define    BLOOMFILTER-H
#include   <string>
#include   <vector>
using    namespace

/**
 * Class to  implement a Bloom  Filter that stores strings
 */
class

    private:
        /**
         * Number of  hash functions to  use (must be in
         *  range [1,5])
         */
        const  unsigned int


        /**
         * Length of the bitset (m)
         */
        const  unsigned int
```

```cpp
    /**
     *  Bitset (boolean vector) of length m
     */
        bool

public:
    /**
     * BloomFilter constructor
     */
                const unsigned int &      const unsigned
            int &                              false

    /**
     * Insert  string s into this BloomFilter
     */
    void           const            &

    /**
     * Return false if s definitely doesn't exist in this
     * BloomFilter,
     * Or true  it MIGHT exist in this BloomFilter
     */
    bool           const         &


#endif
```

```cpp
#ifndef COUNTMINSKETCH_H
#define COUNTMINSKETCH_H
#include <string>
#include <vector>
using namespace

/**
 * Class to implement a CountMinSketch that stores
 * strings
 */
class
    private:
        /**
         * Number of hash functions to use (must be in
         * range 1,5])
         */
        const unsigned int

        /**
         * Length of the bitset (m)
         */
        const unsigned int

        /**
         * Count Matrix with k rows and m columns
         */
```

unsigned int

```cpp
public:
    /**
     * CountMinSketch  constructor
     */

                        const unsigned int &
                        const unsigned int &

                                unsigned int



    /**
     * Increment the count of s
     */
    void                const            &


#endif
```

```cpp
#ifndef   HASHFUNCTIONS_H
#define   HASHFUNCTIONS_H
#include <functional>
#include <vector>
using namespace
extern const                    unsigned int const
```