DAA PROGRAMS

1)

#include <stdio.h>

int fibonacci(int n) {

if (n <= 1) {

return n;

} else {

return fibonacci(n - 1) + fibonacci(n - 2);

}

}

int main() {

int n, i;

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

scanf("%d", &n);

printf("Fibonacci Series: ");

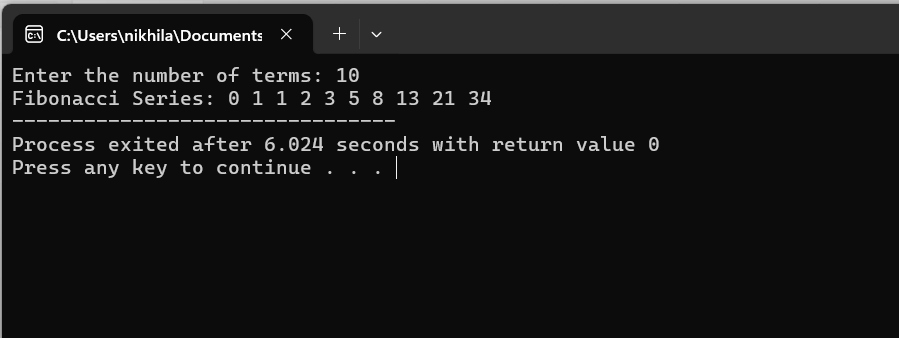
for (i = 0; i < n; i++) {

printf("%d ", fibonacci(i));

}

return 0;

}



2) #include <stdio.h>

#include <math.h>

int numDigits(int n) {

return (n == 0) ? 0 : 1 + numDigits(n / 10);

}

int sumOfPowers(int n, int num\_digits) {

return (n == 0) ? 0 : pow(n % 10, num\_digits) + sumOfPowers(n / 10, num\_digits);

}

int isArmstrong(int n) {

int num\_digits = numDigits(n);

return (n == sumOfPowers(n, num\_digits));

}

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

if (isArmstrong(num)) {

printf("%d is an Armstrong number.\n", num);

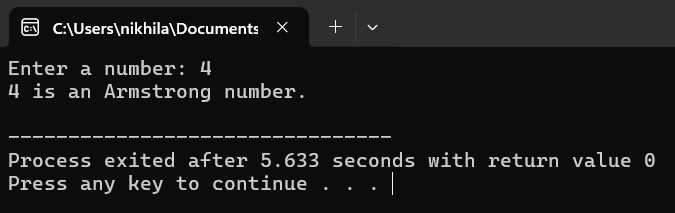
} else {

printf("%d is not an Armstrong number.\n", num);

}

return 0;

}



3) #include <stdio.h>

int gcd(int a, int b) {

if (b == 0) {

return a;

} else {

return gcd(b, a % b);

}

}

int main() {

int num1, num2;

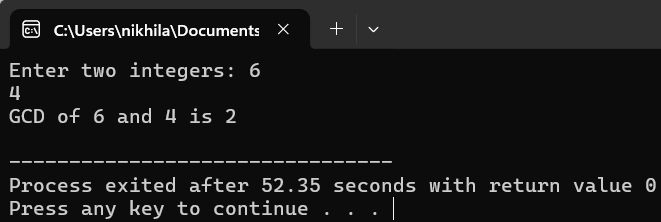
printf("Enter two integers: ");

scanf("%d %d", &num1, &num2);

printf("GCD of %d and %d is %d\n", num1, num2, gcd(num1, num2));

return 0;

}



4) #include <stdio.h>

int main() {

int n;

printf("Enter the number of elements in the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array:\n");

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

scanf("%d", &arr[i]);

}

int largest = arr[0];

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

if (arr[i] > largest) {

largest = arr[i];

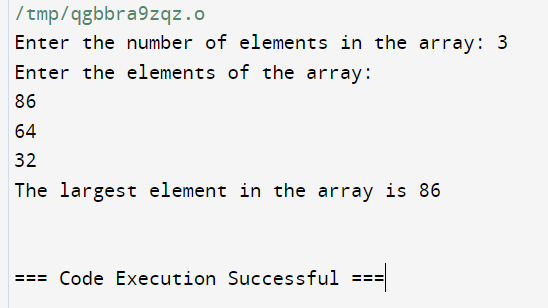
}

}

printf("The largest element in the array is %d\n", largest);

return 0;

}



5) #include <stdio.h>

unsigned long long factorial(int n) {

if (n == 0) {

return 1;

} else {

return n \* factorial(n - 1);

}

}

int main() {

int num;

printf("Enter a non-negative integer: ");

scanf("%d", &num);

if (num < 0) {

printf("Factorial is not defined for negative numbers.\n");

} else {

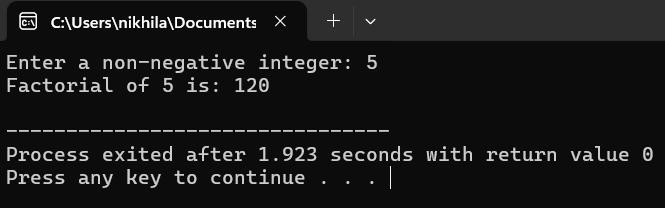
unsigned long long fact = factorial(num);

printf("Factorial of %d is: %llu\n", num, fact);

}

return 0;

}



6) #include <stdio.h>

#include <stdbool.h>

bool isPrime(int n, int i) {

if (n <= 1) {

return false;

}

if (i == 1) {

return true;

}

if (n % i == 0) {

return false;

}

return isPrime(n, i - 1);

}

int main() {

int num;

printf("Enter a number to check if it is prime: ");

scanf("%d", &num);

if (isPrime(num, num / 2)) {

printf("%d is a prime number.\n", num);

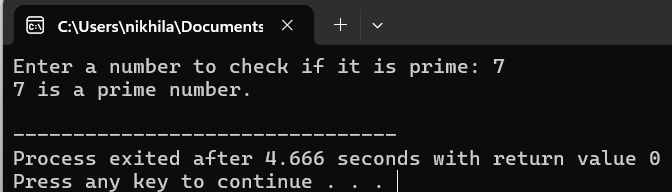
} else {

printf("%d is not a prime number.\n", num);

}

return 0;

}



7) #include <stdio.h>

void selectionSort(int arr[], int n) {

int i, j, min\_index;

for (i = 0; i < n - 1; i++) {

min\_index = i;

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

if (arr[j] < arr[min\_index]) {

min\_index = j;

}

}

if (min\_index != i) {

int temp = arr[i];

arr[i] = arr[min\_index];

arr[min\_index] = temp;

}

}

}

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

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

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

}

printf("\n");

}

int main() {

int n;

printf("Enter the number of elements in the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array:\n");

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

scanf("%d", &arr[i]);

}

printf("Array before sorting: ");

printArray(arr, n);

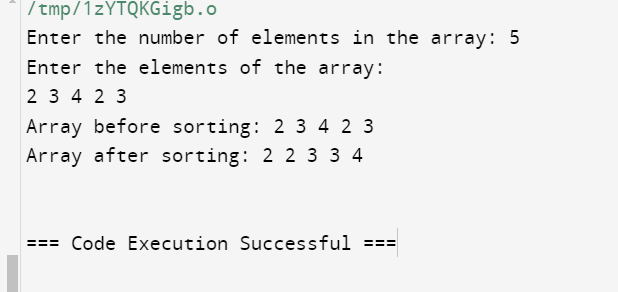
selectionSort(arr, n);

printf("Array after sorting: ");

printArray(arr, n);

return 0;

}



8) #include <stdio.h>

void bubbleSort(int arr[], int n) {

int i, j;

for (i = 0; i < n - 1; i++) {

for (j = 0; j < n - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

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

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

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

}

printf("\n");

}

int main() {

int n;

printf("Enter the number of elements in the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array:\n");

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

scanf("%d", &arr[i]);

}

printf("Array before sorting: ");

printArray(arr, n);

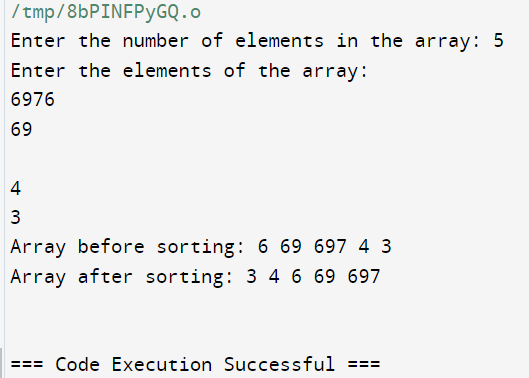
bubbleSort(arr, n);

printf("Array after sorting: ");

printArray(arr, n);

return 0;

}



9) #include <stdio.h>

#include <time.h>

#define N 100

void multiply\_matrices(int mat1[][N], int mat2[][N], int res[][N]) {

int i, j, k;

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

for (j = 0; j < N; j++) {

res[i][j] = 0;

for (k = 0; k < N; k++) {

res[i][j] += mat1[i][k] \* mat2[k][j];

}

}

}

}

int main() {

int mat1[N][N], mat2[N][N], res[N][N];

int i, j;

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

for (j = 0; j < N; j++) {

mat1[i][j] = i + j;

mat2[i][j] = i - j;

}

}

clock\_t start\_time = clock();

multiply\_matrices(mat1, mat2, res);

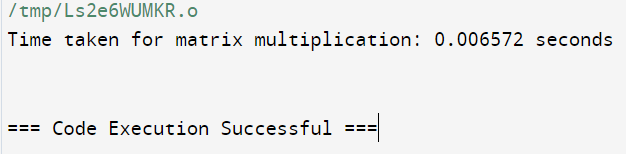
clock\_t end\_time = clock();

double time\_spent = (double)(end\_time - start\_time) / CLOCKS\_PER\_SEC;

printf("Time taken for matrix multiplication: %f seconds\n", time\_spent);

return 0;

}



10) #include <stdio.h>

#include <math.h>

int reverse(int num);

int isPalindrome(int num);

int main()

{

int num;

printf("Enter any number: ");

scanf("%d", &num);

if(isPalindrome(num) == 1)

{

printf("%d is palindrome number.\n", num);

}

else

{

printf("%d is NOT palindrome number.\n", num);

}

return 0;

}

int isPalindrome(int num)

{

if(num == reverse(num))

{

return 1;

}

return 0;

}

int reverse(int num)

{

int digit = (int)log10(num);

if(num == 0)

return 0;

return ((num%10 \* pow(10, digit)) + reverse(num/10));

}

