Assignment 3

Deadline : May 12, 2023

Instructions:

- You are expected to use ADTs or ideas from ADTs to solve these problems
- While submitting your code, please ensure that are organizing it according to the following structure:
  1. `main.c`
  2. `ADT.c` (ADT is a placeholder; for **hash**-based questions, the file should be named `hash.c`. Same idea extends for **heap** based questions)
  3. `ADT.h` (ADT is a placeholder; for **hash**-based questions, the file should be named `hash.h`. Same idea extends for **heap** based questions)
- Ensure that the `main.c` file does not contain any data structure specific routine
- Ensure that your Moodle submission and OJ submission are the same (except for the code organization part)
- Failure to comply with instructions will result in penalties

---

1 Anagrams (50 points)

You are given a dataset of N strings, where each string consists of lowercase English letters only. Your task is to find all pairs of strings in the dataset that are anagrams of the given query, and print them in the order they are inserted.

1.1 Input Format

The first line of input contains an integer N ($1 <= N <= 10^4$), Q($1 <= N <= 10^4$) denoting the number of strings in the dataset and number of queries respectively.

Each of the next N lines contains a string S of size ($1 <= |S| <= 10$), consisting of lowercase English letters only.
Each of the next Q lines contains a string S of size ($1 <= |S| <= 10$), consisting of lowercase English letters only.

1.2 Output Format

The output should contain one line for each query if there are anagrams of the given string print the anagrams in the order they were inserted into the dataset. (Note: In this case, the query need not be the first element, but rather the first element inserted is the first element. Look at sample cases for more Clarity). If the given string doesn't have any anagram in the dataset, print "-1". The strings are seperated by ' ' space, even after printing the last string print the space ' '.

1.3 Constraints

- All strings in the input dataset consist of lowercase English letters only.
- The sum of the lengths of all strings in the dataset is at most 10^5.
- The sum of the lengths of all queries in the dataset is at most 10^5.

1.4 Example Test Cases

1.4.1 Test case 1

Input

```
5 5
listen
silent
party
trapy
elbow
listen
twenty
trapy
blowy
blowe
```

Output

```
listen silent
-1
party trapy
-1
elbow
```

1.4.2 Test case 2

Input

```
6 4
cat
tac
bat
```

```
rat
ogd
god
rat
act
dog
dig
```

Input

```
rat
cat tac
ogd god
-1
```

Explanation for test case 1:
The first query is obvious listen is inserted first after that silent so listen, silent. For "twenty" there are no anagrams in the dataset hence -1. for trapy the anagrams are trapy and party but party is inserted first hence it should appear followed by trapy which are separated by space. blowy doesn't have any anagram. blowe has an anagram elbow in the dataset hence should be included in the answer.
Explanation for test case 2:
The first query is obvious only one anagram of rat is inserted so rat. For "act" there are two anagrams in the dataset hence cat, tac inserted in that order. Dog has two anagrams in the dataset ogd and dog it should be printed in that order .dig has no anagrams that are preset hence -1.

2 Palindromic substring (120 pts)

Given a string $S = s_1 s_2 \dots s_n$ of length N, and Q queries, each query having two integers l and r. Find whether the substring $s_l s_{l+1} \dots s_r$ is a Palindrome or not. Print `"YES"` if it is a palindrome and `"NO"` otherwise.

2.1 Constraints

- $1 \leq N, Q \leq 10^5$
- $1 \leq l \leq r \leq N$
- All characters are lowercase english alphabets ('a' to 'z')

2.2 Input format

The first line consists of two space separated integers N and Q, where N is the length of the string and Q is the number of queries.
The following line consists of an N-letter string S consisting of only lowercase english alphabets.
The following Q lines consist of two space separated integers l and r, which represent the left and the right indices of the substring.

```
N Q
S
l_1 r_1
l_2 r_2
.
.
.
l_Q r_Q
```

2.3 Output

For every query, print `"YES"` if the substring $s_l s_{l+1} \dots s_r$ is a palindrome, and `"NO"` otherwise. Note that the output is case sensitive, therefore the strings `"YeS"`,`"YEs"`,`"No"`,`"no"`, etc. will **not** be accepted.

2.4 Example test cases

2.4.1 Test Case 1

Input

```
5 3
abbab
1 2
2 3
3 5
```

Output

```
NO
YES
YES
```

2.4.2 Test Case 2

Input

```
6 3
abcdef
1 1
2 4
3 6
```

Output

```
YES
NO
NO
```

Explanation for test case 1: The substrings corresponding to the three queries are "ab","bb","bab" of which only the first is **NOT** a palindrome

3 The Mad Scientist (100 points)

A mad scientist is conducting an extremely delicate experiment that requires her to precisely control the temperature of the experiment. In front of her there are knobs which alter the temperature of this experiment (in Celcius). The scientist loves to turn knobs (they don't call her mad for nothing), but she is also very careful and wants to make sure that the temperature never goes below 0 degrees or else the experiment will fail. The experiment starts at 0 degrees.

There are N knobs which are numbered from 1 to N from left to right, where the ith knob controls the temperature by a value of A_i degrees. Each knob can be turned only once. She starts from the first knob and goes till the Nth knob in order and for each knob, she can either turn it once or choose to ignore it.

"Knobviously", the mad scientist wants to figure out the maximum number of knobs she can turn without the temperature ever going below 0 degrees.

3.1 Constraints

- $1 \leq N \leq 10^5$ and $-10^6 \leq A_i \leq 10^6$

3.2 Input format

The first line contains an integer N, the total number of knobs. The next line contains N space-separated Integers A_i $(1 \leq i \leq N)$, the temperature change caused by the ith knob.

```
N
A1 A2 . . . AN
```

3.3 Output format

Print the maximum number of knobs the mad scientist can turn.

3.4 Example test cases

3.4.1 Test case 1

Input

```
12
-3 -3 -7 -7 -1 -7 3 3 -2 -1 0 -7
```

Output

```
5
```

Explanation
In the final run, the mad scientist will ignore the first 6 knobs and the last knob, and the final temperature will be 3.

3.4.2 Test case 2

Input

```
8
9 -5 -3 -2 -1 7 6 -14
```

Output

```
7
```

Explanation
In the final run, the mad scientist will ignore only the second knob ($A_2 = -5$) and the final temperature will be 2.

4 Permuting numbers (50 points)

Given a sequence of numbers as an input, we can generate different permutations of the numbers in the sequence. One type is called *derangement*, which is a permutation of the elements in the sequence where no element appears in its original position. For example, for the sequence '1 2 3', one possible derangement is '2 3 1'.
For this task, you are given a number n and a n-length sequence of numbers as input. You need to find a permutation of the sequence such that no element appears in its original position and it is ordered in smallest permutation (in terms of ordering) is produced.
Note:

- The elements in the input sequence are distinct
- The rule for ordering sequences: Sequences (or sub-sequences) starting with 1 are smaller that sequences (or sub-sequences) starting with 2 and so on. That is, use the general ordering of the numbers to find the smallest possible permutation
- Sequences may not necessarily be from 1 to n

4.1 Constraints

- $1 \leq n \leq 10^6$
- $0 \leq Ai \leq 10^6$

4.2 Input

Input will contain multiple lines

```
n
A1 A2 . . . An
```

4.3 Output

Permutation of the sequence according to the conditions specified.

4.4 Example test cases

4.4.1 Test case 1

Input

```
3
3 2 1
```

Output

```
1 3 2
```

Explanation
The smallest sequence we can generate from the given input is '1 2 3'. However, '2' will be in its original position in the given string. So, the sequence must be '1 3 2'

4.5 Test case 2

Input

```
10
2 1 4 3 5 9 6 10 8 7
```

Output

```
1 2 3 4 6 5 7 8 9 10
```