1. #include <stdio.h>

#include <ctype.h>

void expandString(char \*input) {

    while (\*input) {

        char c = \*input;

        input++;

        int count = 0;

        while (isdigit(\*input)) {

            count = count \* 10 + (\*input - '0');

            input++;

        }

        for (int i = 0; i < count; i++) {

            printf("%c", c);

        }

    }

    printf("\n");

}

int main() {

    char input[100];

    printf("Enter the input string: ");

    scanf("%s", input);

    printf("Output: ");

    expandString(input);

    printf("Enter the next input string: ");

    scanf("%s", input);

    printf("Output: ");

    expandString(input);

    return 0;

}

Output:

Enter the input string: a1b10

Output: abbbbbbbbbb

Enter the next input string: b3c6d15

Output: bbbccccccddddddddddddddd

2. #include <stdio.h>

#include <string.h>

void compressString(char \*input) {

    int length = strlen(input);

    int count = 1;

    for (int i = 0; i < length; i++) {

        if (input[i] == input[i + 1]) {

            count++;

        } else {

            printf("%c", input[i]);

            if (count > 1) {

                printf("%d", count);

            }

            count = 1;

        }

    }

    printf("\n");

}

int main() {

    char input[100];

    printf("Enter the input string: ");

    scanf("%[^\n]", input);

    printf("Output 1: ");

    compressString(input);

    printf("Output 2: ");

    printf("%c", input[0]);

    compressString(input + 1);

    return 0;

}

Output:

Enter the input string: AAABBC

Output 1: A3B2C

Output 2: AA2B2C

Enter the input string: AAABBCCCDE

Output 1: A3B2C3DE

Output 2: AA2B2C3DE

3. #include <stdio.h>

#include <string.h>

void convertToWords(int num, char \*output) {

    char \*units[] = {"", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine"};

    char \*teens[] = {"Ten", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen"};

    char \*tens[] = {"", "", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety"};

    if (num >= 1000) {

        strcat(output, units[num / 1000]);

        strcat(output, " Thousand ");

        num %= 1000;

    }

    if (num >= 100) {

        strcat(output, units[num / 100]);

        strcat(output, " Hundred ");

        num %= 100;

    }

    if (num >= 20) {

        strcat(output, tens[num / 10]);

        strcat(output, " ");

        num %= 10;

    }

    if (num >= 10) {

        strcat(output, teens[num - 10]);

        strcat(output, " ");

        num = 0; // Already processed the tens place

    }

    if (num > 0) {

        strcat(output, units[num]);

    }

}

int main() {

    int num;

    printf("Enter the numeric input: ");

    scanf("%d", &num);

    char output[100] = "";

    convertToWords(num, output);

    printf("Output: %s\n", output);

    return 0;

}

Output:

Enter the numeric input: 1213

One Thousand Two Hundred and Thirteen

4.

#include <stdio.h>

#include <string.h>

void compare(char \*str1, char \*str2) {

int len = strlen(str1);

printf("Output:\n");

for (int i = 0; i < len; i++) {

if (str1[i] != str2[i]) {

printf("%c,%c\n", str1[i], str2[i]);

}

}

}

int main() {

char str1[] = "antonyandcleopatra";

char str2[] = "antaniandcleapadra";

char str3[] = "abcddefgikom ";

char str4[] = "abdcdeffgklm";

printf("Input:\nstr1 = \"%s\"\nstr2 = \"%s\"\n", str1, str2);

compare(str1, str2);

printf("\nInput:\nstr1 = \"%s\"\nstr2 = \"%s\"\n", str3, str4);

compare(str3, str4);

return 0;

}

Output:

Input:

str1 = "antonyandcleopatra"

str2 = "antaniandcleapadra"

Output:

o,a

y,i

o,a

t,d

Input:

str1 = "abcddefgikom "

str2 = "abdcdeffgklm"

Output:

c,d

d,c

g,f

i,g

o,l

5. #include <stdio.h>

#include <string.h>

void justifyText(char \*text, int length) {

    int len = strlen(text);

    int spaces\_needed = length - len;

    int space\_count = 0;

    int space\_per\_gap = 0;

    int extra\_spaces = 0;

    // Count the number of spaces between words

    for (int i = 0; i < len; i++) {

        if (text[i] == '\_') {

            space\_count++;

        }

    }

    // Calculate the number of spaces needed per gap

    if (space\_count > 0) {

        space\_per\_gap = spaces\_needed / space\_count;

        extra\_spaces = spaces\_needed % space\_count;

    }

    // Output the text with justified spaces

    printf("Output: ");

    for (int i = 0; i < len; i++) {

        if (text[i] == '\_') {

            printf("\_");

            for (int j = 0; j < space\_per\_gap; j++) {

                printf(" ");

            }

            if (extra\_spaces > 0) {

                printf(" ");

                extra\_spaces--;

            }

        } else {

            printf("%c", text[i]);

        }

    }

    printf("\n");

}

int main() {

    char text[] = "Zoho\_Corp\_Madurai";

    int length = 25;

    printf("Input:\nText = %s //(length is %d) Padding = %d\n", text, (int)strlen(text), length - (int)strlen(text));

    justifyText(text, length);

    return 0;

}

Output:

Input:

Text = Zoho\_Corp\_Madurai //(length is 17) Padding = 8

Output: Zoho\_ Corp\_ Madurai

6. #include <stdio.h>

#include <stdbool.h>

#include <ctype.h>

#include <string.h>

bool isPalindrome(char \*str) {

int left = 0;

int right = strlen(str) - 1;

while (left < right) {

while (!isalpha(str[left])) {

left++;

}

while (!isalpha(str[right])) {

right--;

}

if (tolower(str[left]) != tolower(str[right])) {

return false;

}

left++;

right--;

}

return true;

}

int main() {

char str1[] = "malayalam";

char str2[] = "m @ala$$y\*a&lam";

char str3[] = "Something";

printf("Input: %s\nOutput: %s\n", str1, isPalindrome(str1) ? "True" : "False");

printf("Input: %s\nOutput: %s\n", str2, isPalindrome(str2) ? "True" : "False");

printf("Input: %s\nOutput: %s\n", str3, isPalindrome(str3) ? "True" : "False");

return 0;

}

Output:

Input: malayalam

Output: True

Input: m @ala$$y\*a&lam

Output: True

Input: Something

Output: False

7.

#include <stdio.h>

#include <string.h>

void swap(char \*x, char \*y) {

char temp = \*x;

\*x = \*y;

\*y = temp;

}

void permute(char \*str, int left, int right) {

if (left == right) {

printf("%s\n", str);

} else {

for (int i = left; i <= right; i++) {

if (i != left && str[i] == str[left]) {

continue;

}

swap(&str[left], &str[i]);

permute(str, left + 1, right);

swap(&str[left], &str[i]);

}

}

}

int main() {

char str[] = "Good";

int n = strlen(str);

printf("Input: %s\nOutput:\n", str);

permute(str, 0, n - 1);

return 0;

}

Output:

Input: Good

Output:

Good

Godo

Gdoo

oGod

oGdo

ooGd

oodG

odoG

odGo

ooGd

oodG

oGod

oGdo

odGo

odoG

dooG

doGo

dGoo

8. #include <stdio.h>

#include <string.h>

void findMismatchedSubstrings(char \*str1, char \*str2) {

int len1 = strlen(str1);

int len2 = strlen(str2);

printf("Output:\n");

for (int i = 0; i < len1 && i < len2; i++) {

if (str1[i] != str2[i]) {

printf("%c%c%c, %c%c%c\n", str1[i], str1[i+1], str1[i+2], str2[i], str2[i+1], str2[i+2]);

i += 2;

}

}

}

int main() {

char str1[] = "AABBCCDD";

char str2[] = "ABCDCCAD";

printf("Input:\n%s, %s\n", str1, str2);

findMismatchedSubstrings(str1, str2);

return 0;

}

Output:

Input:

AABBCCDD, ABCDCCAD

Output:

ABB, BCD

D,A

9. #include <stdio.h>

#include <string.h>

#include <ctype.h>

void countVowels(char \*str) {

int vowels[5] = {0};

for (int i = 0; i < strlen(str); i++) {

char c = tolower(str[i]);

switch (c) {

case 'a':

vowels[0]++;

break;

case 'e':

vowels[1]++;

break;

case 'i':

vowels[2]++;

break;

case 'o':

vowels[3]++;

break;

case 'u':

vowels[4]++;

break;

default:

break;

}

}

printf("Output:\n");

printf("a:%d\n", vowels[0]);

printf("e:%d\n", vowels[1]);

printf("i:%d\n", vowels[2]);

printf("o:%d\n", vowels[3]);

printf("u:%d\n", vowels[4]);

}

int main() {

char str[] = "India";

printf("Input: %s\n", str);

countVowels(str);

return 0;

}

Output:

Input: India

Output:

a:1

e:0

i:2

o:0

u:0

10. #include <stdio.h>

#include <stdbool.h>

#include <string.h>

bool isPalindrome(int num) {

char str[20];

sprintf(str, "%d", num);

int len = strlen(str);

for (int i = 0; i < len / 2; i++) {

if (str[i] != str[len - 1 - i]) {

return false;

}

}

return true;

}

int nextPalindrome(int num) {

while (true) {

num++;

if (isPalindrome(num)) {

return num;

}

}

}

int main() {

int input1 = 123;

int input2 = 12345;

printf("Input: %d\nOutput: %d\n", input1, nextPalindrome(input1));

printf("Input: %d\nOutput: %d\n", input2, nextPalindrome(input2));

return 0;

}

Output:

Input: 123

Output: 131

Input: 12345

Output: 12421