Indian Statistical Institute
Semester-I 2017-2018
M.Tech.(CS) - First Year
Lab Test II (15 September, 2017)
Subject: Data and File Structures Laboratory
Total: 75 marks Duration: 3.5 hrs.

INSTRUCTIONS

1. Naming convention for your programs: cs17xx-test2-progy.c
2. When you have finished, copy all your files to ∼dfslab/2017/labtest2/cs17xx/.
3. You are allowed to use your stack / queue implementation codes developed earlier.
4. The marks you obtain for a problem depends on the efficiency of your algorithm.

1. *Push or Pop.* Given a sequence of -1, $+1$, you need to test whether the sum of all the elements in the sequence is 0 or not. But, the machine has no arithmetic operation facility only it can increment/decrement the stack pointer using *push* and *pop*. For an empty stack if you perform pop, it will get a signal indicating empty stack. [10]

   Input Format

   The input will contain a single line of sequence of integers -1 and $+1$. The end of the sequence will be denoted by a 0.

   Output Format

   The output will print either YES or NO in a new line, depending on whether the sequence sums to 0 or not respectively.

   Sample Input 0

   -1 1 1 1 -1 -1 -1 0

   Sample Output 0

   NO

   Sample Input 1

   1 1 1 1 -1 -1 -1 -1 0

   Sample Output 1

   YES

   Sample Input 2

   0

   Sample Output 2

   EMPTY STACK

2. *Queries with Fixed Length.* Consider an n-integer sequence, $A = [a_0, a_1, \ldots, a_{n-1}]$. We perform a query on A by using an integer, d, to calculate the result of the following expression:

$$\min_{0 \le i \le n-d} (\max_{i \le j < i+j} a_j)$$

   In other words, if we let $m_i = \max(a_i, a_{i+1}, a_{i+2}, \ldots, a_{i+d-1})$, then you need to calculate

$$\min(m_0, m_1, \ldots, m_{n-d})$$

   .

   Given A and an integer, d, print the result in a new line. [15]

Input Format

The first line consists of two space-separated integers describing the respective values of n and d. The second line consists of n space-separated integers describing the respective values of $a_0, a_1, \ldots, a_{n-1}$.

Output Format

Print an integer denoting the query's answer on a new line.

Sample Input 0

```
5 5
33 11 44 11 55
```

Sample Output 0

```
55
```

Explanation 0

For $d = 5$, the answer is :

$$\min(\max(a_0, a_1, a_2, a_3, a_4))) = 55$$

Sample Input 1

```
5 1
33 11 44 11 55
```

Sample Output 1

```
11
```

Explanation 1

For $d = 1$, the answer is :

$$\min(\max(a_0), \max(a_1), \max(a_2), \max(a_3), \max(a_4)) = 11$$

Sample Input 2

```
5 3
1 2 3 4 5
```

Sample Output 2

```
3
```

Explanation 2

The "prefix" has the least maximum value among the consecutive sub-sequences of the same size.

Hint: It may help if you use queues.

3. *Set operations.* Consider the problem of creating the union and intersection of two sets A and B containing integers. Implement these set operations using linked lists. You may use any of the linked list implementations described in class. [10]

Input Format

The first line will consist of space separated integers representing the sequence of numbers for the set A and the next line for the set B.

Output Format

The first line should print the union of A and B and the second line should print the intersection of A and B.

Sample Input 0

```
1 2 3 6 7
2 3 6 7
```

Sample Output 0

1 2 3 6 7
2 3 6 7

Sample Input 1

91 21 35 86 77
12 13 16 77

Sample Output 1

12 13 16 21 35 77 86 91
77

4. Consider a positive integer M and the set $A = \{0,1,2,3,...,M-1\}$, the set of all remainders modulo M. We start with a random element x_0 of A. Subsequently, for i = 1,2,3,..., we generate the next element x_i = f(x_i - 1), where f(x) = (x^2 + 1) % M. Since the elements of the sequence $x_0, x_1, x_2 \ldots$ are from the finite set A, there must be a match $x_i = x_j$ after finitely many iterations. After that, the sequence is periodic, since every element of the sequence is uniquely determined only by the previous element. To sum up, the sequence $x_0, x_1, x_2, \ldots$ looks like the Greek letter ρ, as shown in Figure 1. The first three elements (11, 22 and 35) constitute the tail of the rho. After that, there is a cycle of length 6.

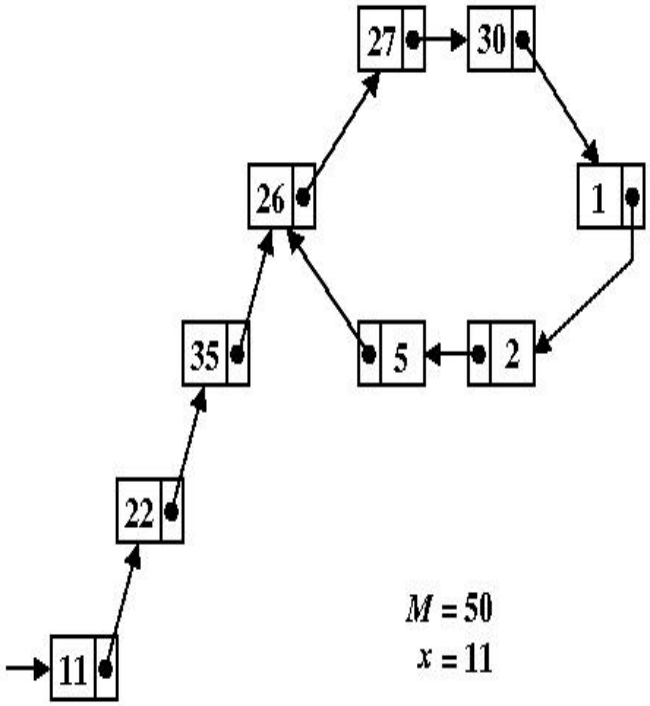

Figure 1: A ρ-like linked list

A. Write a function *genrho*() with the following prototype:

```
node *genrho ( int M , int x );
```

The function accepts the modulus M and the initial value x_0. It creates a ρ-like list using the function f(x) described above, and returns a pointer to the header node of the list. Do not use a dummy node at the beginning of the list. Assume the values of M and x are such that no overflow happens while computing f. [5]

B. Write a function *cyclelen*() with the following prototype:

```
int cyclelen ( node *head );
```

The function accepts, as its only parameter, a pointer to the header of a ρ-like list and returns the length of the cycle of the ρ. It is important to mention that the number of nodes or any such auxiliary information must not be generated during the creation of the list. The function *cyclelen* only assumes that the header pointer points to a valid rho-like list.

In order to detect the cycle length, use the following algorithm. Initialize two pointers p and q to the header pointer. Subsequently, enter a loop in which p advances by one node in the list and q by two nodes. Eventually, the two pointers must meet at some node in the cyclic part of the rho. Once a node in the cycle is detected, make a complete traversal along the cycle in order to determine its length. [15]

Report the output of your program for the following values.

```
M = 100, x = 5.
M = 6543, x = 3456.
M = 35791, x = 13579;
```

A sample main() function is given below. For simplicity, you are not required to free the memory allocated to a list, before creating the next list with different parameter values.

```
int main ()
{
  printf("Run 1: M =   100, x =     5, cycle length = %d\n\n", cyclelen(genrho(100,5)));
  printf("Run 2: M =  6543, x =  3456, cycle length = %d\n\n", cyclelen(genrho(6543,3456)));
  printf("Run 3: M = 35791, x = 13579, cycle length = %d\n\n", cyclelen(genrho(35791,13579)));
  exit(0);
}
```

Here is a sample output for M = 50 and x = 11.

```
 11: Inserted... continuing...
 22: Inserted... continuing...
 35: Inserted... continuing...
 26: Inserted... continuing...
 27: Inserted... continuing...
 30: Inserted... continuing...
  1: Inserted... continuing...
  2: Inserted... continuing...
  5: Inserted... continuing...
 26: Cycle detected... breaking...
 M = 50, x = 11, cycle length = 6
```

5. You are given an array of digits (0-9). Find the largest multiple of 3 that can be formed using these array elements. For example, if the input array is {8, 1, 9}, the output should be 981, and if the input array is {8, 1, 7, 6, 0}, output should be 8760. [20]