

**Bangladesh University of Professionals**

Department of CSE

Faculty of Science and Technology

Structured Programming Language Laboratory

CSE 1204

Submitted By : Nusair Ahmed Chowdhury

ID : 23524202052

Submitted To : Sobhana Jahan

Date : 18rd May2024

1. GCD and LCM of two numbers.

Code:

**#include** <stdio.h>

**int** GCD(**int** *x*, **int** *y*) {

**if**(*y***==**0) **return** *x*;

**else** **return** GCD(*y*, *x***%***y*);

}

**int** LCM(**int** *x*, **int** *y*) {

**if**(*x* **==** 0 **||** *y* **==** 0) {

**return** 0;

    } **else**

**return** (*x***\****y*)**/**GCD(*x*,*y*);

}

**int** main() {

**int** x,y;

    printf("Enter first number: ");

    scanf("%d", **&**x);

    printf("Enter second number: ");

    scanf("%d", **&**y);

    printf("---Select Option---\n");

    printf("1. GCD 2.LCM\n");

    printf("Enter preference: ");

**int** choice;

    scanf("%d", **&**choice);

**if**(choice **==** 1){

**int** result **=** GCD(x,y);

        printf("GCD of %d and %d is %d\n", x, y, result);

    }

**else** **if**(choice **==** 2){

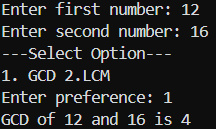
**int** result **=** LCM(x,y);

        printf("LCM of %d and %d is %d\n", x, y, result);

    }

**return** 0;

}

Output:

1. Check if a year is leap year or not.

Code:

**#include** <stdio.h>

**int** LeapYear(**int** *year*) {

**int** flag;

**if**((*year* **%** 4 **==** 0 **&&** *year* **%** 100 **!=**0) **||** (*year* **%** 400 **==** 0)) {

        flag **=** 1;

    } **else** flag **=** 0;

**return** flag;

}

**int** main() {

**int** year;

    printf("Enter year: ");

    scanf("%d", **&**year);

**int** result **=** LeapYear(year);

    (result **==** 1) **?** printf("%d is a Leap Year", year) **:** printf("%d is not a Leap Year", year);

**return** 0;

}

Output:

1. Compare two strings same or not.

Code:

**#include** <stdio.h>

**#include** <string.h>

**int** main() {

**char** str1[50];

**char** str2[50];

    printf("Enter string 1: ");

    gets(str1);

    printf("Enter string 2: ");

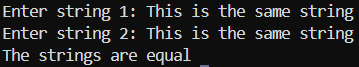
    gets(str2);

**int** result **=** strcmp(str1, str2);

    (result **==** 0) **?** printf("The strings are equal") **:** printf("The strings are not equal");

**return** 0;

}

Output:

1. Calculate grades from the number.

Code:

**#include** <stdio.h>

**int** main() {

**int** marks;

    printf("Enter your marks: ");

    scanf("%d", **&**marks);

**if**(marks **>** 100 **||** marks **<** 0)

        printf("Invalid marks");

**else** **if**(marks **>=** 90 **&&** marks **<**100)

        printf("You've got A+!");

**else** **if**(marks **>=**80 **&&** marks **<** 90)

        printf("You've got A!");

**else** **if**(marks **>=**70 **&&** marks **<** 80)

        printf("You've got B!");

**else** **if**(marks **>=**60 **&&** marks **<** 70)

        printf("You've got C!");

**else** **if**(marks **>=**50 **&&** marks **<**60)

        printf("You've got D!");

**else** **if**(marks **!=** 0 **&&** marks **<** 50)

        printf("You've failed!");

**return** 0;

}

Output:

1. Separate odd and even numbers from an array and put it in another array called odd and even.

Code:

**#include** <stdio.h>

**int** checkEven(**int** *x*) {

**if**(*x* **%** 2 **==** 0) {

**return** 1;

    } **else** {

**return** 0;

    }

}

**int** main() {

**int** n;

    printf("Enter the size of the array: ");

    scanf("%d", **&**n);

**int** num[n];

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

        printf("Enter number %d: ", i **+** 1);

        scanf("%d", **&**num[i]);

    }

**int** even[n];

**int** odd[n];

**int** j **=** 0, k **=** 0;

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

**int** flag **=** checkEven(num[i]);

**if**(flag **==** 1) {

**if**(num[i] **==** 0) **continue**;

**else**{

                even[j] **=** num[i];

                j**++**;

            }

        } **else** {

            odd[k] **=** num[i];

            k**++**;

        }

    }

    printf("Even numbers: ");

**for**(**int** i **=** 0; i **<** j; i**++**) {

        printf("%d ", even[i]);

    }

    printf("\nOdd numbers: ");

**for**(**int** i **=** 0; i **<** k; i**++**) {

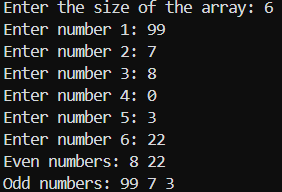
        printf("%d ", odd[i]);

    }

    printf("\n");

**return** 0;

}

Output:

1. Separate the prime numbers from array and print Fibonacci series using array.

Code:

#include <stdio.h>

int checkPrime(int x) {

    if(x <= 1) return 0;

    for(int i = 2; i < x; i++) {

        if(x%i == 0) return 0;

    }

    return 1;

}

void fibonacciPrint(int n1, int n2) {

    int n;

    printf("Fibonacci Series upto : ");

    scanf("%d", &n);

    int fib[n];

    fib[0] = n1;

    fib[1] = n2;

    printf("%d %d ",fib[0], fib[1]);

    for(int i = 2; i < n; i++) {

        fib[i] = fib[i - 1] + fib[i - 2];

        printf("%d ", fib[i]);

    }

    printf("\n");

}

int main() {

    int k;

    printf("Enter the size of the array: ");

    scanf("%d", &k);

    int n[k];

    for(int i = 0; i < k; i++) {

        printf("Enter the %dth element: ", i+1);

        scanf("%d", &n[i]);

    }

    int j = 0;

    int prime[j];

    printf("The prime numbers are: ");

    for(int i = 0; i < k; i++) {

        if(checkPrime(n[i]) == 1) {

            prime[j] = n[i];

            j++;

            printf("%d ", prime[j-1]);

        }

    }

    printf("\n");

    int num1 = prime[0];

    int num2 = prime[1];

    fibonacciPrint(num1, num2);

    return 0;

}

Output:

1. Sum of numbers in Fibonacci Series until nth position.

Code:

**#include** <stdio.h>

**int** main() {

**int** n;

    printf("Enter the nth number of position: ");

    scanf("%d", **&**n);

**int** a **=** 0, b **=** 0, c **=** 1, sum **=** 1;

**if**(n **<=** 0) {

        sum **=** 0;

    }

**else** {

        printf("0 + 1 ");

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

            a **=** b;

            b **=** c;

            c **=** a **+** b;

            sum **+=** c;

            printf(" + %d", c);

        }

        printf(" = %d\n", sum);

    }

**return** 0; }

Output:

1. Find out if a number is Armstrong number or not.

Code:

#include <stdio.h>

int Count(int n) {

    int count = 0;

    while(n!= 0) {

        count++;

        n /= 10;

    }

    return count;

}

int power(int n, int count) {

    int result = 1;

    for(int i = 0; i < count; i++) {

        result \*= n;

    }

    return result;

}

int checkArmstrong(int n) {

    int count = Count(n);

    int sum = 0;

    int temp = n;

    while(temp!= 0) {

        int rem = temp % 10;

        sum += power(rem, count);

        temp /= 10;

    }

    printf("= %d\n", sum);

    if(sum == n) {

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

    }

    else {

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

    }

}

int main() {

    long long int number;

    printf("Enter then number to be checked : ");

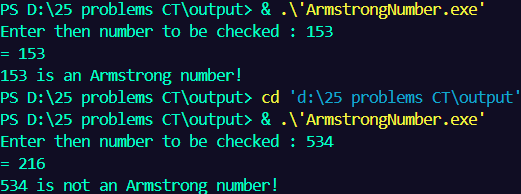
    scanf("%lld", &number);

    checkArmstrong(number);

    return 0;

}

Output:



1. Find the sum of the numbers in an array.

Code:

#include <stdio.h>

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

    for(int i = 0; i<n; i++) {

        printf("Enter element %d: ", i+1);

        scanf("%d", &arr[i]);

    }

}

int sumOfArrays(int n,int arr[]) {

    int sum = 0;

    for (int i = 0; i < n; i++)

    {

        sum += arr[i];

    }

    return sum;

}

int main() {

    int n;

    printf("Number of elements in array: ");

    scanf("%d", &n);

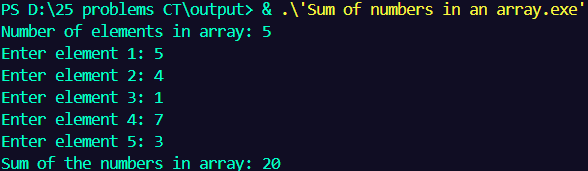
    int arr[n];

    arrayInput(n,arr);

    printf("Sum of the numbers in array: %d\n",sumOfArrays(n,arr));

    return 0;

}

Output:

1. Find the largest, smallest and the average value in an array.

Code:

#include <stdio.h>

int main() {

    int n;

    printf("Enter the size of array: ");

    scanf("%d", &n);

    int a[n];

    for(int i = 0; i < n; i++) {

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

        scanf("%d", &a[i]);

    }

    int largest = a[0], smallest = a[0], sum = a[0], avg;

    for(int i = 1; i < n; i++) {

        if(largest < a[i])

            largest = a[i];

        if(smallest > a[i])

            smallest = a[i];

        sum += a[i];

    }

    avg = sum / n;

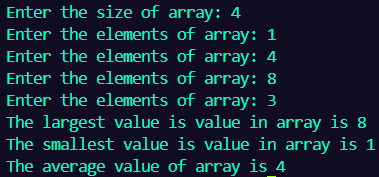
    printf("The largest value is value in array is %d\n", largest);

    printf("The smallest value is value in array is %d\n", smallest);

    printf("The average value of array is %d\n", avg);

    return 0;

}

Output:

1. Count even and odd in an array.

Code:

#include <stdio.h>

int main() {

    int n;

    printf("Enter the size of array: ");

    scanf("%d", &n);

    int a[n];

    for(int i = 0; i < n; i++) {

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

        scanf("%d", &a[i]);

    }

    int even = 0, odd = 0;

    for(int i = 0; i < n; i++) {

        if(a[i]%2==0)

            even++;

        if(a[i]%2==1)

            odd++;

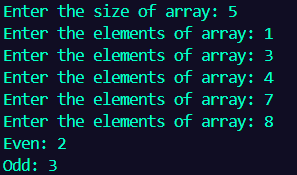
    }

    printf("Even: %d\n", even);

    printf("Odd: %d\n", odd);

    return 0;

}

Output:

1. Reverse an array.

Code:

#include <stdio.h>

int main() {

    int num[5] = {1,4,3,8,2};

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

    printf("The array is: ");

    for(int i=0; i<size; i++) {

        printf("%d ",num[i]);

    }

    printf("\n");

    for(int i=0; i<size/2; i++) {

        for(int j=size-1; j>=size/2; j--) {

            int temp = num[i];

            num[i] = num[j];

            num[j] = temp;

            i++;

        }

    }

    printf("The reversed array is: ");

    for(int i=0; i<size; i++) {

        printf("%d ",num[i]);

    }

    return 0;

}

Output:



1. Print all prime numbers of an array.

Code:

#include <stdio.h>

int checkPrime(int x) {

    if(x <= 1) return 0;

    for(int i = 2; i < x; i++) {

        if(x%i == 0) return 0;

    }

    return 1;

}

int main() {

    int k;

    printf("Enter the size of the array: ");

    scanf("%d", &k);

    int n[k];

    for(int i = 0; i < k; i++) {

        printf("Enter the %dth element: ", i+1);

        scanf("%d", &n[i]);

    }

    printf("The array is: ");

    for(int i = 0; i < k; i++) {

        printf("%d ", n[i]);

    }

    printf("\n");

    int j = 0;

    int prime[j];

    printf("The prime numbers are: ");

    for(int i = 0; i < k; i++) {

        if(checkPrime(n[i]) == 1) {

            prime[j] = n[i];

            j++;

            printf("%d ", prime[j-1]);

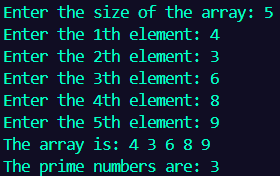
        }

    }

    printf("\n");

    return 0;

}

Output:

1. Check whether the number is Palindrome or not.

Code:

#include <stdio.h>

int main() {

    int n, reversed = 0, remainder, original;

    printf("Enter an integer : ");

    scanf("%d", &n);

    original = n;

    while(n != 0) {

        remainder = n % 10;

        reversed = reversed \* 10 + remainder;

        n /= 10;

    }

    (original == reversed) ? printf("%d is a palindrome", original) : printf("%d is not a palindrome",original);

    return 0;

}

Output:



1. Find decimal to Octal and Octal to decimal.

Code:

#include <stdio.h>

int Dec\_Oct(int dec) {

    int oct = 0, i = 1;

    while (dec!= 0) {

        oct += (dec % 8) \* i;

        dec /= 8;

        i \*= 10;

    }

    return oct;

}

int Oct\_Dec (int oct) {

    int dec = 0, i = 1;

    while (oct!= 0) {

        dec += (oct % 10) \* i;

        oct /= 10;

        i \*= 8;

    }

    return dec;

}

int main() {

    printf("Choose option: \n");

    printf("1. Decimal to octal\n");

    printf("2. Octal to decimal\n Option: ");

    int option;

    scanf("%d", &option);

    int num;

    printf("Input number : ");

    scanf("%d", &num);

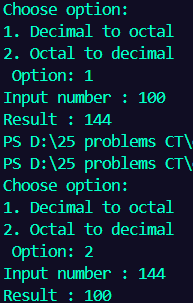
    int result = (option == 1) ? (result =  Dec\_Oct(num)) : (result = Oct\_Dec(num));

    printf("Result : %d", result);

    return 0;

}

Output:



1. Print the n Cr and n Pr of the given n and r.

Code:

#include <stdio.h>

int printFactorization(int a) {

    int product = 1;

    for(int i = 1; i <= a; i++) {

        product \*= i;

    }

    return product;

}

int main() {

    int n,r;

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

    scanf("%d", &n);

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

    scanf("%d", &r);

    int nCr = printFactorization(n) / (printFactorization(r) \* printFactorization(n - r));

    printf("The value of nCr is : %d\n", nCr);

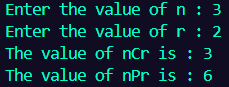
    int nPr = printFactorization(n) / printFactorization(n - r);

    printf("The value of nPr is : %d", nPr);

    return 0;

}

Output:



1. Copy elements of an array to another array.

Code:

#include <stdio.h>

int main() {

    int arr1[5] = {1, 2, 3, 4, 5};

    int arr2[5];

    printf("Elements of the original array : ");

    for(int i = 0; i < 5; i++) {

        printf("%d ", arr1[i]);

    }

    for(int i = 0; i < 5; i++) {

        arr2[i] = arr1[i];

    }

    printf("\n");

    printf("Elements of the coppied array : ");

    for(int i = 0; i < 5; i++) {

        printf("%d ", arr2[i]);

    }

    return 0;

}

Output:



1. Find factorial using a function.

Code:

#include <stdio.h>

void factorial(int a) {

    int product = 1, temp = a;

    while (temp) {

        product \*= temp;

        temp--;

    }

    printf("The factorial of %d is %d", a, product);

}

int main() {

    int n;

    printf("Enter a number: ");

    scanf("%d", &n);

    factorial(n);

    return 0;

}

Output:



1. Find the length of the given string using function.

Code:

#include <stdio.h>

int strLength(char str[]) {

    int len = 0;

    while (str[len] != '\0') {

        len++;

    }

    return len;

}

int main() {

    char str[11] = "Hello World";

    printf("Length of the string : %d", strLength(str));

    return 0;

}

Output:



1. Reverse the array using a function.

Code:

#include <stdio.h>

void reverseArray(int num[], int size) {

    for(int i=0; i<size/2; i++) {

        for(int j=size-1; j>=size/2; j--) {

            int temp = num[i];

            num[i] = num[j];

            num[j] = temp;

            i++;

        }

    }

}

int main() {

    int num[5] = {1,4,3,8,2};

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

    printf("The array is: ");

    for(int i=0; i<size; i++) {

        printf("%d ",num[i]);

    }

    printf("\n");

    reverseArray(num, size);

    printf("The reversed array is: ");

    for(int i=0; i<size; i++) {

        printf("%d ",num[i]);

    }

    return 0;

}

Output:

1. Print the pattern.

1

2 2

3 3 3

4 4 4 4

Code:

#include <stdio.h>

int main() {

    int n;

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

    scanf("%d", &n);

    for(int i = 1; i <=n; i++) {

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

            printf("  ");

        }

        for(int j = 1; j <= i; j++) {

            printf("%4d", i);

        }

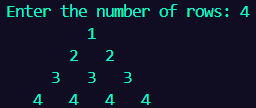
        printf("\n");

    }

    return 0;

}

Output:



1. Print the pattern:

\*

\* \*

\* \*

\* \*

\*

Code:

    #include <stdio.h>

int main() {

    int n;

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

    scanf("%d", &n);

    for(int i = 1; i <=n; i++) {

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

            printf(" ");

        }

        for(int j = 1; j <= i; j++) {

            if(j == 1 || j == i) {

                printf(" \* ");

            } else printf("  ");

        }

        printf("\n");

    }

    for(int i = n - 1; i > 0; i--) {

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

            printf(" ");

        }

        for(int j = 1; j <= i; j++) {

            if(j == 1 || j == i) {

                printf(" \* ");

            } else printf("  ");

        }

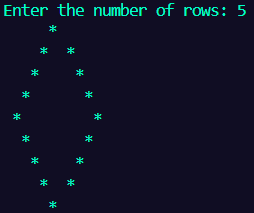
        printf("\n");

    }

    return 0;

}

Output:



1. Print the following pattern:

\*

\*\*

\*\*\*

\*\*\*\*

Code:

#include <stdio.h>

int main() {

    int n;

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

    scanf("%d", &n);

    for(int i = 1; i <=n; i++) {

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

            printf(" ");

        }

        for(int j = 1; j <= i; j++) {

            printf(" \*");

        }

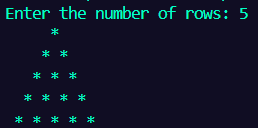
        printf("\n");

    }

    return 0;

}

Output:



1. Print the following pattern:

1

12

123

1234

12345

Code:

    #include <stdio.h>

    int main() {

        int n;

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

        scanf("%d", &n);

        for(int i = 1; i <= n; i++) {

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

                printf("%d", j);

            }

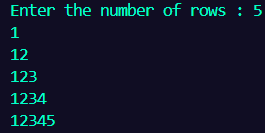
            printf("\n");

        }

        return 0;

    }

Output:



1. Print the pattern:

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

Code:

#include <stdio.h>

int main() {

    int n;

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

    scanf("%d", &n);

    for(int i = n; i >= 1; i--) {

        for(int j = 1; j <= i; j++) {

            printf("\* ");

        }

        printf("\n");

    }

    return 0;

}

Output:

