Name: Section: Rollno:

ESC101: Fundamentals of Computing(Lab Exam 1)

13th February, 2014

Total Number of Pages: 8

Total Points 100

Question	Points	Score
1	30	
2	30	
3	30	
4	10	
Total:	100	

Instructions

1. Read these instructions carefully.
2. **DO NOT** use the `math.h` library for any of the questions.
3. Create a folder in your home directory where you will write your programs.
4. The folder MUST BE named based on YOUR section no, roll no and userid, as follows:
   if you are in section **A1**, your roll no is **13999** and your IITK id is **rtewari**, then your folder name will be
   `a1-13999-rtewari`
5. For every program that you write, use the following naming convention:
   `section-roll-userid-q number.c`

   For example, if your IITK id is **rtewari**, your roll no is **13999** and you are in section **A1**, then your program name for
   Question 1 should be `a1-13999-rtewari-q1.c`
   Question 2 should be `a1-13999-rtewari-q2.c`
   Question 3 should be `a1-13999-rtewari-q3.c`
   Question 4 should be `a1-13999-rtewari-q4.c`
6. Every program should complete its execution within 5 seconds, on every input. If it takes longer time then marks will be deducted even if your answer is correct.
7. Range of `int` is -2147483648 to 2147483647.
8. Apart from the given test cases, you should check the correctness of your program on your own test cases as well.

Helpful hints

1. The questions are arranged according to the increasing order of difficulty.

Question 1. (30 points) **Sum of absolute deviations**

The sum of absolute deviations of a set of numbers $X = \{x_1, x_2, \ldots, x_n\}$ is defined as

$$\sum_{i=1}^{n} |x_i - \text{avg}(X)|, \qquad \text{where } \text{avg}(X) = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Write a program to:

1. Take the size of the set "n" as input from the user (assume $n \leq 100$)
2. Take n numbers as input (each of type `int`)
3. Print the sum of absolute deviation of the numbers

Note: The input will always be such that the avg(X) is an integer.

Here are some sample interactions of the program:

```
$./a.out
Enter n: 3
Enter the number: 1
Enter the number: 2
Enter the number: 3
Sum of absolute deviations: 2
```

```
$./a.out
Enter n: 5
Enter the number: 1
Enter the number: 2
Enter the number: 13
Enter the number: 7
Enter the number: 7
Sum of absolute deviations: 18
```

Table 1: **Test Cases**

Value of `n`	`n` integers	*Expected Output*
8	2 5 4 2 5 4 7 3	10
14	16 18 13 17 19 18 0 0 19 17 9 1 10 11	82
20	-5 -2 -2 -4 2 3 -2 3 -5 0 3 4 4 2 -3 -2 -1 -1 -4 5	57
25	8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	0
30	0 0 7 7 0 7 0 0 0 7 7 0 0 7 7 7 0 0 7 0 7 7 7 0 0 0 7 0 7 0	104
30	-193 249 -927 658 -70 272 544 -122 923 709 -560 -835 -508 42 987 -497 -673 729 -160 -388 -697 169 709 157 560 -67 99 -722 816 326	14284

Question 2. (30 points) **Smallest lexicographical rotation of an array**

Consider two integer sequences having the same length

$$\begin{aligned} A &= (a_1, a_2, a_3, \ldots, a_n) \\ B &= (b_1, b_2, b_3, \ldots, b_n) \end{aligned}$$

We will say that $A < B$ if there is an integer i ($0 \leq i \leq n-1$) such that:

1. a_1=b_1, a_2=b_2, a_3=b_3, $\ldots$, a_i=b_i
2. $a_{i+1} < b_{i+1}$

For example,

(3, 2, 5)	<	(3, 2, 10)
(5, 16, 14, 12)	<	(7, 6, 4, 2)
(10, 8, 13, 17, 3)	<	(10, 9, -7, 6, 5)
(12, 11, 15, 14, 9, 7, 5)	<	(12, 11, 15, 14, 9, 7, 6)

This ordering on sequences is called *lexicographical ordering.*

Given an array of size n, there are n *'rotations'* of the array (shown below).

For example, the array (3, 20, 3, 15) has 4 rotations as follows:

1. (3, 20, 3, 15)
2. (20, 3, 15, 3)
3. (3, 15, 3, 20)
4. (15, 3, 20, 3)

The rotation which is the *smallest lexicographical rotation* amongst all the other rotations is (3, 15, 3, 20).

Note that an array can have more than 1 rotations, that are lexicographically smallest (eg. (6, 6, 6)).

Write a program that does the following:

1. Take "n" as input (assume $n \leq 100$)
2. Reads n numbers as input, into an array A (of type `int`)
3. Prints the smallest lexicographical rotation of the array A.

Here are some sample interactions of the program:

```
$./a.out
Enter n: 3
Enter the number: 1
Enter the number: 2
Enter the number: 3
Smallest lexicographic rotation: 1 2 3
```

```
$./a.out
Enter n: 3
Enter the number: 2
Enter the number: -1
Enter the number: 8
Smallest lexicographic rotation: -1 8 2
```

```
$./a.out
Enter n: 5
Enter the number: 10
Enter the number: 21
Enter the number: 10
Enter the number: 17
Enter the number: 13
Smallest lexicographic rotation: 10 17 13 10 21
```

```
$./a.out
Enter n: 8
Enter the number: 3
Enter the number: 5
Enter the number: 8
Enter the number: 9
Enter the number: 3
Enter the number: 5
Enter the number: 8
Enter the number: 7
Smallest lexicographic rotation: 3 5 8 7 3 5 8 9
```

Table 2: **Test Cases**

Value of `n`	`n` integers	*Expected Output*
10	78 78 78 78 78 78 78 78 78 78	78 78 78 78 78 78 78 78 78 78
5	-1 -2 -3 -4 -5	-5 -1 -2 -3 -4
12	23 23 23 23 23 61 61 61 61 23 23 23	23 23 23 23 23 23 23 23 61 61 61 61
14	6 17 17 17 31 6 17 17 17 31 6 17 17 17	6 17 17 17 6 17 17 17 31 6 17 17 17 31
16	5 4 3 6 5 4 3 6 5 4 3 6 5 4 3 6	3 6 5 4 3 6 5 4 3 6 5 4 3 6 5 4
18	13 4 7 32 7 4 7 4 7 24 4 13 13 7 13 7 7 7	4 7 4 7 24 4 13 13 7 13 7 7 7 13 4 7 32 7
22	2 5 2 2 2 2 9 2 2 2 2 5 5 2 7 7 5 9 5 2 2 2	2 2 2 2 5 2 2 2 2 9 2 2 2 2 5 5 2 7 7 5 9 5
24	8 6 0 2 5 5 5 2 8 4 5 2 5 1 0 9 5 4 1 5 9 4 9 4	0 2 5 5 5 2 8 4 5 2 5 1 0 9 5 4 1 5 9 4 9 4 8 6
30	307 -251 -427 158 430 -228 44 378 423 209 -60 -335 -8 -458 487 3 -173 229 340 112 -197 -331 209 -343 60 433 -401 -222 316 -165	-458 487 3 -173 229 340 112 -197 -331 209 -343 60 433 -401 -222 316 -165 307 -251 -427 158 430 -228 44 378 423 209 -60 -335 -8

Question 3. (30 points) **Sphenic triplets**

A *sphenic number* is defined as a number of the form $p \times q \times r$, where p, q and r are all distinct primes.

For example, $30 = 2 \times 3 \times 5$, $190 = 2 \times 5 \times 19$ and $1001 = 7 \times 11 \times 13$ are all sphenic numbers.

On the other hand, $20 = 2 \times 2 \times 5$ and $90 = 2 \times 3^2 \times 5$ are NOT sphenic numbers.

A *sphenic triplet* is defined as a triplet of three consecutive numbers $s-1$, s and $s+1$, such that all three of them are sphenic numbers. For example, $\{1309, 1310, 1311\}$, $\{1885, 1886, 1887\}$ and $\{2013, 2014, 2015\}$ are all sphenic triplets.

Write a program to:

1. Take a positive integer "n" as input (assume $n \leq 10^6$)
2. Print the largest $s \leq n$ such that $\{s-1, s, s+1\}$ is a sphenic triplet, if there exists such an s, otherwise print "`No such number`".

Here are a few sample interactions of the program:

```
$./a.out
Enter the number: 1310
The value of s is 1310

$./a.out
Enter the number: 2500
The value of s is 2014

$./a.out
Enter the number: 45000
The value of s is 44914

$./a.out
Enter the number: 450
No such number
```

Table 3: **Test Cases**

Value of `n`	*Expected Output*
30000	29954
1000000	999986
999900	999786
100001	99506
1600	1310
1000	No such number
9991	9878
40	No such number
79885	79646
199838	199838
499919	499918

Question 4. (10 points) **Country Roads** (*This is a tough problem*)

There is a country with N cities and M roads connecting its various cities such that there is no way to travel from one city of the country to another, using two different sequence of roads. More formally, the structure of the country's road network is defined as follows:

1. The cities are numbered from 1 to N, with each city having a unique number.
2. There are M roads, where each road connects a pair of cities and $M < N$ (obviously both the cities connected by a road are different). Also every road has some positive length associated with it.
3. Every road can be traversed in both directions.
4. A *path* is sequence of roads connecting two cities such that every road can appear at most once in a path.
5. Between every pair of cities, there is at most one path.

Write a program to do the following:

1. Read an integer N (the number of cities in the country, $N \leq 100$) from the input.
2. Read an integer M (the number of roads in the country) from the input.
3. Read M triplets of integers in the form u, v, d, meaning that there is a road between city u and city v, and the length of the road is d (both u and v are valid city numbers and d is of type `int`).
4. Read an integer Q (the number of queries, $Q \leq 100$) from the input.
5. Read Q lines, each containing a pair of integers a and b (both a and b are valid city numbers).
6. For each query, print the length of the path from a to b if there exists one, otherwise print "No Path" in a separate line.

Look at the following sample interactions and their corresponding figures:

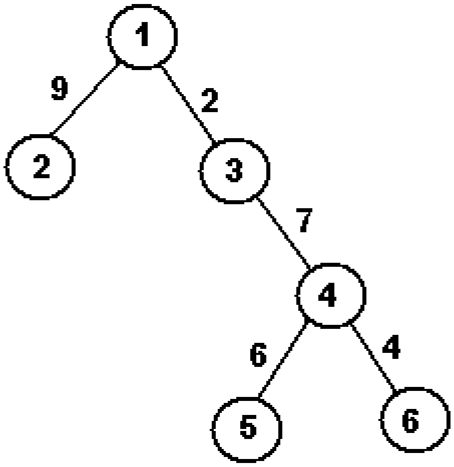

Figure 1: Sample Interaction 1

```
$./a.out
Enter N: 6
Enter M: 5
Enter road 1: 1 3 2
Enter road 2: 4 6 4
Enter road 3: 4 5 6
Enter road 4: 1 2 9
Enter road 5: 4 3 7
Enter Q: 4
Enter query 1: 2 5
Path length: 24
Enter query 2: 1 4
Path length: 9
Enter query 3: 3 5
Path length: 13
Enter query 4: 6 5
Path length: 10
```

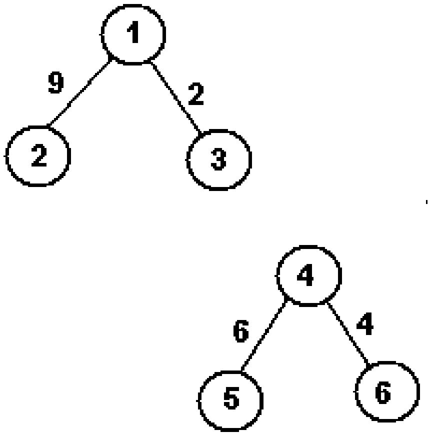

Figure 2: Sample Interaction 2

```
$./a.out
Enter N: 6
Enter M: 4
Enter road 1: 1 3 2
Enter road 2: 4 6 4
Enter road 3: 4 5 6
Enter road 4: 1 2 9
Enter Q: 4
Enter query 1: 2 5
Path length: No Path
Enter query 2: 1 4
Path length: No Path
Enter query 3: 3 2
Path length: 11
Enter query 4: 6 5
Path length: 10
```

Table 4: **Test Cases**

Value of `N`	Value of `M`	`M` triplets	Value of Q	Value of `a`	Value of `b`	*Expected Output*
7	6	1 2 1	6	4	6	4
		1 3 1		5	6	4
		4 2 1		1	7	2
		5 2 1		4	3	3
		6 3 1		3	5	3
		3 7 1		2	7	3
8	6	1 3 5	8	4	6	8
		2 4 1		5	4	12
		2 6 7		1	7	7
		7 1 7		4	3	No Path
		7 8 10		3	5	No Path
		6 5 4		2	7	No Path
				3	8	22
				2	5	11
20	18	1 2 1	21	8	12	29
		1 3 2		8	13	30
		2 4 3		8	14	32
		2 5 4		8	15	33
		3 6 5		9	12	30
		3 7 6		9	13	31
		4 9 8		9	15	34
		5 10 9		10	12	32
		5 11 10		10	13	33
		6 13 12		10	15	36
		7 14 13		11	12	33
		7 15 14		11	13	34
		16 17 15		11	14	36
		16 18 16		11	15	37
		17 19 17		19	18	48
		17 20 18		20	18	49
		6 12 11		8	7	19
		4 8 7		13	5	24
				6	10	21
				2	15	23
				1	16	No Path
20	18	1 2 1	18	1	20	No Path
		2 3 2		1	19	2145
		3 4 4		1	18	2120
		4 5 8		1	17	2115
		5 6 16		1	16	1386
		6 7 32		2	16	1385
		7 8 64		2	17	2114
		8 9 128		2	18	2119
		9 10 256		2	19	2144
		10 11 512		3	19	2142
		11 12 3		3	18	2117
		12 13 9		3	17	2112
		13 14 27		3	16	1383
		14 15 81		12	19	1119
		15 16 243		15	4	1136
		16 17 729		11	5	1008
		17 18 5		19	20	No Path
		18 19 25		20	2	No Path