

# Hackathon 3 ( Feb 9, 2022)

## General Instructions:

### Rules:

- The allowed libraries are `stdio.h` and `stdlib.h` (only for `malloc`, `calloc`, `realloc`, `free`).
- Your program should be modular. Do not write your entire program in `main`. Create suitable functions.
- For each function, leave a short comment above it describing what the function does.
- You are not allowed to use variable length arrays (VLA). All dynamic memory allocation must be on the heap.
- Your program should not have memory leaks. Free all heap memory used.
- Your program should take input till EOF. You can detect this by checking if `scanf` returned `-1`.

**Deadline:** 1800hrs on Friday 11th Feb

**Submission:** On Autojudge. Limit of 3, 3, and 4 submissions for problems 1,2, and 3 respectively.

## Problem 1

(30 marks)

**Input:**  $m, k \in \mathbb{N}$  given as two numbers separated by space, terminated by `\n`

**Output:** Compute  $m^k$  followed by `\n`

Guarantee: The input will be such that the value of  $m^k$  will fit into `unsigned long` (on the evaluation server).

*Scoring: Assuming multiplication of two numbers takes  $O(1)$  time, you must design an algorithm that has worst case running time polynomial in size of  $k$ . i.e., your algorithm should run in time  $O((\log k)^c)$  for some constant  $c$ . Max score is capped at 10 if you use an exponential time algorithm.*

*Hint:*

To compute  $m^8$ , you should do  $m \rightarrow m^2 \rightarrow m^4 \rightarrow m^8$ .

To compute  $m^{20}$ , you should multiply  $m^4$  and  $m^{16}$ .

## Problem 2

(30 marks)

**Input:** Each instance is two lines that give  $a, b \in \mathbb{Z}$  separated by `\n`.

**Goal:** Compute the sum of  $a$  and  $b$ .

**Output:** Output the value of  $a + b$  followed by `\n`.

Example:

**Input instance:**

79879847598798290384897987589792312536787138957 `\n`

897586789063852687972987643578 `\n`

**Output:**

79879847598798291282484776653645000509774782535 `\n`

**Input instance:**

```
85738921798758979873284789789298121230989080 \n
-228972198379712978891479827148979871242880000 \n
```

**Output:**

```
-143233276580953999018195037359681750011890920 \n
```

*Remark: Every 2 lines of the input forms one input instance. Process each input instance as it is received. Stop at EOF.*

## Problem 3

(40 marks)

**Input:**  $n \in \mathbb{N}$ .

**Output:** Print every way of writing  $n$  as a sum of natural numbers, with repetitions, in non-decreasing order.

**Format:**

- Each number in the sum should be separated by a '+'.  
• The numbers should be listed in non-decreasing order.  
• End each line with `\n`

**Examples:**

**Input instance:** `3 \n`

**Output:**

```
1+1+1 \n
1+2 \n
3 \n
```

**Input instance:** `5 \n`

**Output:**

```
1+1+1+1+1 \n
1+1+1+2 \n
1+1+3 \n
1+2+2 \n
1+4 \n
2+3 \n
5 \n
```

*Observation: In each line, the numbers appear in non-decreasing order.*

*Remark 1: Every line of the input forms an input instance. Process each input instance as it is received. Stop at EOF.*

*Remark 2: The evaluation server has been configured to match output lines in the order listed above. However, you are allowed to output the possibilities in any order you wish. Your program will be manually evaluated anyway.*