Computer programming

1. C program to perform all the arithmetic operations.

Input:-

#include <stdio.h>

int main(void) {

double num1, num2;

char op;

printf("Enter an operator (+, -, \*, /): ");

scanf("%c", &op);

printf("Enter two numbers: ");

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

if (op == '+') {

printf("%.1lf + %.1lf = %.1lf\n", num1, num2, num1 + num2);

} else if (op == '-') {

printf("%.1lf - %.1lf = %.1lf\n", num1, num2, num1 - num2);

} else if (op == '\*') {

printf("%.1lf \* %.1lf = %.1lf\n", num1, num2, num1 \* num2);

} else if (op == '/') {

printf("%.1lf / %.1lf = %.1lf\n", num1, num2, num1 / num2);

} else {

printf("Error: Invalid operator\n");

}

return 0;}

2. C program to find area of a triangle if base and height are given.

#include <stdio.h>

int main(void) {

double base, height, area;

printf("Enter the base of the triangle: ");

scanf("%lf", &base);

printf("Enter the height of the triangle: ");

scanf("%lf", &height);

area = (base \* height) / 2;

printf("The area of the triangle is: %.2lf\n", area);

return 0;

}

3.C program to find all angles of a triangle if two angles are given.

Input:-

#include <stdio.h>

int main(void) {

int angle1, angle2, angle3;

printf("Enter the first angle of the triangle: ");

scanf("%d", &angle1);

printf("Enter the second angle of the triangle: ");

scanf("%d", &angle2);

angle3 = 180 - (angle1 + angle2);

printf("The third angle of the triangle is: %d\n", angle3);

return 0;

}

4. C program to convert days in to years, weeks and days.

Input :-

#include <stdio.h>

int main()

{

int days, years, weeks;

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

scanf("%d", &days);

years = days / 365;

weeks = (days % 365) / 7;

days = (days % 365) % 7;

printf("Years: %d\n", years);

printf("Weeks: %d\n", weeks);

printf("Days: %d\n", days);

return 0;

}

5. C program to find power and square root of any number.

Input:-

#include <stdio.h>

#include <math.h>

int main()

{

double num, power, square\_root;

printf("Enter a number: ");

scanf("%lf", &num);

power = pow(num, 2);

square\_root = sqrt(num);

printf("Square: %.2lf\n", power);

printf("Square root: %.2lf\n", square\_root);

return 0;

}

6. C program to calculate total, average and percentage and grades of five subjects.

Input:-

#include <stdio.h>

int main() {

float marks[5];

float total = 0;

float average;

float percentage;

int i;

for (i = 0; i < 5; i++) {

printf("Enter marks for subject %d: ", i+1);

scanf("%f", &marks[i]);

total += marks[i];

}

average = total / 5;

percentage = (total / 500) \* 100;

printf("Total marks: %.2f\n", total);

printf("Average marks: %.2f\n", average);

printf("Percentage: %.2f\n", percentage);

if (percentage >= 90) {

printf("Grade: A+\n");

} else if (percentage >= 80) {

printf("Grade: A\n");

} else if (percentage >= 70) {

printf("Grade: B+\n");

} else if (percentage >= 60) {

printf("Grade: B\n");

} else if (percentage >= 50) {

printf("Grade: C+\n");

} else if (percentage >= 40) {

printf("Grade: C\n");

} else {

printf("Grade: D\n");

} return 0;}

7. C program to check Least Significant Bit (LSB) and MSB of a number using bitwise operator.

Input:-

#include <stdio.h>

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

// Check least significant bit (LSB)

if (num & 1) {

printf("LSB is set (1).\n");

} else {

printf("LSB is not set (0).\n");

}

// Check most significant bit (MSB)

if (num & (1 << (sizeof(int) \* 8 - 1))) {

printf("MSB is set (1).\n");

} else {

printf("MSB is not set (0).\n");

}

return 0;

}

8. C program to swap two numbers USING 3RD VARIABLE AND WITHOUT 3RD VARIABLE.

Input:-

#include <stdio.h>

int main() {

int a, b, temp;

printf("Enter two numbers: ");

scanf("%d%d", &a, &b);

printf("Before swapping: a = %d, b = %d\n", a, b);

// Swap using a third variable

temp = a;

a = b;

b = temp;

printf("After swapping: a = %d, b = %d\n", a, b);

return 0;

}

9. C program to find maximum between three numbers using conditional operator AND Ternary Operator.

#include <stdio.h>

int main() {

int a, b, c;

printf("Enter three numbers: ");

scanf("%d%d%d", &a, &b, &c);

int max = a > b ? (a > c ? a : c) : (b > c ? b : c);

printf("Maximum: %d\n", max);

return 0;

}

10. C program to check alphabet, digit or special character using Conditional operator.

Input:-

#include <stdio.h>

int main()

{

char ch;

printf("Enter a character: ");

scanf("%c", &ch);

(ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z') ? printf("'%c' is an alphabet.\n", ch) :

(ch >= '0' && ch <= '9') ? printf("'%c' is a digit.\n", ch) :

printf("'%c' is a special character.\n", ch);

return 0;

}

11. C program to calculate total electricity bill.

Input:-

#include <stdio.h>

int main()

{

int units;

float bill;

printf("Enter the number of units consumed: ");

scanf("%d", &units);

if (units <= 50)

bill = units \* 0.50;

else if (units <= 150)

bill = 25 + (units - 50) \* 0.75;

else if (units <= 250)

bill = 100 + (units - 150) \* 1.20;

else

bill = 220 + (units - 250) \* 1.50;

printf("Total electricity bill: $%.2f\n", bill);

return 0;

}

12. C program to create Simple Calculator AND Days of week using switch case.

Input:-

#include <stdio.h>

int main() {

int num1, num2, result;

char operator;

int day;

// Simple calculator

printf("Enter an operator (+, -, \*, /): ");

scanf("%c", &operator);

printf("Enter two operands: ");

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

switch (operator) {

case '+':

result = num1 + num2;

break;

case '-':

result = num1 - num2;

break;

case '\*':

result = num1 \* num2;

break;

case '/':

result = num1 / num2;

break;

default:

printf("Error! operator is not correct");

return 0;

}

printf("%.1d %c %.1d = %.1d", num1, operator, num2, result);

// Days of the week

printf("\nEnter a number between 1 and 7: ");

scanf("%d", &day);

switch (day) {

case 1:

printf("Monday");

break;

case 2:

printf("Tuesday");

break;

case 3:

printf("Wednesday");

break;

case 4:

printf("Thursday");

break;

case 5:

printf("Friday");

break;

case 6:

printf("Saturday");

break;

case 7:

printf("Sunday");

break;

default:

printf("Error! Invalid number entered");

}

return 0;

}

13. C program to check vowel or consonant using switch case.

Input:-

#include <stdio.h>

int main() {

char ch;

printf("Enter a character: ");

scanf("%c", &ch);

switch (ch) {

case 'a':

case 'e':

case 'i':

case 'o':

case 'u':

case 'A':

case 'E':

case 'I':

case 'O':

case 'U':

printf("%c is a vowel.", ch);

break;

default:

printf("%c is a consonant.", ch);

}

return 0;

}

14. C program to check positive negative or zero using switch case.

Input:-

#include <stdio.h>

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

switch (num > 0) {

case 1:

printf("%d is positive.", num);

break;

case 0:

switch (num == 0) {

case 1:

printf("%d is zero.", num);

break;

case 0:

printf("%d is negative.", num);

}

break;

}

return 0;

}

15. C program to check whether a triangle is Equilateral, Isosceles or Scalene.

Input :-

#include <stdio.h>

int main() {

int side1, side2, side3;

printf("Enter the sides of the triangle: ");

scanf("%d %d %d", &side1, &side2, &side3);

switch (side1 == side2) {

case 1:

switch (side2 == side3) {

case 1:

printf("The triangle is equilateral.");

break;

case 0:

printf("The triangle is isosceles.");

}

break;

case 0:

switch (side2 == side3) {

case 1:

printf("The triangle is isosceles.");

break;

case 0:

printf("The triangle is scalene.");

}

}

return 0;

}

16. C program to print all natural numbers AND sum of it from 1 to n.

Input:-

#include <stdio.h>

int main(void) {

int n, i, sum = 0;

printf("Enter a positive integer: ");

scanf("%d", &n);

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

printf("%d ", i);

sum += i;

}

printf("\nSum = %d", sum);

return 0;

}

17. C program to print all even numbers AND sum of it from 1 to n.

Input:-

#include <stdio.h>

int main(void) {

int n, i, sum = 0;

printf("Enter a positive integer: ");

scanf("%d", &n);

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

printf("%d ", i);

sum += i;

}

printf("\nSum = %d", sum);

return 0;

}

18. C program to print multiplication table of a number.

Input:-

#include <stdio.h>

int main(void) {

int n, i;

printf("Enter a number: ");

scanf("%d", &n);

for (i = 1; i <= 10; ++i) {

printf("%d x %d = %d\n", n, i, n \* i);

}

return 0;

}

19. C program to calculate factorial of a number.

Input :-

#include <stdio.h>

int main(void) {

int n, i, fact = 1;

printf("Enter a positive integer: ");

scanf("%d", &n);

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

fact \*= i;

}

printf("Factorial of %d = %d", n, fact);

return 0;

}

20. C program to check whether a number is palindrome or not.

Input :-

#include <stdio.h>

int main(void) {

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;

}

if (original == reversed) {

printf("%d is a palindrome.", original);

} else {

printf("%d is not a palindrome.", original);

}

return 0;

}

21. C program to count frequency of digits in a given number.

Input :-

#include <stdio.h>

int main(void) {

int n, digit, count[10] = {0};

printf("Enter a number: ");

scanf("%d", &n);

while (n > 0) {

digit = n % 10;

count[digit]++;

n /= 10;

}

printf("Digit:\t\t");

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

printf("%d\t", i);

}

printf("\nFrequency:\t");

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

printf("%d\t", count[i]);

}

return 0;

}

22. C program to find HCF(GCD) AND LCM of two numbers.

Input :-

#include <stdio.h>

int main(void) {

int n1, n2, i, gcd, lcm;

printf("Enter two positive integers: ");

scanf("%d %d", &n1, &n2);

for (i = 1; i <= n1 && i <= n2; ++i) {

if (n1 % i == 0 && n2 % i == 0) {

gcd = i;

}

}

lcm = (n1 \* n2) / gcd;

printf("GCD of %d and %d is %d\n", n1, n2, gcd);

printf("LCM of %d and %d is %d\n", n1, n2, lcm);

return 0;

}

23. C program to print all prime numbers between 1 to n.

Input :-

#include <stdio.h>

#include <stdbool.h>

int main(void) {

int n, i, j;

bool is\_prime;

printf("Enter a positive integer: ");

scanf("%d", &n);

printf("All prime numbers between 1 and %d are:\n", n);

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

is\_prime = true;

for (j = 2; j <= i / 2; ++j) {

if (i % j == 0) {

is\_prime = false;

break;

}

}

if (is\_prime) {

printf("%d ", i);

}

}

return 0;

}

24. C program to print all Strong Numbers between 1 to n.

Input :-

#include <stdio.h>

#include <stdbool.h>

int main(void) {

int n, i, j, last\_digit, sum;

bool is\_strong;

printf("Enter a positive integer: ");

scanf("%d", &n);

printf("All strong numbers between 1 and %d are:\n", n);

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

sum = 0;

j = i;

is\_strong = true;

while (j > 0) {

last\_digit = j % 10;

j /= 10;

int fact = 1;

for (int k = 1; k <= last\_digit; ++k) {

fact \*= k;

}

sum += fact;

}

if (sum == i) {

printf("%d ", i);

}

}

return 0;

}

25. C program to print Fibonacci series up to n terms.

Input :-

#include <stdio.h>

int main(void) {

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

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

scanf("%d", &n);

printf("Fibonacci series: ");

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

printf("%d, ", first);

next = first + second;

first = second;

second = next;

}

return 0;

}

26. C program to print Armstrong numbers from 1 to n AND Check a given number is Armstrong numbers or not.

Input :-

#include <stdio.h>

#include <math.h>

int main(void) {

int n, i, num, last\_digit, digits, sum, original;

bool is\_armstrong;

printf("Enter a positive integer: ");

scanf("%d", &n);

printf("All Armstrong numbers between 1 and %d are:\n", n);

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

sum = 0;

num = i;

original = num;

digits = (int)log10(num) + 1;

while (num > 0) {

last\_digit = num % 10;

sum += pow(last\_digit, digits);

num /= 10;

}

if (original == sum) {

printf("%d ", original);

}

}

printf("\nEnter a number to check if it is an Armstrong number: ");

scanf("%d", &num);

original = num;

sum = 0;

digits = (int)log10(num) + 1;

while (num > 0) {

last\_digit = num % 10;

sum += pow(last\_digit, digits);

num /= 10;

}

is\_armstrong = (original == sum);

printf("%d is ", original);

if (is\_armstrong) {

printf("an Armstrong number.");

} else {

printf("not an Armstrong number.");

}

return 0;

}

27. C program to print all Perfect numbers between 1 to n AND Check a given number is Perfect numbers or not.

Input :-

#include <stdio.h>

int main(void) {

int n, i, j, sum, num;

bool is\_perfect;

printf("Enter a positive integer: ");

scanf("%d", &n);

printf("All perfect numbers between 1 and %d are:\n", n);

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

sum = 0;

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

if (i % j == 0) {

sum += j;

}

}

if (sum == i) {

printf("%d ", i);

}

}

printf("\nEnter a number to check if it is a perfect number: ");

scanf("%d", &num);

sum = 0;

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

if (num % i == 0) {

sum += i;

}

}

is\_perfect = (sum == num);

printf("%d is ", num);

if (is\_perfect) {

printf("a perfect number.");

} else {

printf("not a perfect number.");

}

return 0;

}

28. C program to find power of any number using for loop.

Input :-

#include <stdio.h>

int main(void) {

int base, exponent, result = 1;

printf("Enter base: ");

scanf("%d", &base);

printf("Enter exponent: ");

scanf("%d", &exponent);

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

result \*= base;

}

printf("%d to the power of %d is %d\n", base, exponent, result);

return 0;

}

29. C program to print ASCII values of all characters.

Input :-

#include <stdio.h>

int main(void) {

printf("ASCII values of all characters:\n");

for (int i = 0; i < 256; ++i) {

printf("%d: %c\n", i, i);

}

return 0;

}

30. C program to print Pascal triangle up to n rows.

Input:-

#include <stdio.h>

int main()

{

int rows, coef = 1, space, i, j;

printf("Enter number of rows: ");

scanf("%d", &rows);

for(i=0; i<rows; i++)

{

for(space=1; space <= rows-i; space++)

printf(" ");

for(j=0; j <= i; j++)

{

if (j==0 || i==0)

coef = 1;

else

coef = coef\*(i-j+1)/j;

printf("%4d", coef);

}

printf("\n");

}

return 0;

}

31. C program to find sum of all elements of array.

Input:-

#include <stdio.h>

int main()

{

int arr[100], size, i, sum = 0;

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

scanf("%d", &size);

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

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

{

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

sum = sum + arr[i];

}

printf("Sum of all elements of array = %d", sum);

return 0;

}

32. C program to copy one array to another array.

Input :-

#include <stdio.h>

int main()

{

int source[100], target[100], size, i;

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

scanf("%d", &size);

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

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

{

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

}

// Copying elements of source array to target array

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

{

target[i] = source[i];

}

printf("Elements of target array: ");

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

{

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

}

return 0;

}

33. C program to insert an element in array at specified position.

Input :-

#include <stdio.h>

int main()

{

int arr[100], size, i, pos, element;

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

scanf("%d", &size);

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

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

{

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

}

printf("Enter the position where you want to insert an element: ");

scanf("%d", &pos);

printf("Enter the element you want to insert: ");

scanf("%d", &element);

// Shifting elements of array after position to right

for(i=size-1; i>=pos; i--)

{

arr[i+1] = arr[i];

}

// Inserting element at given position

arr[pos] = element;

printf("Array after insertion: ");

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

{

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

}

return 0;

}

34. C program to delete an element in array at specified position.

Input :-

#include <stdio.h>

int main()

{

int arr[100], size, i, pos;

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

scanf("%d", &size);

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

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

{

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

}

printf("Enter the position of the element you want to delete: ");

scanf("%d", &pos);

// Shifting elements after the element to be deleted to left

for(i=pos-1; i<size-1; i++)

{

arr[i] = arr[i+1];

}

printf("Array after deleting an element: ");

for(i=0; i<size-1; i++)

{

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

}

return 0;

}

35. C program to search element in array using Linear Search.

Input :-

#include <stdio.h>

#define ARRAY\_SIZE 10

int main(void) {

int array[ARRAY\_SIZE] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

int search\_key = 5;

int found = 0;

int i;

for (i = 0; i < ARRAY\_SIZE; i++) {

if (array[i] == search\_key) {

found = 1;

break;

}

}

if (found) {

printf("The search key was found at index %d\n", i);

} else {

printf("The search key was not found in the array\n");

}

return 0;

}

36. C program to find second largest number and Sorting Using Bubble sort in an array.

Input:-

#include <stdio.h>

#define ARRAY\_SIZE 10

void bubble\_sort(int array[], int size) {

int i, j;

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

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

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

int temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;

}

}

}

}

int main(void) {

int array[ARRAY\_SIZE] = {9, 8, 7, 6, 5, 4, 3, 2, 1, 0};

int second\_largest;

bubble\_sort(array, ARRAY\_SIZE);

second\_largest = array[ARRAY\_SIZE - 2];

printf("The second largest number is %d\n", second\_largest);

return 0;

}

37. C program to count total number of duplicate elements in an array.

Input:-

#include <stdio.h>

#define ARRAY\_SIZE 10

int main(void) {

int array[ARRAY\_SIZE] = {1, 2, 3, 3, 4, 4, 4, 5, 5, 5};

int i, j;

int count = 0;

for (i = 0; i < ARRAY\_SIZE; i++) {

for (j = i + 1; j < ARRAY\_SIZE; j++) {

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

count++;

break;

}

}

}

printf("The total number of duplicate elements is %d\n", count);

return 0;

}

38. C program to perform scalar matrix multiplication.

Input:-

#include <stdio.h>

#define ROWS 3

#define COLUMNS 3

int main(void) {

int matrix[ROWS][COLUMNS] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

int scalar = 2;

int i, j;

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

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

matrix[i][j] \*= scalar;

}

}

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

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

printf("%d ", matrix[i][j]);

}

printf("\n");

}

return 0;

}

39. C program to find sum of main diagonal elements of a matrix.

Input:-

#include <stdio.h>

#define ROWS 3

#define COLUMNS 3

int main(void) {

int matrix[ROWS][COLUMNS] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

int i;

int sum = 0;

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

sum += matrix[i][i];

}

printf("The sum of the main diagonal elements is %d\n", sum);

return 0;

}

40. C program to check sparse AND transpose matrix.

Input:-

#include <stdio.h>

#define ROWS 5

#define COLUMNS 5

int main(void) {

int matrix[ROWS][COLUMNS] = {{0, 1, 0, 0, 0},

{0, 0, 2, 0, 0},

{0, 0, 0, 3, 0},

{0, 0, 0, 0, 4},

{0, 0, 0, 0, 0}};

int i, j;

int is\_sparse = 1;

int is\_transpose = 1;

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

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

if (matrix[i][j] != 0) {

is\_sparse = 0;

break;

}

}

}

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

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

if (matrix[i][j] != matrix[j][i]) {

is\_transpose = 0;

break;

}

}

}

if (is\_sparse && is\_transpose) {

printf("The matrix is both sparse and transpose\n");

} else {

printf("The matrix is not both sparse and transpose\n");

}

return 0;

}

41. C program to check whether a matrix is Identity matrix or not.

Input:-

#include <stdio.h>

#define ROWS 3

#define COLUMNS 3

int main(void) {

int matrix[ROWS][COLUMNS] = {{1, 0, 0},

{0, 1, 0},

{0, 0, 1}};

int i, j;

int is\_identity = 1;

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

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

if (i == j && matrix[i][j] != 1) {

is\_identity = 0;

break;

} else if (i != j && matrix[i][j] != 0) {

is\_identity = 0;

break;

}

}

}

if (is\_identity) {

printf("The matrix is an identity matrix\n");

} else {

printf("The matrix is not an identity matrix\n");

}

return 0;

}

42. C program to merge two sorted array in ascending order.

Input:-

#include <stdio.h>

#define ARRAY\_SIZE\_1 5

#define ARRAY\_SIZE\_2 8

#define MERGED\_ARRAY\_SIZE (ARRAY\_SIZE\_1 + ARRAY\_SIZE\_2)

int main(void) {

int array\_1[ARRAY\_SIZE\_1] = {1, 3, 5, 7, 9};

int array\_2[ARRAY\_SIZE\_2] = {2, 4, 6, 8, 10, 12, 14, 16};

int merged\_array[MERGED\_ARRAY\_SIZE];

int i = 0, j = 0, k = 0;

while (i < ARRAY\_SIZE\_1 && j < ARRAY\_SIZE\_2) {

if (array\_1[i] < array\_2[j]) {

merged\_array[k] = array\_1[i];

i++;

} else {

merged\_array[k] = array\_2[j];

j++;

}

k++;

}

while (i < ARRAY\_SIZE\_1) {

merged\_array[k] = array\_1[i];

i++;

k++;

}

while (j < ARRAY\_SIZE\_2) {

merged\_array[k] = array\_2[j];

j++;

k++;

}

for (i = 0; i < MERGED\_ARRAY\_SIZE; i++) {

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

}

printf("\n");

return 0;

}

43. All Operations of String.

#include<stdio.h>

#include<conio.h>

Void main(){

Char string1[25],string2[25];

int l;

Clrscr();

Printf(“\*\*\*\*\* performing string length \*\*\*\*\*\*\n”);

Printf(“enter only one string \n”);

Scanf(“%s”,string1);

l = strlen(string1);

printf(“the string length is %d\n\n”,l);

printf(“\*\*\*\* performing string concatenation \*\*\*\*\n”);

printf(“enter two strings\n”);

scanf(“%s%s”,string1,string2);

printf(“the concatenated string is %s\n\n”,strcat(string1,string2));

printf(“\*\*\*\*\* performing string compare \*\*\*\*\*\n”);

printf(“enter two strings \n”);

scanf(“%s%s”,string1,string2);

if(strcmp(string1,string2) = = 0)

printf(“strings are equal\n”);

else

printf(“strings are not equal\n”);

printf(“\*\*\* performing string copy \*\*\*\*\n”);

printf(“enter the two strings\n”);

scanf(“%d%d”,string1,string2);

printf(“the first string is %s and second string is %s\n”,string1,string2);

strcpy(string1,string2);

printf(“the first string is %s and second string is %s\n”,string1,string2);

getch();}

44. C program to check whether a string is palindrome or not without Compare Function of String.

Input:-

#include <stdio.h>

#include <string.h>

int main()

{

    char str[] = { "abbba" };

    // Start from leftmost and

    // rightmost corners of str

    int l = 0;

    int h = strlen(str) - 1;

    // Keep comparing characters

    // while they are same

    while (h > l) {

        if (str[l++] != str[h--]) {

            printf("%s is not a palindrome\n", str);

            return 0;

            // will return from here

        }

    }

    printf("%s is a palindrome\n", str);

    return 0;

}

45. C program to count frequency of each character in a string.

Input:-

#include <stdio.h>

int main() {

char str[1000], ch;

int count = 0;

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

printf("Enter a character to find its frequency: ");

scanf("%c", &ch);

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

if (ch == str[i])

++count;

}

printf("Frequency of %c = %d", ch, count);

return 0;

}

46. C program to find diameter, circumference and area of a circle using functions.

Input:-

#include <stdio.h>

#include <math.h> // Used for constant PI referred as M\_PI

/\* Function declaration \*/

double getDiameter(double radius);

double getCircumference(double radius);

double getArea(double radius);

int main()

{

float radius, dia, circ, area;

/\* Input radius of circle from user \*/

printf("Enter radius of circle: ");

scanf("%f", &radius);

dia = getDiameter(radius); // Call getDiameter function

circ = getCircumference(radius); // Call getCircumference function

area = getArea(radius); // Call getArea function

printf("Diameter of the circle = %.2f units\n", dia);

printf("Circumference of the circle = %.2f units\n", circ);

printf("Area of the circle = %.2f sq. units", area);

return 0;

}

/\*\*

\* Calculate diameter of circle whose radius is given

\*/

double getDiameter(double radius)

{

return (2 \* radius);

}

/\*\*

\* Calculate circumference of circle whose radius is given

\*/

double getCircumference(double radius)

{

return (2 \* M\_PI \* radius); // M\_PI = PI = 3.14 ...

}

/\*\*

\* Find area of circle whose radius is given

\*/

double getArea(double radius)

{

return (M\_PI \* radius \* radius); // M\_PI = PI = 3.14 ...

}

47. C program to check prime, armstrong and perfect numbers using functions.

Input:-

#include <stdio.h>

#include <math.h>

/\* Function declarations \*/

int isPrime(int num);

int isArmstrong(int num);

int isPerfect(int num);

int main()

{

int num;

printf("Enter any number: ");

scanf("%d", &num);

// Call isPrime() functions

if(isPrime(num))

{

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

}

else

{

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

}

// Call isArmstrong() function

if(isArmstrong(num))

{

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

}

else

{

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

}

// Call isPerfect() function

if(isPerfect(num))

{

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

}

else

{

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

}

return 0;

}

/\*\*

\* Check whether a number is prime or not.

\* Returns 1 if the number is prime otherwise 0.

\*/

int isPrime(int num)

{

int i;

for(i=2; i<=num/2; i++)

{

/\*

\* If the number is divisible by any number

\* other than 1 and self then it is not prime

\*/

if(num%i == 0)

{

return 0;

}

}

return 1;

}

/\*\*

\* Check whether a number is Armstrong number or not.

\* Returns 1 if the number is Armstrong number otherwise 0.

\*/

int isArmstrong(int num)

{

int lastDigit, sum, originalNum, digits;

sum = 0;

originalNum = num;

/\* Find total digits in num \*/

digits = (int) log10(num) + 1;

/\*

\* Calculate sum of power of digits

\*/

while(num > 0)

{

// Extract the last digit

lastDigit = num % 10;

// Compute sum of power of last digit

sum = sum + round(pow(lastDigit, digits));

// Remove the last digit

num = num / 10;

}

return (originalNum == sum);

}

/\*\*

\* Check whether the number is perfect number or not.

\* Returns 1 if the number is perfect otherwise 0.

\*/

int isPerfect(int num)

{

int i, sum, n;

sum = 0;

n = num;

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

{

/\* If i is a divisor of num \*/

if(n%i == 0)

{

sum += i;

}

}

return (num == sum);

}

48. C program to add two number using pointers.

Input:-

#include <stdio.h>

int main()

{

int first, second, \*p, \*q, sum;

printf("Enter two integers to add\n");

scanf("%d%d", &first, &second);

p = &first;

q = &second;

sum = \*p + \*q;

printf("Sum of entered numbers = %d\n",sum);

return 0;

}

49. Swap 2 numbers using Call by Value AND Call by reference.

Input:-

#include <stdio.h>

void swap(int\*, int\*);

int main()

{

int x, y;

printf("Enter the value of x and y\n");

scanf("%d%d",&x,&y);

printf("Before Swapping\nx = %d\ny = %d\n", x, y);

swap(&x, &y);

printf("After Swapping\nx = %d\ny = %d\n", x, y);

return 0;

}

void swap(int \*a, int \*b)

{

int temp;

temp = \*b;

\*b = \*a;

\*a = temp;

}

50. C program to copy an array to another array AND reverse an array using pointers.

Input:-

#include <stdio.h>

#define MAX\_SIZE 100

/\* Function declaration \*/

void printArr(int \*arr, int size);

int main()

{

int arr[MAX\_SIZE];

int size;

int \*left = arr; // Pointer to arr[0]

int \*right;

// Input size of array

printf("Enter size of array: ");

scanf("%d", &size);

right = &arr[size - 1]; // Pointer to arr[size - 1]

/\*

\* Input elements in array

\*/

printf("Enter elements in array: ");

while(left <= right)

{

scanf("%d", left++);

}

printf("\nArray before reverse: ");

printArr(arr, size);

// Make sure that left points to arr[0]

left = arr;

// Loop to reverse array

while(left < right)

{

/\*

\* Swap element from left of array to right of array.

\*/

\*left ^= \*right;

\*right ^= \*left;

\*left ^= \*right;

// Increment left array pointer and decrement right array pointer

left++;

right--;

}

printf("\nArray after reverse: ");

printArr(arr, size);

return 0;

}

void printArr(int \* arr, int size)

{

// Pointer to arr[size - 1]

int \* arrEnd = (arr + size - 1);

/\* Loop till last array element \*/

while(arr <= arrEnd)

{

printf("%d, ", \*arr);

// Move pointer to next array element.

arr++;

}

}

51. Pattern programme-1

[\*\*\*\*\*](javascript:%20void(0);)

[\*\*\*\*\*](javascript:%20void(0);)

[\*\*\*\*\*](javascript:%20void(0);)

[\*\*\*\*\*](javascript:%20void(0);)

[\*\*\*\*\*](javascript:%20void(0);)

Input:-

#include <stdio.h>

int main()

{

int i, j, N;

/\* Input number of rows from user \*/

printf("Enter number of rows: ");

scanf("%d", &N);

/\* Iterate through N rows \*/

for(i=1; i<=N; i++)

{

/\* Iterate over columns \*/

for(j=1; j<=N; j++)

{

/\* Print star for each column \*/

printf("\*");

}

/\* Move to the next line/row \*/

printf("\n");

}

return 0;

}

52. Pattern programme -2.

[\*](javascript:%20void(0);)

[\*\*](javascript:%20void(0);)

[\*\*\*](javascript:%20void(0);)

[\*\*\*\*](javascript:%20void(0);)

[\*\*\*\*\*](javascript:%20void(0);)

Input:-

#include <stdio.h>

int main()

{

int i, j, n;

/\* Input number of rows from user \*/

printf("Enter value of n: ");

scanf("%d", &n);

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

{

/\* Print i number of stars \*/

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

{

printf("\*");

}

/\* Move to next line \*/

printf("\n");

}

return 0;

}

[53. Pattern Program-3.](javascript:%20void(0);)

[\*  
 \*\*  
 \*\*\*  
 \*\*\*\*  
\*\*\*\*\*](javascript:%20void(0);)

Input:-

#include <stdio.h>

int main()

{

int i, j, rows;

/\* Input rows from user \*/

printf("Enter number of rows: ");

scanf("%d", &rows);

/\* Iterate through rows \*/

for(i=1; i<=rows; i++)

{

/\* Print spaces in decreasing order of row \*/

for(j=i; j<rows; j++)

{

printf(" ");

}

/\* Print star in increasing order or row \*/

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

{

printf("\*");

}

/\* Move to next line \*/

printf("\n");

}

return 0;

}

54. Pattern program-4.

11111

11111

11111

11111

11111

Input:-

#include <stdio.h>

int main()

{

int rows, cols, i, j;

/\* Input rows and columns from user \*/

printf("Enter number of rows: ");

scanf("%d", &rows);

printf("Enter number of columns: ");

scanf("%d", &cols);

/\* Iterate through rows \*/

for(i=1; i<=rows; i++)

{

/\* Iterate through columns \*/

for(j=1; j<=cols; j++)

{

printf("1");

}

printf("\n");

}

return 0;

}

55. Pattern program-5.

11111

00000

11111

00000

11111

Input:-

#include <stdio.h>

int main()

{

int rows, cols, i, j;

/\* Input rows and columns from user \*/

printf("Enter number of rows: ");

scanf("%d", &rows);

printf("Enter number of columns: ");

scanf("%d", &cols);

for(i=1; i<=rows; i++)

{

for(j=1; j<=cols; j++)

{

// Print 1 if current row is odd

if(i%2 == 1)

{

printf("1");

}

else

{

printf("0");

}

}

printf("\n");

}

return 0;

}