Note: You can use any function in this paper, by calling where ever you need it.

Length of the string

int string_len(char str[]) {

}

Check if a given string is binary string or not.

e.g:

Input: str = “01010101010” Output: Yes

Input: str = “Strings101” Output: No

// print Yes or No

void isBinary(char str[]) {

}

Find second maximum digit in the given number (with 2 or more digits) in the form of string.

e.g:

Input: “572” Output: 5

Input: “38754329” Output: 8

```
int second_max_digit(char number[]) {









}
```

Replace all occurrences of "---" with "RRR"

e.g:

Input: "co--de---well-"

Output: "co--deRRRwell-"

```
void replace_3hypens(char str[]) {

}
```

The given char occurs odd number of times in the given string.

Return the index of middle occurrence of the character.

e.g:

Input: 'a', "avacado"

Output: 2

```
int middle_occurance(char ch, char str[]) {









}
```

Delete the character at the given index of the string.

void delete_at_index(char str[], int index) {

}

Given two strings, delete all the characters from bigger string, which are present in smaller string.

e.g#1:

Input: "abacdef", "abxz" Output: "cdef", "abxz"

e.g#2:

Input: "xyz", “abcdefx” Ouput: "xyz", "abcdef"

Note:

(1) The 2 strings will never be of equal length.

(2) Update the bigger string only.

(3) You must use the previous function delete_at_index.

```
void remove_chars(char str1[], char str2[]) {









}
```

What is the output of following program?

```
#include<stdio.h>
void swap(char *str1, char *str2) {
  char *temp = str1;
  str1 = str2;
  str2 = temp;
}
int main() {
  char *str1 = "compiler";
  char *str2 = "interpreter";
  swap(str1, str2);
  printf("str1 is %s, str2 is %s", str1, str2);
  return 0;
}
```

What is the output?

```
int main() {
        char name[] = "Welcome to coding";
        name[10] = '\0';
        name[2] = '3';
        name[5] = '\0';
        printf("%s", name);
        return 0;
}
```
