Q1. Write a program to input n numbers on command line argument and calculate maximum of them.

package assingmentQuestions;

public class MaxNumber {

public static void main(String[] args) {

if (args.length == 0) {

System.out.println("Please provide some numbers as command-line arguments.");

return;

}

// Parse the first argument as an integer

int max = Integer.parseInt(args[0]);

// Loop through the rest of the arguments to find the maximum number

for (int i = 1; i < args.length; i++) {

int number = Integer.parseInt(args[i]);

if (number > max) {

max = number;

}

}

// Output the maximum number

System.out.println("The maximum number is: " + max);

}

}

Q2. Write a program to calculate a Factorial of a number.

package assingmentQuestions;

import java.util.Scanner;

public class factorial {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number: ");

int number = scanner.nextInt();

long factorial = 1;

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

factorial \*= i;

}

System.out.println("The factorial of " + number + " is: " + factorial);

}

}

Q3. Write a program to calculate Fibonacci Series up to n numbers

import java.util.Scanner;

public class fibbonacci {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of terms in the Fibonacci series: ");

int n = scanner.nextInt();

int first = 0, second = 1;

System.out.print("Fibonacci Series up to " + n + " terms: ");

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

System.out.print(first + " ");

// Calculate the next number in the series

int next = first + second;

first = second;

second = next;

}

}

}

Q4. Write a program to calculate the grade of a student. There are five subjects. Marks in each subject are entered from keyboard. Assign grade based on the following rule:

public class studentGrade {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int totalMarks = 0;

// Input marks for 5 subjects

for (int i = 1; i <= 5; i++) {

System.out.print("Enter marks for subject " + i + ": ");

int marks = scanner.nextInt();

totalMarks += marks;

}

// Calculate average marks

int averageMarks = totalMarks / 5;

// Determine the grade based on the average marks

char grade;

if (averageMarks >= 90) {

grade = 'E' + 'x';

} else if (averageMarks >= 80) {

grade = 'A';

} else if (averageMarks >= 70) {

grade = 'B';

} else if (averageMarks >= 60) {

grade = 'C';

} else {

grade = 'F';

}

System.out.println("Total Marks: " + totalMarks);

System.out.println("Average Marks: " + averageMarks);

System.out.println("Grade: " + grade);

}

}

Q5. Write a program to check the input characters for uppercase, lowercase, number of digits and other characters. Display appropriate message.

public class checkChar {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a string: ");

String input = scanner.nextLine();

int upperCaseCount = 0;

int lowerCaseCount = 0;

int digitCount = 0;

int otherCharCount = 0;

// Iterate through each character in the input string

for (char ch : input.toCharArray()) {

if (Character.isUpperCase(ch)) {

upperCaseCount++;

} else if (Character.isLowerCase(ch)) {

lowerCaseCount++;

} else if (Character.isDigit(ch)) {

digitCount++;

} else {

otherCharCount++;

}

}

// Display the results

System.out.println("Uppercase letters: " + upperCaseCount);

System.out.println("Lowercase letters: " + lowerCaseCount);

System.out.println("Digits: " + digitCount);

System.out.println("Other characters: " + otherCharCount);

}

}

Q6. Write a program to perform matrix multiplication.

public class MatrixMultiplication {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input dimensions of the matrices

System.out.print("Enter the number of rows in the first matrix: ");

int rows1 = scanner.nextInt();

System.out.print("Enter the number of columns in the first matrix (and rows in the second matrix): ");

int cols1 = scanner.nextInt();

System.out.print("Enter the number of columns in the second matrix: ");

int cols2 = scanner.nextInt();

// Initialize the matrices

int[][] matrix1 = new int[rows1][cols1];

int[][] matrix2 = new int[cols1][cols2];

int[][] result = new int[rows1][cols2];

// Input elements of the first matrix

System.out.println("Enter the elements of the first matrix:");

for (int i = 0; i < rows1; i++) {

for (int j = 0; j < cols1; j++) {

matrix1[i][j] = scanner.nextInt();

}

}

// Input elements of the second matrix

System.out.println("Enter the elements of the second matrix:");

for (int i = 0; i < cols1; i++) {

for (int j = 0; j < cols2; j++) {

matrix2[i][j] = scanner.nextInt();

}

}

// Perform matrix multiplication

for (int i = 0; i < rows1; i++) {

for (int j = 0; j < cols2; j++) {

for (int k = 0; k < cols1; k++) {

result[i][j] += matrix1[i][k] \* matrix2[k][j];

}

}

}

// Display the result

System.out.println("The result of matrix multiplication is:");

for (int i = 0; i < rows1; i++) {

for (int j = 0; j < cols2; j++) {

System.out.print(result[i][j] + " ");

}

System.out.println();

}

}

}

Q7. Write a program to accept a number from user using command line argument and display its table.

public class MultiplicationTable {

public static void main(String[] args) {

if (args.length == 0) {

System.out.println("Please provide a number as a command-line argument.");

return;

}

// Parse the command-line argument to an integer

int number = Integer.parseInt(args[0]);

// Display the multiplication table

System.out.println("Multiplication Table for " + number + ":");

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

System.out.println(number + " x " + i + " = " + (number \* i));

}

}

}

Q8. Write a program to read the name of a student (studentName), roll Number (rollNo) and marks (totalMarks) obtained. rollNo may be an alphanumeric string. Display the data as read. Hint: Create a Student structure and write appropriate functions.

#include <stdio.h>

// Define a structure to hold student information

struct Student {

char studentName[50];

char rollNo[20];

int totalMarks;

};

// Function to display student details

void displayStudentDetails(struct Student student) {

printf("Student Name: %s\n", student.studentName);

printf("Roll Number: %s\n", student.rollNo);

printf("Total Marks: %d\n", student.totalMarks);

}

int main() {

struct Student student;

// Reading student details

printf("Enter student name: ");

fgets(student.studentName, sizeof(student.studentName), stdin);

printf("Enter roll number: ");

fgets(student.rollNo, sizeof(student.rollNo), stdin);

printf("Enter total marks: ");

scanf("%d", &student.totalMarks);

// Displaying student details

displayStudentDetails(student);

return 0;

}

Q9. Accept an integer number and when the program is executed print the binary, octal and hexadecimal equivalent of the given number.

Sample Output:

Enter Number : 20

Given Number :20

Binary equivalent :10100

Octal equivalent :24

#include <stdio.h>

// Function to print the binary equivalent using bitwise operations

void printBinary(int n) {

printf("Binary equivalent: ");

// Iterate over each bit (assuming a 32-bit integer)

for (int i = 31; i >= 0; i--) {

int bit = (n >> i) & 1; // Right shift and mask with 1 to get the i-th bit

printf("%d", bit);

}

printf("\n");

}

// Recursive function to print the octal equivalent

void printOctal(int n) {

if (n == 0) {

return;

}

printOctal(n / 8); // Recursive call with n divided by 8

printf("%d", n % 8); // Print the remainder when divided by 8

}

// Recursive function to print the hexadecimal equivalent

void printHexadecimal(int n) {

if (n == 0) {

return;

}

printHexadecimal(n / 16); // Recursive call with n divided by 16

int remainder = n % 16;

// Print the corresponding hexadecimal digit

if (remainder < 10) {

printf("%d", remainder);

} else {

printf("%c", 'A' + (remainder - 10));

}

}

int main() {

int number;

// Accept an integer number from the user

printf("Enter Number: ");

scanf("%d", &number);

// Print the given number

printf("Given Number: %d\n", number);

// Print the binary equivalent

printBinary(number);

// Print the octal equivalent

printf("Octal equivalent: ");

if (number == 0) {

printf("0");

} else {

printOctal(number);

}

printf("\n");

// Print the hexadecimal equivalent

printf("Hexadecimal equivalent: ");

if (number == 0) {

printf("0");

} else {

printHexadecimal(number);

}

printf("\n");

return 0;

}

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

Q10. Read at most 10 names of students and store them into an array of char nameOfStudents[10][50]. Sort the array and display them back. Hint: Use qsort() method.

#define MAX\_STUDENTS 10

#define MAX\_NAME\_LENGTH 50

// Comparison function to be used by qsort

int compareNames(const void \*a, const void \*b) {

return strcmp((char \*)a, (char \*)b);

}

int main() {

char nameOfStudents[MAX\_STUDENTS][MAX\_NAME\_LENGTH];

int n;

// Read the number of students

printf("Enter the number of students (max %d): ", MAX\_STUDENTS);

scanf("%d", &n);

// Check if the number of students is within the valid range

if (n > MAX\_STUDENTS || n <= 0) {

printf("Invalid number of students!\n");

return 1;

}

// Read the names of students

printf("Enter the names of the students:\n");

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

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

scanf(" %[^\n]", nameOfStudents[i]);

}

// Sort the array using qsort

qsort(nameOfStudents, n, MAX\_NAME\_LENGTH, compareNames);

// Display the sorted names

printf("\nSorted list of student names:\n");

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

printf("%s\n", nameOfStudents[i]);

}

return 0;

}

Q11. Create a structure called Employee that includes three fields - a first name (type String), a last name (type String) and a monthly salary (double). Write functions to initialize the fields, print them and modify the values in the given object. Example methods:

 void emp\_init(struct emp\* e);

 void set\_salary(struct emp \*e, double sal);

 void emp\_display(struct emp \*e);

Write the test code in the main(). Create two emp objects and display each object’s yearly salary. Then give each Employee a 10% raise and display each Employee’s yearly salary again.

#include <stdio.h>

#include <string.h>

// Define the Employee structure

struct emp {

char firstName[50];

char lastName[50];

double monthlySalary;

};

// Function to initialize an Employee's details

void emp\_init(struct emp \*e, const char \*first, const char \*last, double sal) {

strcpy(e->firstName, first);

strcpy(e->lastName, last);

e->monthlySalary = sal;

}

// Function to set the salary of an Employee

void set\_salary(struct emp \*e, double sal) {

e->monthlySalary = sal;

}

// Function to display Employee details

void emp\_display(struct emp \*e) {

printf("Employee: %s %s\n", e->firstName, e->lastName);

printf("Monthly Salary: $%.2f\n", e->monthlySalary);

printf("Yearly Salary: $%.2f\n\n", e->monthlySalary \* 12);

}

// Function to give a 10% raise to the Employee

void give\_raise(struct emp \*e) {

e->monthlySalary \*= 1.10; // Increase the salary by 10%

}

int main() {

struct emp emp1, emp2;

// Initialize the Employee objects

emp\_init(&emp1, "John", "Doe", 3000.00);

emp\_init(&emp2, "Jane", "Smith", 3500.00);

// Display initial details

printf("Initial details:\n");

emp\_display(&emp1);

emp\_display(&emp2);

// Give each Employee a 10% raise

give\_raise(&emp1);

give\_raise(&emp2);

// Display updated details

printf("After 10%% raise:\n");

emp\_display(&emp1);

emp\_display(&emp2);

return 0;

}

Q12. Write a Program to reverse the letters present in the given String. Do not use strrev() function.

#include <stdio.h>

#include <string.h>

// Function to reverse the string

void reverseString(char str[]) {

int n = strlen(str); // Get the length of the string

int start = 0;

int end = n - 1;

char temp;

// Swap characters from the start and end of the string

while (start < end) {

// Swap the characters

temp = str[start];

str[start] = str[end];

str[end] = temp;

// Move the pointers towards the center

start++;

end--;

}

}

int main() {

char str[100];

// Input the string

printf("Enter a string: ");

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

// Remove newline character if present (due to fgets)

str[strcspn(str, "\n")] = '\0';

// Reverse the string

reverseString(str);

// Output the reversed string

printf("Reversed string: %s\n", str);

return 0;

}

Q13. Declare an Array of type char\* (Strings) and initialize it with a few strings (hard coded). Display the strings which are duplicated in that array. (Hint: use strcmp()).

#include <stdio.h>

#include <string.h>

int main() {

// Initialize an array of strings

char \*strArray[] = {

"apple",

"banana",

"orange",

"apple",

"grape",

"banana",

"kiwi",

"apple",

"kiwi"

};

int n = sizeof(strArray) / sizeof(strArray[0]); // Number of strings in the array

int i, j;

int duplicateFound;

printf("Duplicate strings in the array are:\n");

// Iterate through the array and check for duplicates

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

duplicateFound = 0;

// Check if the string has already been printed as a duplicate

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

if (strcmp(strArray[i], strArray[j]) == 0) {

duplicateFound = 1;

break;

}

}

// If it's the first time this string is found to be a duplicate, print it

if (!duplicateFound) {

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

if (strcmp(strArray[i], strArray[j]) == 0) {

printf("%s\n", strArray[i]);

break;

}

}

}

}

return 0;

}

Q14 Write a code to check if string is palindrome.

#include <stdio.h>

#include <string.h>

int isPalindrome(char str[]) {

int low = 0;

int high = strlen(str) - 1;

// Keep comparing characters while they are same

while (low < high) {

if (str[low] != str[high]) {

return 0; // not a palindrome.

}

low++; // move the low index forward

high--; // move the high index backwards

}

return 1; // is a palindrome

}

int main() {

char str[] = "abbba";

char str1[] = "abcded";

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

printf("%s is palindrome %d\n", str1, isPalindrome(str1));

return 0;

}

Q15. Input a string from the user. Count occurrences (case insensitive) of each alphabet in the string.

#include <stdio.h>

#include <ctype.h>

#define ALPHABET\_SIZE 26

int main() {

char str[1000];

int count[ALPHABET\_SIZE] = {0};

printf("Input a string: ");

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

// Normalize the string to lowercase and count occurrences

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

char ch = tolower(str[i]);

if (ch >= 'a' && ch <= 'z') {

count[ch - 'a']++;

}

}

// Print occurrences of each alphabet letter

printf("Output:\n");

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

if (count[i] > 0) {

printf("%c : %d\n", 'A' + i, count[i]);

}

}

return 0;

}