Day 8 assignement

#include <stdio.h>

int isPrime(int n) {

if (n <= 1) return 0;

if (n <= 3) return 1;

if (n % 2 == 0 || n % 3 == 0) return 0;

for (int i = 5; i \* i <= n; i += 6) {

if (n % i == 0 || n % (i + 2) == 0) return 0;

}

return 1;

}

int findPrimeSum(int n) {

int count = 0;

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

if (isPrime(i) && isPrime(n - i)) {

printf("%d = %d + %d\n", n, i, n - i);

count++;

}

}

return count;

}

int main() {

int num;

printf("Enter a positive integer: ");

scanf("%d", &num);

int ways = findPrimeSum(num);

if (ways == 0)

printf("NoofWays = -1\n");

else

printf("NoofWays = %d\n", ways);

return 0;

}

output

Enter a positive integer: 8

8 = 3 + 5

NoofWays =1

=== Code Execution Successful ===

2.

#include <stdio.h>

int isPrime(int n) {

if (n <= 1) return 0; // 0 and 1 are not prime

if (n <= 3) return 1; // 2 and 3 are prime

if (n % 2 == 0 || n % 3 == 0) return 0; // multiples of 2 and 3 are not prime

for (int i = 5; i \* i <= n; i += 6) {

if (n % i == 0 || n % (i + 2) == 0) return 0;

}

return 1;

}

int findPrimeSum(int n) {

int count = 0;

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

if (isPrime(i) && isPrime(n - i)) {

printf("%d = %d + %d\n", n, i, n - i);

count++;

}

}

return count;

}

int main() {

int num;

printf("Enter a positive integer: ");

scanf("%d", &num);

int ways = findPrimeSum(num);

if (ways == 0)

printf("NoofWays = -1\n");

else

printf("NoofWays = %d\n", ways);

return 0;

}

Output

nter a positive integer: 8

8 = 3 + 5

NoofWays = 1

3

#include <stdio.h>

#include <stdbool.h>

bool isPrime(int n) {

if (n <= 1) return false; // 0 and 1 are not prime

if (n <= 3) return true; // 2 and 3 are prime

if (n % 2 == 0 || n % 3 == 0) return false; // multiples of 2 and 3 are not prime

for (int i = 5; i \* i <= n; i += 6) {

if (n % i == 0 || n % (i + 2) == 0) return false;

}

return true;

}

int findNthPrime(int n) {

int count = 0;

int num = 2; // Start checking from 2

while (count < n) {

if (isPrime(num)) {

count++;

if (count == n) {

return num;

}

}

num++;

}

return -1; // If n is out of range

}

int main() {

int t;

printf("Enter the number of test cases: ");

scanf("%d", &t);

while (t--) {

int n;

printf("Enter the value of N: ");

scanf("%d", &n);

int nthPrime = findNthPrime(n);

printf("The %dth prime number made of prime digits is: %d\n", n, nthPrime);

}

return 0;

}

4.

#include <stdio.h>

#include <stdbool.h>

#include <string.h>emove characters from str2 which are present in str1

void removeChars(char str1[], char str2[]) {

bool present[256] = {false};

for (int i = 0; str1[i]; i++) {

present[str1[i]] = true;

}

int index = 0; // Index to store the result string

// Iterate through str2 and copy characters not present in str1 to result string

for (int i = 0; str2[i]; i++) {

if (!present[str2[i]]) {

str2[index++] = str2[i];

}

}

str2[index] = '\0'; // Null terminate the result string

}

int main() {

char str1[100], str2[100];

printf("Enter the first string: ");

fgets(str1, sizeof(str1), stdin);

str1[strcspn(str1, "\n")] = '\0'; // Remove newline character

printf("Enter the second string: ");

fgets(str2, sizeof(str2), stdin);

str2[strcspn(str2, "\n")] = '\0'; // Remove newline character

removeChars(str1, str2);

printf("Resultant string: %s\n", str2);

return 0;

}