1. **Reversing a 32 bit signed intergers**

#include <stdio.h>

int reverse\_number(int n) {

int sign = (n < 0) ? -1 : 1;

n = (n < 0) ? -n : n;

int reversed\_n = 0;

while (n > 0) {

reversed\_n = reversed\_n \* 10 + n % 10;

n /= 10;

}

return reversed\_n \* sign;

}

int main() {

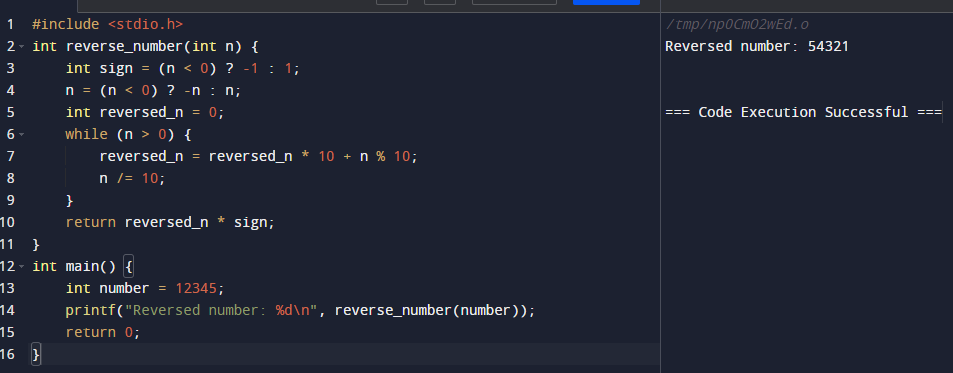
int number = 12345;

printf("Reversed number: %d\n", reverse\_number(number));

return 0;

}

**OUTPUT:**



**2. Check for a valid String**

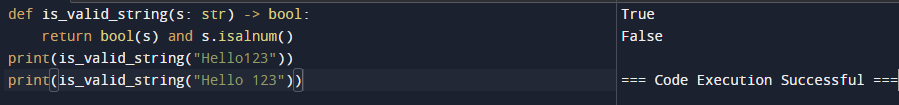
def is\_valid\_string(s: str) -> bool:

return bool(s) and s.isalnum()

print(is\_valid\_string("Hello123"))

print(is\_valid\_string("Hello 123"))

**OUTPUT:**



**3. Merging two Arrays**

#include <stdio.h>

int main() {

int arr1[] = {1, 2, 3, 4, 5};

int n1 = sizeof(arr1) / sizeof(arr1[0]);

int arr2[] = {6, 7, 8, 9, 10};

int n2 = sizeof(arr2) / sizeof(arr2[0]);

int merged[n1 + n2];

for (int i = 0; i < n1; i++) {

merged[i] = arr1[i];

}

for (int i = 0; i < n2; i++) {

merged[n1 + i] = arr2[i];

}

printf("Merged Array: ");

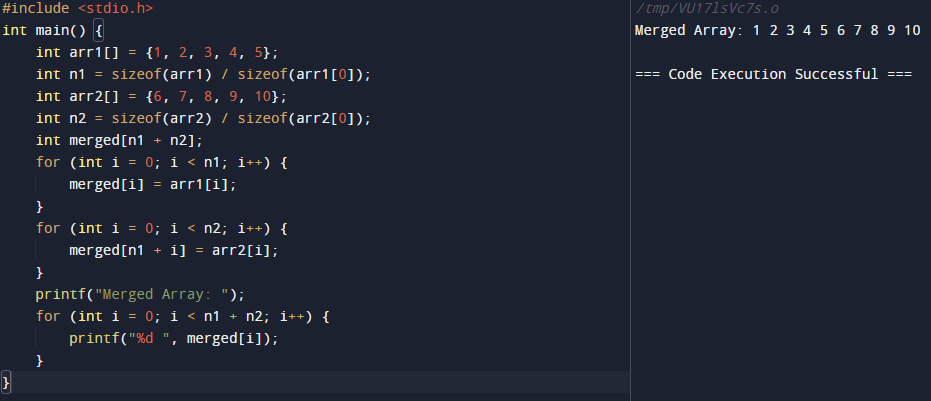
for (int i = 0; i < n1 + n2; i++) {

printf("%d ", merged[i]);

}

}

**OUTPUT:**



1. **Given an array finding duplication values**

#include <stdio.h>

void findDuplicates(int arr[], int size) {

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

for (int j = i + 1; j < size; j++) {

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

printf("Duplicate value: %d\n", arr[i]);

break;

}

}

}

}

int main() {

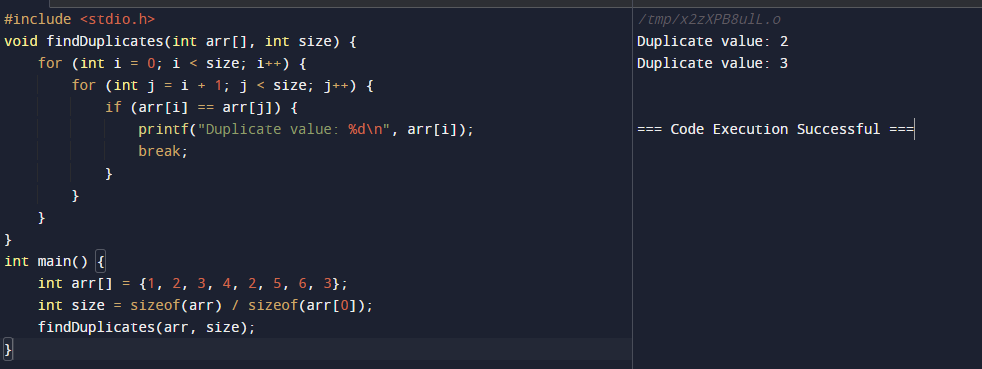
int arr[] = {1, 2, 3, 4, 2, 5, 6, 3};

int size = sizeof(arr) / sizeof(arr[0]);

findDuplicates(arr, size);

}

**OUTPUT:**



**5. Merging of list**

#include <stdio.h>

#include <stdlib.h>

int main() {

int list1[] = {1, 2, 3, 4};

int list2[] = {5, 6, 7, 8};

int size1 = sizeof(list1) / sizeof(list1[0]);

int size2 = sizeof(list2) / sizeof(list2[0]);

int mergedSize = size1 + size2;

int mergedList[mergedSize];

for (int i = 0; i < size1; i++) {

mergedList[i] = list1[i];

}

for (int i = 0; i < size2; i++) {

mergedList[size1 + i] = list2[i];

}

printf("Merged List: ");

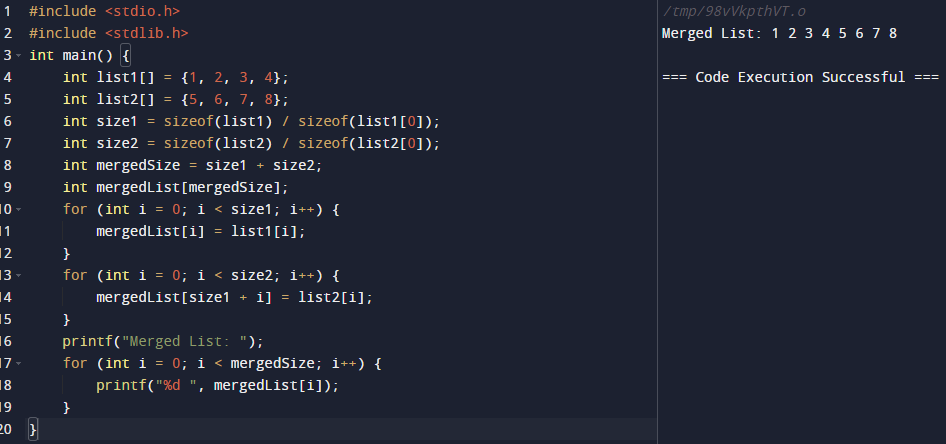
for (int i = 0; i < mergedSize; i++) {

printf("%d ", mergedList[i]);

}

}

**OUTPUT:**



**6. Given array of reg nos need to search for particular reg no**

#include <stdio.h>

int main() {

int regNos[] = {123, 456, 789, 1011, 1213};

int searchRegNo = 1011;

int size = sizeof(regNos) / sizeof(regNos[0]);

int found = 0;

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

if (regNos[i] == searchRegNo) {

found = 1;

break;

}

}

if (found) {

printf("Registration number %d found in the array.", searchRegNo);

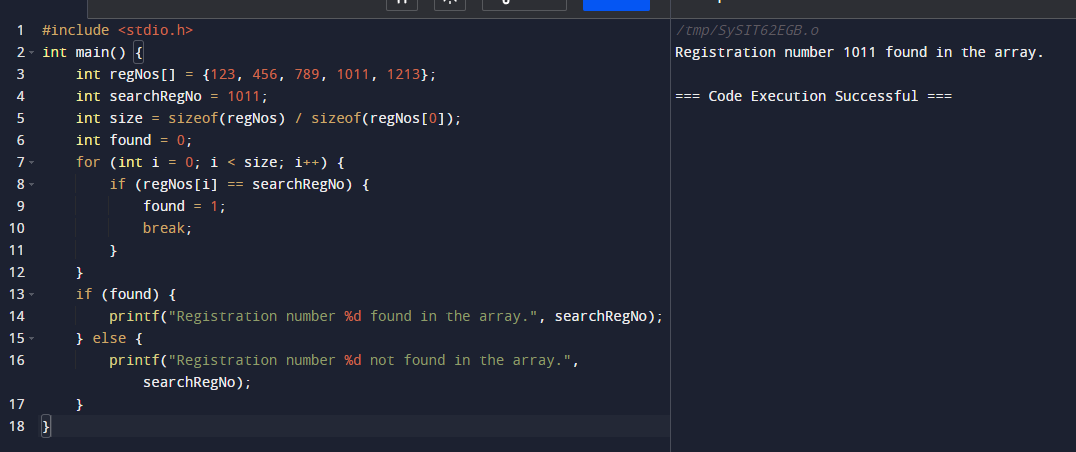
} else {

printf("Registration number %d not found in the array.", searchRegNo);

}

}

**OUTPUT:**



**7. Identify location of element in given array**

#include <stdio.h>

int main() {

int arr[] = {10, 20, 30, 40, 50};

int search\_element = 30;

int size = sizeof(arr) / sizeof(arr[0]);

int position = -1;

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

if (arr[i] == search\_element) {

position = i;

break;

}

}

if (position != -1) {

printf("Element found at index: %d", position);

} else {

printf("Element not found in the array");

}

}

**OUTPUT:**



1. **Given array print odd and even values**

#include <stdio.h>

int main() {

int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

int i;

printf("Odd numbers: ");

for (i = 0; i < 10; i++) {

if (arr[i] % 2 != 0) {

printf("%d ", arr[i]);

}

}

printf("\nEven numbers: ");

for (i = 0; i < 10; i++) {

if (arr[i] % 2 == 0) {

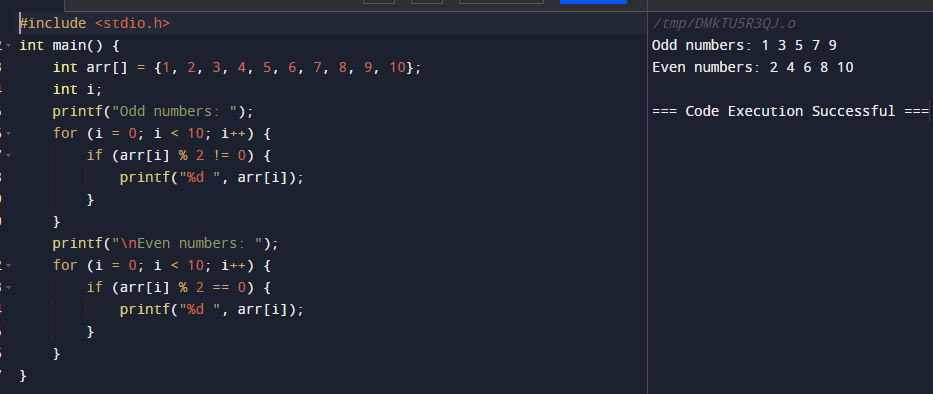
printf("%d ", arr[i]);

}

}

}

**OUTPUT:**



1. **Sum of Fibonacci Series**

#include <stdio.h>

int main() {

int n, first = 0, second = 1, next, sum = 0;

printf("Enter the number of terms: ");

scanf("%d", &n);

printf("Fibonacci Series: ");

for (int i = 1; i <= n; ++i) {

printf("%d, ", first);

sum += first;

next = first + second;

first = second;

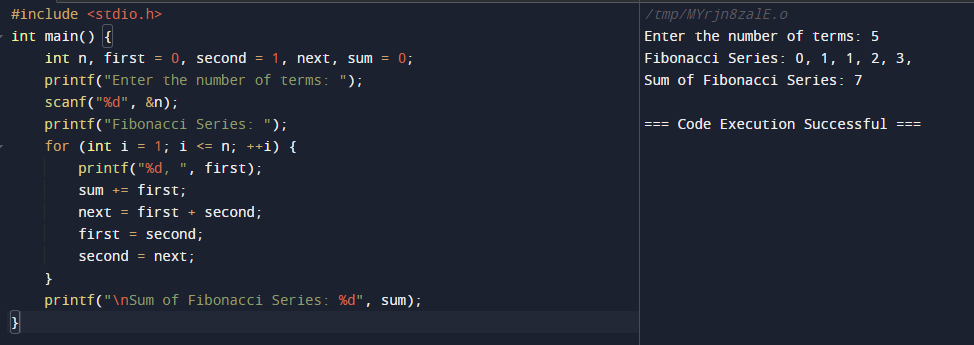
second = next;

}

printf("\nSum of Fibonacci Series: %d", sum);

}

**OUTPUT:**



**10. Finding factorial of a number**

#include <stdio.h>

int main() {

int number, i, factorial = 1;

printf("Enter a positive integer: ");

scanf("%d", &number);

for (i = 1; i <= number; ++i) {

factorial \*= i;

}

printf("Factorial of %d = %llu", number, factorial);

}

**OUTPUT:**

