<Advanced C Programming and Lab>
Ch 12. Dynamic Memory Allocation

※ Note

- If not mentioned, assume that there is no additional inputs.
- If not mentioned, do not print a space in the beginning and end of each line.
- In input and output examples, after ↦ symbol is to explain the input and output.
- In output examples, □ symbol indicates a space.

Section2 [Problem 1] Receive N integers and store them in an array using dynamic memory allocation (N ≤ 20). Calculate and print the sum of the array elements.

Input Example 1	Output Example 1
6 ↦ N 3 2 0 1 4 6	16

Section2 [Problem 2] Receive N float data type real numbers and store them in an array using dynamic memory allocation (N ≤ 20). Find and print out the maximum value. (2 decimal points).

Input Example 1	Output Example 1
5 ↦ N 1.1 2.5 3.4 6.1 7.8	7.80

Section2 [Problem 3] Write a program that manages students' ID.

- Receive N (number of students). Receive N students' ID using dynamic memory allocation (int data type)
- Receive D (number of student IDs to delete; D<N). Reduce the size of the allocated memory by D.
- Delete D number of student IDs from the end of the array.
- Terminating the program, free the allocated memory.

Input Example 1	Output Example 1
3 ↦ N 16011111 16011123 16011145 2 ↦ D	16011111

Input Example 2	Output Example 2
4 ↦ N 120111 15011123 16011145 16011300 1 ↦ D	120111 15011123 16011145

Section2 [Problem 4] Receive the size of a column and row. Dynamically allocate 2-dimensional strings array. Store alphabet letters in an alphabetical order and print them tou.

- Print lower-case letters first
- Print an upper-case letter 'A' following a lower-case letter 'z'. Print a lower-case letter 'a' following an upper-case letter 'z'
- Terminating the program, free the allocated memory.

Input Example 1	Output Example 1
6 6	a b c d e f□ g h I j k l□ m n o p q r□ s t u v w x□ y z A B C D□ E F G H I J□

Input Example 2	Output Example 2
9 6	a b c d e f g h I□ j k l m n o p q r□ s t u v w x y z A□ B C D E F G H I J□ K L M N O P Q R S□ T U V W X Y Z a b□

Section2 [Problem 5] Receive an integer N. Receive a string of length N. Receive two characters. Print how many times each character appears in the string.

- N≥3
- Terminating the program, free the allocated memory

Input Example 1	Output Example 1
5 ↦ N apple a x	1 0 ↦ 'a' once, 'x' none

Input Example 2			Output Example 2		
6 people e o	↦	N	2 1	↦	'e' twice, 'o' once

Section2 [Problem 6] Receive N students' information (name, test scores – korean, english, math) and store them using dynamic memory allocation. Calculate each student's average score. Print 'GREAT' or 'BAD' based on the student's score.

- 1≤N≤50
- Print the average score using 1 decimal point
- Print 'GREAT' or 'BAD':
  - If any of korean, english, math scores ≥ 90, print 'GREAT'
  - If any of korean, english, math scores < 90, print 'BAD'
  - If you print both 'GREAT' and 'BAD' for a student, print 'GREAT' first. Print a space in between the two.
- Define and use a stdent structure.
  - name: a string without spaces (1≤length of a string≤7)
  - test scores(korean, english, math): int data type. 0≤score≤100
  - average score: float data type

Input Example 1	Output Example 1
2 Kim 100 82 34 Young 90 100 99	Kim 72.0 GREAT BAD Young 96.3 GREAT

Section3 [Problem 7] Receive an integer N. Receive N integers using dynamic memory allocation.

- Compare the first two elements and swap them if the first element is larger than the second element.
- Repeat the swap for each element (except the last element), i.e., compare ith element to i+1th element and swap them if ith element is larger than i+1th element.
- (dynamic memory allocation) print the elements in an array as stored.
- Hint: the largest integer becomes the last element.

Input Example 1	Output Example 1
5 ↦ N=5 integers 5 4 3 2 1	4 3 2 1 5

Section3 [Problem 8] Receive an integer N. Receive N characters using dynamic memory allocation. Count and print how many times "cat" appears.

Input Example 1	Output Example 1
7 ↦ N=7 characters catbcat	2

Section3 [Problem 9] Receive an integer N. Receive N strings with spaces using dynamic memory allocation. Print the shortest string with spaces..

- Maximum length of a string is 100.
- Allow to use strings standard library

Input Example 1	Output Example 1
4 ↦ N=4 strings Program Good This is string language	Good

Section3 [Problem 10] Receive an integer N. Receive N strings with spaces using dynamic memory allocation. Sort and print the strings in descending order by the length of the strings.

- Maximum length of a string is 100.
- Allow to use strings standard library

Input Example 1	Output Example 1
4 ↦ N=4 strings Program Good This is string language	This is string language Program Good

Section3 [Problem 11] Keep receiving integers and store them in an array until –1 is received. The size of the array is initially set to 5. As receiving integers, increase the size by 3 if it exceeds the size of the array. When increasing the size of the array, create a new array and replace the old array. If –1 is received, print all elements in the array.

- Use dynamic memory allocation(malloc).
- The maximum size of an array is 20.

Input Example 1	Output Example 1
3 4 5 6 7 8 9 -1	□3 4 5 6 7 8 9 -1

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

Section3 [Problem 12] Receive an integer N (odd number). Dynamically allocate the space to store N integers. Store numbers from 0 to N-1 (0 1 2 ... N-1). Dynamically allocate the space to store M(=N-1) integers. Store the numbers to the new space excluding the median value among the N integers.

- Use dynamic memory allocation.
- N≤20.

Input Example 1	Output Example 1
5 ↦ N	□0 1 3 4

Input Example 2	Output Example 2
11	□0 1 2 3 4 6 7 8 9 10

Section3 [Problem 13] Receive an integer N. Receive N strings. Sort and print the N strings in lexicographic order.

- Use malloc( ).
- Use dynamic memory allocation to allocate the required space only.
- Maximum length of a string is 100.
- Assume that lower-case letters are only received.
- Allow to use strings standard libraryd

Input Example 1	Output Example 1
4 apricot peach willow birch	apricot birch peach willow