

# Task #1

A simple way to encrypt a message is to repeat characters in some way. Emily has decided to encrypt her message when she sends text to her friends. She writes the first character once, second character twice, third character three times, .., and mth character m times. For example, a message “brb” will be encrypted as “brrbbb”. Write a program to decrypt a message that is encrypted with this algorithm. It is guaranteed that the answer to the encrypted message exists.

---

## Requirements

- Assume the input length for the string is no more than 1000 characters.
- **Assume the input string contains only lowercase alphabetic letters.**
- The program should include the following function:

**void decrypt(char \*input, char \*output);**

- The function expects input contain the input string and output stores the decrypted string.
  - This function should use **pointer arithmetic**– not subscripting – to visit array elements. In other words, eliminate the loop index variables and all use of the [] operator in the function.
  - String library functions are NOT allowed.
  - main function calls the decrypt function and displays the output.
  - Follow the format of the examples below.
- 

## Examples (your program must follow this format precisely)

### Example #1

Enter message: siisss

Output: sis

### Example #2

Enter message: teemmmpppp

Output: temp

### Example #3

Enter message: npp

Output: np

## Task #2

In this program, we define an input to be valid if all of following conditions hold:

1. All alphabetic letters are lower case, like “c programming!”,
2. Non alphabetic letters can only be digits, white space, exclamation point, question mark, or period.

Write a program that prompts the user to enter input. The program determines if the input is a valid or invalid.

So that input characters are command line arguments.

---

### Requirements

- Input characters are command line argument.
- Character handling library functions in ctype.h are allowed.
- The program should also check if the number of arguments on the command line is greater than 2 (including ./a.out as one of the arguments). If the number of arguments is not greater than 2, the program should print **"invalid number of arguments"**.
- Follow the format of the examples below.

---

### Examples (your program must follow this format precisely)

#### Example #1

```
$ ./a.out spring_24
```

```
invalid
```

#### Example #2

```
$ ./a.out 8 little pigs!
```

```
valid
```

#### Example #3

```
$ ./a.out
```

```
invalid number of arguments
```