1. Write a C program to print numbers from 0 to 100. (You are required to write 3 separate answers each using While, Do..While, For, looping structures)

* While

#include <stdio.h>

int main() {

    int number = 0;

    while (number <= 100) {

        printf("%d ", number);

        number++;

    }

    return 0;

}

* Do while

#include <stdio.h>

int main() {

    int number = 0;

    do {

        printf("%d ", number);

        number++;

    } while (number <= 100);

    return 0;

}

* For

#include <stdio.h>

int main() {

    for (int number = 0; number <= 100; number++) {

        printf("%d ", number);

    }

    return 0;

}

1. Write a C program to calculate and print the total of 10 marks and the average. If the average is less than 50 program should print “Fail!” otherwise “Pass!”

#include <stdio.h>

int main() {

    int marks[10];

    int total = 0;

    printf("Enter 10 marks:\n");

    for (int i = 0; i < 10; i++) {

        scanf("%d", &marks[i]);

        total += marks[i];

    }

    float average = (float)total / 10;

    printf("Total: %d\n", total);

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

    if (average < 50) {

        printf("Fail!\n");

    } else {

        printf("Pass!\n");

    }

    return 0;

}

1. Write a C program to calculate factorial of a user given number.

Hint:

* + Select an appropriate looping structure.
  + Factorial of ‘0’ is ‘1’ (0! = 1)
  + Ex: factorial of number 5 is calculated as 5! = 5\*4\*3\*2\*1

#include <stdio.h>

int main() {

    int number;

    int factorial = 1;

    printf("Enter a number: ");

    scanf("%d", &number);

    if (number < 0) {

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

    } else {

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

            factorial \*= i;

        }

        printf("Factorial of %d is %d\n", number, factorial);

    }

    return 0;

}

1. Write a C program to calculate the sum of all digits of a user given number.

If user input 123 your program should output 6. (calculated as 1+2+3)

#include <stdio.h>

int main() {

    int number, sum = 0;

    printf("Enter a number: ");

    scanf("%d", &number);

    int remainder;

    while (number > 0) {

        remainder = number % 10;

        sum += remainder;

        number /= 10;

    }

    printf("Sum of digits: %d\n", sum);

    return 0;

}

1. Write a C program to reverse the digits of a number using do-while statement.

#include <stdio.h>

int main() {

    int number, reversedNumber = 0, remainder;

    printf("Enter a number: ");

    scanf("%d", &number);

    do {

        remainder = number % 10;

        reversedNumber = reversedNumber \* 10 + remainder;

        number = number / 10;

    } while (number != 0);

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

    return 0;

}

1. Write a C program to calculate nth power of a given integer. The user input base and exponent. (Do NOT use inbuilt functions, instead use a loop)

#include <stdio.h>

int main() {

    int base, exponent, result = 1;

    printf("Enter the base: ");

    scanf("%d", &base);

    printf("Enter the exponent: ");

    scanf("%d", &exponent);

    int i;

    for (i = 0; i < exponent; i++) {

        result \*= base;

    }

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

    return 0;

}

1. Write a C program to print first 10 numbers of “Fibonacci Sequence”.

#include <stdio.h>

int main() {

    int n = 10;

    int fib[n];

    int i;

    fib[0] = 0;

    fib[1] = 1;

    for (i = 2; i < n; i++) {

        fib[i] = fib[i-1] + fib[i-2];

    }

    printf("The first 10 numbers of the Fibonacci sequence are:\n");

    for (i = 0; i < n; i++) {

        printf("%d ", fib[i]);

    }

    printf("\n");

    return 0;

}

1. Write a C program to check whether a given number is an Armstrong Number! (Refer to previous flowcharts)

#include <stdio.h>

int main() {

    int number, originalNumber, remainder, result = 0, n = 0;

    printf("Enter a number: ");

    scanf("%d", &number);

    originalNumber = number;

    while (originalNumber != 0) {

        originalNumber /= 10;

        ++n;

    }

    originalNumber = number;

    while (originalNumber != 0) {

        remainder = originalNumber % 10;

        int power = 1;

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

            power \*= remainder;

        }

        result += power;

        originalNumber /= 10;

    }

    if (result == number)

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

    else

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

    return 0;

}

1. Write a C program to print all the ASCII values for letters A to Z.

#include <stdio.h>

int main() {

    char letter;

    printf("ASCII values for letters A to Z:\n");

    for (letter = 'A'; letter <= 'Z'; ++letter) {

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

    }

    return 0;

}

1. Write a program to print this pattern.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

#include <stdio.h>

int main() {

    int rows = 5;  // number of rows in the pattern

    int i, j;

    for (i = 1; i <= rows; ++i) {

        for (j = 1; j <= i; ++j) {

            printf("\*");

        }

        printf("\n");

    }

    return 0;

}

1. Write a program to check whether a given number is prime or not.

#include <stdio.h>

int main() {

    int number, i, isPrime = 1;

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    if (number == 0 || number == 1) {

        isPrime = 0;

    } else {

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

            if (number % i == 0) {

                isPrime = 0;

                break;

            }

        }

    }

    if (isPrime) {

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

    } else {

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

    }

    return 0;

}

1. Write a C program to print all factors of a given integer.

#include <stdio.h>

int main() {

    int number, i;

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    printf("Factors of %d are: ", number);

    for (i = 1; i <= number; ++i) {

        if (number % i == 0) {

            printf("%d ", i);

        }

    }

    printf("\n");

    return 0;

}

1. Write a C program to add all user inputs until user input ‘-1’. And then display the sum.

#include <stdio.h>

int main() {

    int number;

    int sum = 0;

    printf("Enter numbers to be added (enter -1 to stop):\n");

    while (1) {

        scanf("%d", &number);

        if (number == -1) {

            break;

        }

        sum += number;

    }

    printf("The sum is: %d\n", sum);

    return 0;

}

1. Write a C program to read user inputs for an integer array (size = 10) and print the array.

#include <stdio.h>

int main() {

    int array[10];

    int i;

    printf("Enter 10 integers:\n");

    for (i = 0; i < 10; i++) {

        scanf("%d", &array[i]);

    }

    printf("The entered array is: ");

    for (i = 0; i < 10; i++) {

        printf("%d ", array[i]);

    }

    printf("\n");

    return 0;

}

1. Re-Write the above code to count all the even numbers in above integer array and display the count.

#include <stdio.h>

int main() {

    int array[10];

    int i, count = 0;

    printf("Enter 10 integers:\n");

    for (i = 0; i < 10; i++) {

        scanf("%d", &array[i]);

    }

    for (i = 0; i < 10; i++) {

        if (array[i] % 2 == 0) {

            count++;

        }

    }

    printf("The count of even numbers in the array is: %d\n", count);

    return 0;

}

Section B

1. Input 10 numbers and to output number of positive, number of negative, number of zeros.

#include <stdio.h>

int main() {

    int numbers[10];

    int i, positiveCount = 0, negativeCount = 0, zeroCount = 0;

    printf("Enter 10 numbers:\n");

    for (i = 0; i < 10; i++) {

        scanf("%d", &numbers[i]);

    }

    for (i = 0; i < 10; i++) {

        if (numbers[i] > 0) {

            positiveCount++;

        } else if (numbers[i] < 0) {

            negativeCount++;

        } else {

            zeroCount++;

        }

    }

    printf("Positive numbers: %d\n", positiveCount);

    printf("Negative numbers: %d\n", negativeCount);

    printf("Zeros: %d\n", zeroCount);

    return 0;

}

1. Input Marks of 10 students and output the maximum , minimum and average Marks

#include <stdio.h>

int main() {

    int marks[10];

    int i, totalMarks = 0, maxMarks, minMarks;

    printf("Enter marks of 10 students:\n");

    for (i = 0; i < 10; i++) {

        scanf("%d", &marks[i]);

        totalMarks += marks[i];

        if (i == 0) {

            maxMarks = marks[i];

            minMarks = marks[i];

        } else {

            if (marks[i] > maxMarks) {

                maxMarks = marks[i];

            }

            if (marks[i] < minMarks) {

                minMarks = marks[i];

            }

        }

    }

    double averageMarks = (double) totalMarks / 10;

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

    printf("Minimum Marks: %d\n", minMarks);

    printf("Average Marks: %.2lf\n", averageMarks);

    return 0;

}

1. Input price of 10 items and display the average value of an Item , number of items which the price is greater than 200.

#include <stdio.h>

int main() {

    double prices[10];

    int i, count = 0;

    double total = 0.0;

    printf("Enter prices of 10 items:\n");

    for (i = 0; i < 10; i++) {

        scanf("%lf", &prices[i]);

        total += prices[i];

        if (prices[i] > 200) {

            count++;

        }

    }

    double average = total / 10;

    printf("Average value of an item: %.2lf\n", average);

    printf("Number of items with price > 200: %d\n", count);

    return 0;

}

1. Input the Employee no and the Basic Salary of the Employees in an organisation ending with the dummy value -999 for Employee no and count the number Employees whose Basic Salary >=5000.

#include <stdio.h>

int main() {

    int employeeNo, count = 0;

    double basicSalary;

    printf("Enter employee number and basic salary : \n");

    while (1) {

        scanf("%d", &employeeNo);

        if (employeeNo == -999) {

            break;

        }

        scanf("%lf", &basicSalary);

        if (basicSalary >= 5000) {

            count++;

        }

    }

    printf("Number of employees with a basic salary >= 5000: %d\n", count);

    return 0;

}

1. Input employee number, and hours worked by employees, and to display the following: Employee number, Over Time Payment, and the percentage of employees whose Over Time Payment exceeding the Rs. 4000/-. The user should input –999 as employee number to end the program, and the normal Over Time Rate is Rs.150 per hour and Rs. 200 per hour for hours in excess of 40.

#include <stdio.h>

int main() {

    int employeeNo, count = 0, overtimeCount = 0;

    double hoursWorked, overtimePayment, totalOvertimePayment = 0.0;

    printf("Enter employee number and hours worked :\n");

    scanf("%d", &employeeNo);

    while (employeeNo != -999) {

        scanf("%lf", &hoursWorked);

        if (hoursWorked > 40) {

            overtimePayment = 150 \* 40 + 200 \* (hoursWorked - 40);

        } else {

            overtimePayment = 150 \* hoursWorked;

        }

        printf("Employee number: %d\n", employeeNo);

        printf("Overtime payment: %.2lf\n", overtimePayment);

        totalOvertimePayment += overtimePayment;

        count++;

        if (overtimePayment > 4000) {

            overtimeCount++;

        }

        scanf("%d", &employeeNo);

    }

    double percentageExceeding4000 = (double) overtimeCount / count \* 100;

    printf("\nSummary:\n");

    printf("Total employees: %d\n", count);

    printf("Total overtime payment: %.2lf\n", totalOvertimePayment);

    printf("Percentage of employees with overtime payment exceeding Rs. 4000: %.2lf%%\n", percentageExceeding4000);

}