Programming Lab

Practice Problem Set #1

Ashoka University

Note

These are practice problems designed to help you learn and improve your programming skills. They are not graded - focus on understanding the concepts and experimenting with different approaches. If you think you are stuck, you are encouraged to google and seek resources online!

Problem 1 - Back to the basics

How can you read input from stdin and print output to stdout in C? Given this, write a program that takes an integer n, and prints the first n natural numbers.

Example:
Input: 5
Output: 1 2 3 4 5

Problem 2

1. Maximum Finding

Given an integer n, followed by n integers $a_1, a_2, \ldots, a_n$, print the maximum a_i to stdout.

Example:
Input: 5 3 1 2 1 4
Output: 4

2. Maximum and Minimum

Given an integer n, followed by n integers $a_1, a_2, \ldots, a_n$, print the maximum and minimum a_i to stdout.

Example:
Input: 5 3 1 2 -1 4
Output: 4 -1

3. Second Largest

Given an integer n, followed by n integers $a_1, a_2, \ldots, a_n$, print the second largest a_i to stdout.

Example:
Input: 5 3 1 2 -1 4
Output: 3

4. Median

Given an integer n, followed by n integers $a_1, a_2, \ldots, a_n$, print the median of the numbers.

Example:
Input: 5 3 1 2 -1 4
Output: 2

5. k-th Largest

Given an integer n and k, followed by n integers $a_1, a_2, \ldots, a_n$, print the k-th largest a_i to stdout.

Example:
Input:
5 4
3 1 2 -1 4
Output: 1

Problem 3 - Sift up

For this, and the following questions, recall how we defined sift up and sift down on an array. If $[a_0, a_1, \ldots, a_{n-1}]$ is an array, then calling sift up on index i means:

- If $a_{(i-1)/2} > a_i$, swap them and call sift up on index $(i-1)/2$.
- Otherwise, do nothing.
- What are the base cases? Make sure to take care of them.

Given an integer n and $i < n$, followed by n integers $a_0, a_1, \ldots, a_{n-1}$ call sift up on index i and print the resulting array.

Example:
Input:
5 1
1 4 1 2 4
Output: 1 4 1 2 4

Input:
4 3
6 5 4 1
Output: 1 6 4 5

Problem 4 - Sift down

Sift down at i is defined as follows:

- If either of a_{2i+1} or a_{2i+2} are smaller than a_i, swap a_i with the smaller of the two and call sift down on that index.
- Otherwise, do nothing (if a_i is smaller than both children).
- What are the base cases? Make sure to take care of them.

Given an integer n and $i < n$, followed by n integers $a_0, a_1, \ldots, a_{n-1}$ call sift down on index i and print the resulting array.

Example:
Input:
5 1
1 4 1 2 4
Output: 1 2 1 4 4

Problem 5 - Keep on sifting

Given an integer n followed by n integers $a_0, a_1, \ldots, a_{n-1}$, call sift up on every index, starting from index $n/2$ down to index 0. Print the resulting array.

Example:
Input:
6
10 9 8 7 6 5
Output: 5 6 8 7 9 10

Input:
8
9 7 3 10 21 1 4 12
Output: 1 7 3 10 21 9 4 12

Problem 6 - Can you count?

Given a string s consisting on n characters, count the number of times each character appears in the string and print the counts. (The order of the output does not matter)

Example:
Input:
10 helloworld
Output: h:1 e:1 l:3 o:2 w:1 r:1 d:1

Problem 6 - Secret Messages

We went over Caesar cipher in class. Given a string, it shifts each letter by a fixed number of positions. For example, with a shift of 1, 'a' becomes 'b', 'b' becomes 'c', and so on.
Implement a program that takes a n character long string and an integer k and outputs the string with each letter shifted by k positions.

Remember: The goal is to practice and learn. Don't hesitate to experiment and ask questions!