1.

#include <stdio.h>

#include <ctype.h>

int main() {

char input[100];

printf("Enter input string: ");

scanf("%s", input);

char character;

int count = 0;

for (int i = 0; input[i] != '\0'; i++) {

if (isdigit(input[i])) {

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

} else {

character = input[i];

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

printf("%c", character);

}

count = 0;

}

}

printf("\n");

return 0;

}

2.

#include <stdio.h>

#include <string.h>

void compressString(char \*str) {

int len = strlen(str);

if (len == 0) {

printf("Empty string\n");

return;

}

int count = 1;

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

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

count++;

} else {

printf("%c%d", str[i], count);

count = 1;

}

}

}

int main() {

char input[100];

printf("Enter input string: ");

scanf("%s", input);

printf("Compressed string: ");

compressString(input);

printf("\n");

return 0;

}

3.

#include <stdio.h>

#include <string.h>

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

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 == 0) {

strcpy(result, "Zero");

} else {

int thousands = num / 1000;

int hundreds = (num % 1000) / 100;

int tens\_place = (num % 100) / 10;

int ones\_place = num % 10;

result[0] = '\0';

if (thousands > 0) {

strcat(result, units[thousands]);

strcat(result, " Thousand ");

}

if (hundreds > 0) {

strcat(result, units[hundreds]);

strcat(result, " Hundred ");

}

if (tens\_place > 1) {

strcat(result, tens[tens\_place]);

strcat(result, " ");

if (ones\_place > 0) {

strcat(result, units[ones\_place]);

}

} else if (tens\_place == 1) {

strcat(result, teens[ones\_place]);

} else if (ones\_place > 0) {

strcat(result, units[ones\_place]);

}

}

}

int main() {

int num;

printf("Enter a number (0 - 99999): ");

scanf("%d", &num);

if (num < 0 || num > 99999) {

printf("Invalid input. Please enter a number in the range 0 - 99999.\n");

return 1;

}

char result[100];

convertToWords(num, result);

printf("Output: %s\n", result);

return 0;

}

4.

#include <stdio.h>

#include <string.h>

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

int len = strlen(str1);

if (len != strlen(str2)) {

printf("Strings must be of equal length.\n");

return;

}

printf("Mismatched pairs:\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";

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

printf("Output:\n");

compare(str1, str2);

return 0;

}

5.

#include <stdio.h>

#include <string.h>

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

int text\_length = strlen(text);

int spaces\_needed = padding - text\_length;

int num\_spaces = 0;

for (int i = 0; i < text\_length; i++) {

if (text[i] == ' ') {

num\_spaces++;

}

}

int extra\_spaces = spaces\_needed % num\_spaces;

int equal\_spaces = spaces\_needed / num\_spaces;

char justified\_text[100];

int j = 0;

for (int i = 0; i < text\_length; i++) {

if (text[i] != ' ') {

justified\_text[j++] = text[i];

} else {

int spaces\_to\_add = (extra\_spaces-- > 0) ? equal\_spaces + 1 : equal\_spaces;

for (int k = 0; k < spaces\_to\_add; k++) {

justified\_text[j++] = ' ';

}

}

}

justified\_text[j] = '\0';

printf("Output: %s\n", justified\_text);

}

int main() {

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

int padding = 25;

printf("Input:\nText = %s\npadding = %d\n", text, padding);

justifyText(text, padding);

return 0;

}

6.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int isSpecialCharacter(char ch) {

return !isalpha(ch);

}

int isPalindrome(char \*str) {

int left = 0;

int right = strlen(str) - 1;

while (left < right) {

while (left < right && isSpecialCharacter(str[left]))

left++;

while (left < right && isSpecialCharacter(str[right]))

right--;

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

return 0;

left++;

right--;

}

return 1;

}

int main() {

char str1[] = "malayalam";

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

char str3[] = "Something";

printf("Input 1: %s\nOutput 1: %s\n", str1, isPalindrome(str1) ? "True" : "False");

printf("Input 2: %s\nOutput 2: %s\n", str2, isPalindrome(str2) ? "True" : "False");

printf("Input 3: %s\nOutput 3: %s\n", str3, isPalindrome(str3) ? "True" : "False");

return 0;

}

7.

#include <stdio.h>

#include <string.h>

void swap(char \*a, char \*b) {

char temp = \*a;

\*a = \*b;

\*b = temp;

}

void permute(char \*str, int start, int end) {

if (start == end) {

printf("%s, ", str); return;

}

for (int i = start; i <= end; i++) {

int flag = 1;

for (int j = start; j < i; j++) {

if (str[j] == str[i]) {

flag = 0;

break;

}

}

if (flag) {

swap(&str[start], &str[i]); // Swap the characters

permute(str, start + 1, end);

swap(&str[start], &str[i]); // Backtrack

}

}

}

int main() {

char str[] = "Good";

int n = strlen(str);

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

printf("Output: ");

permute(str, 0, n - 1);

printf("\n");

return 0;

}

8.

#include <stdio.h>

#include <string.h>

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

int len1 = strlen(str1);

int len2 = strlen(str2);

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

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

int j = i;

while (j < len1 && j < len2 && str1[j] != str2[j]) {

j++;

}

printf("%.\*s,%.\*s\n", j - i, &str1[i], j - i, &str2[i]);

i = j - 1;

}

}

}

int main() {

char str1[] = "AABBCCDD";

char str2[] = "ABCDCCAD";

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

printf("Output:\n");

findMismatchedSubstrings(str1, str2);

return 0;

}

9.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void countVowels(char \*str) {

int vowels[5] = {0};

for (int i = 0; str[i] != '\0'; i++) {

char ch = tolower(str[i]);

switch (ch) {

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;

}

10.

#include <stdio.h>

#include <stdbool.h>

bool isPalindrome(int n) {

int original = n;

int reversed = 0;

while (n > 0) {

int digit = n % 10;

reversed = reversed \* 10 + digit;

n /= 10;

}

return original == reversed;

}

int nextPalindrome(int n) {

while (true) {

n++;

if (isPalindrome(n))

return n;

}

}

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

int next = nextPalindrome(num);

printf("Next palindrome number: %d\n", next);

return 0;

}