

ASSIGNMENT

(C – PROGRAMMING)

DIVYANKA SAXENA

B. TECH 1ST YEAR

ROLL NO. 26

SECTION: AU

Q1. Write a C program for calculating the price of a product after adding the sales tax to its original price. Where rate of tax and price is inputted by user.

PROGRAM:

```
#include <stdio.h>

int main() {
    float originalPrice, taxRate, totalPrice;
    printf("Enter the original price: $");
    scanf("%f", &originalPrice);
    printf("Enter the sales tax rate (in percentage): ");
    scanf("%f", &taxRate);
    totalPrice = originalPrice + (originalPrice * (taxRate / 100));
    printf("Total price after adding %.2f%% sales tax: $%.2f\n", taxRate, totalPrice);
    return 0;
}
```

Q2. Write a C program to calculate the weekly wages of an employee. The pay depends on wages per hour and number of hours worked. Moreover, if the employee has

worked for more than 30 hours, then he or she gets twice the wages per hour, for every extra hour that he or she has worked.

PROGRAM:

```
#include <stdio.h>

int main() {
    float hourlyWage, weeklyWages;
    int hoursWorked;
    printf("Enter hourly wage: $");
    scanf("%f", &hourlyWage);

    printf("Enter hours worked in the week: ");
    scanf("%d", &hoursWorked);
    if (hoursWorked <= 30) {
        weeklyWages = hourlyWage * hoursWorked;
    } else {
        int regularHours = 30;
        int extraHours = hoursWorked - regularHours;
        weeklyWages = (hourlyWage * regularHours) + (hourlyWage * 2 * extraHours);
    }
    printf("Weekly wages: $%.2f\n", weeklyWages);

    return 0;
}
```

Q.3 Mr. X goes to market for buying some fruits and vegetables. He is having a currency of Rs 500 with him for marketing. From a shop, he purchases 2.0 kg Apple priced Rs. 50.0 per kg, 1.5 kg Mango priced Rs.35.0 per kg, 2.5 kg Potato priced Rs.10.0 per kg, and 1.0 kg Tomato priced Rs.15 per kg. He gives the currency of Rs.

500 to the shopkeeper. Find out the amount shopkeeper will return to X by writing a C program.

```
#include <stdio.h>

int main() {
    float priceApplePerKg = 50.0;
    float priceMangoPerKg = 35.0;
    float pricePotatoPerKg = 10.0;
    float priceTomatoPerKg = 15.0;

    float quantityAppleKg = 2.0;
    float quantityMangoKg = 1.5;
    float quantityPotatoKg = 2.5;
    float quantityTomatoKg = 1.0;

    float costApple = priceApplePerKg * quantityAppleKg;
    float costMango = priceMangoPerKg * quantityMangoKg;
    float costPotato = pricePotatoPerKg * quantityPotatoKg;
    float costTomato = priceTomatoPerKg * quantityTomatoKg;

    float totalCost = costApple + costMango + costPotato + costTomato;

    float currency = 500.0;

    float amountToReturn = currency - totalCost;

    if (amountToReturn >= 0) {
        printf("Amount to be returned to Mr. X: Rs. %.2f\n", amountToReturn);
    } else {
        printf("Mr. X does not have enough money to make the purchase.\n");
    }
}
```

```
    return 0;
}
```

Q4. Write a C program to print your name, date of birth and mobile number in 3 different lines.

Q5. Write a program to read an integer, a character and a float value from keyboard and display the same in different lines on the screen.

```
#include <stdio.h>
```

```
int main() {
```

```
    int integerNumber;
```

```
    char character;
```

```
    float floatNumber;
```

```
    printf("Enter an integer: ");
```

```
    scanf("%d", &integerNumber);
```

```
    printf("Enter a character: ");
```

```
    scanf(" %c", &character);
```

```
    printf("Enter a float: ");
```

```
    scanf("%f", &floatNumber);
```

```
    printf("Integer: %d\n", integerNumber);
```

```
    printf("Character: %c\n", character);
```

```
    printf("Float: %.2f\n", floatNumber);
```

```
    return 0;
```

```
}
```

Q6. Write a program to print the following line (Assume the total value is contained in a variable named cost)

The sales total is : \$ 172.53

```
#include <stdio.h>

int main() {
    double cost = 172.53
    printf("The sales total is : $ %.2f\n", cost);
    return 0;
}
```

Q7. Raju got 6 and half apples from each of Raghu, Sheenu and Akash. He wants to know how many apples he has in total without adding them. Write a program which could help Raju in doing this.

```
#include <stdio.h>

int main() {
    int applesPerPerson = 6;
    float halfApple = 0.5;
    float totalApples = 3 * (applesPerPerson + halfApple);
    printf("Raju has %.1f apples in total.\n", totalApples);
    return 0;
}
```

Q8. Write a program that prints the floating point value in exponential format correct to two decimal places.

```
#include <stdio.h>
```

```
int main() {
```

```
double floatValue = 12345.6789;
printf("Value in exponential format: %.2e\n", floatValue);

return 0;
}
```

Q9. Write a program to input and print your mobile number (i.e. of 10 digits).

```
#include <stdio.h>

int main() {
    long long int mobileNumber;

    printf("Enter your 10-digit mobile number: ");
    scanf("%lld", &mobileNumber);

    if (mobileNumber >= 1000000000LL && mobileNumber <= 9999999999LL) {
        printf("Your mobile number is: %lld\n", mobileNumber);
    } else {
        printf("Invalid input. Please enter a 10-digit mobile number.\n");
    }

    return 0;
}
```

Q10. The population of a city is 30000. It increases by 20 % during first year and 30% during the second year. Write a program to find the population after two years? (Ans: 46800)

```
#include <stdio.h>
```

```

int main() {
    int initialPopulation = 30000;
    float growthRateYear1 = 0.20;
    float growthRateYear2 = 0.30;

    int populationYear1 = initialPopulation + (initialPopulation * growthRateYear1);

    int populationYear2 = populationYear1 + (populationYear1 * growthRateYear2);

    printf("Population after two years: %d\n", populationYear2);

    return 0;
}

```

Q11. Write a program to find the ASCII value of a character.

```

#include <stdio.h>

int main() {
    char ch;
    printf("Enter a character: ");
    scanf("%c", &ch);
    printf("The ASCII value of '%c' is %d\n", ch, (int)ch);
    return 0;
}

```

Q12. Write a program to calculate salary of an employee, given his basic pay (entered by user), HRA=15% of the basic pay and TA=20% of the basic pay.

```

#include <stdio.h>

int main() {
    float basicPay, HRA, TA, salary;

    printf("Enter the basic pay: $");
}

```

```

scanf("%f", &basicPay);

HRA = 0.15 * basicPay;
TA = 0.20 * basicPay;
salary = basicPay + HRA + TA;

printf("Salary: $%.2f\n", salary);

return 0;
}

```

Q13. Write a program to find the slope of a line and angle of inclination that passes through two points P and Q with coordinates (xp, yp) and (xq, yq) respectively.

```

#include <stdio.h>
#include <math.h>

int main() {
    double xp, yp, xq, yq, slope, angle;
    printf("Enter the coordinates of point P (xp, yp): ");
    scanf("%lf %lf", &xp, &yp);

    printf("Enter the coordinates of point Q (xq, yq): ");
    scanf("%lf %lf", &xq, &yq);

    if (xq - xp != 0) {
        slope = (yq - yp) / (xq - xp);

        angle = atan(slope) * 180 / M_PI;

        printf("Slope of the line: %.2lf\n", slope);
    }
}

```



```

printf("Angle of inclination: %.2lf degrees\n", angle);
    } else {
printf("The line is vertical. Slope is undefined.\n");
    }

    return 0;
}

```

Q14. The SPI (Semester Performance Index) is a weighted average of the grade points earned by a student in all the courses he registered for in a semester. If the grade points associated with the letter grades awarded to a student are $g_1, g_2, g_3, \dots, g_k$ etc. and the corresponding credits are $c_1, c_2, c_3, \dots, c_k$, the SPI is given by:

```

#include <stdio.h>

int main() {
    int numCourses;

    printf("Enter the number of courses: ");
    scanf("%d", &numCourses);

    float totalGradePoints = 0.0;
    float totalCredits = 0.0;

    for (int i = 1; i <= numCourses; i++) {
        float gradePoint, credits;

        printf("Enter grade point for course %d: ", i);
        scanf("%f", &gradePoint);

        printf("Enter credit hours for course %d: ", i);
        scanf("%f", &credits);

        totalGradePoints += gradePoint * credits;
        totalCredits += credits;
    }

    if (totalCredits > 0) {
        float SPI = totalGradePoints / totalCredits;

        printf("SPI: %.2f\n", SPI);
    } else {

```

```
printf("Total credits cannot be zero.\n");
}
return 0;
}
```

Q 15. Write a program to calculate the frequency (f) of a given wave with wavelength (λ) and speed (c), where $c=\lambda*f$.

```
#include <stdio.h>

int main() {
    double speedOfWave, wavelength, frequency;
    printf("Enter the speed of the wave (c) in m/s: ");
    scanf("%lf", &speedOfWave);
    printf("Enter the wavelength ( $\lambda$ ) in meters (m): ");
    scanf("%lf", &wavelength);
    frequency = speedOfWave / wavelength;
    printf("The frequency of the wave is %.2lf Hz\n", frequency);
    return 0;
}
```

Q 16. A car travelling at 30 m/s accelerates steadily at 5 m/s² for a distance of 70 m. What is the final velocity of the car? [Hint: $v^2 = u^2 + 2as$]

```
#include <stdio.h>
#include <math.h>

int main() {
    double u = 30.0;
    double a = 5.0;
    double s = 70.0;
    double v;
```

```

v = sqrt(u * u + 2 * a * s);
printf("The final velocity of the car is %.2f m/s\n", v);
return 0;
}

```

Q 17. A horse accelerates steadily from rest at 4 m/s² for 3s. (a) What is its final velocity? (b) How far has it travelled? [Hint: (a) $v = u + at$ (b) $s = ut + \frac{1}{2}at^2$]

```

#include <stdio.h>

int main() {
    double u = 0.0;
    double a = 4.0;
    double t = 3.0;
    double v, s;
    v = u + a * t;
    s = u * t + 0.5 * a * t * t;
    printf("The final velocity of the horse is %.2f m/s\n", v);
    printf("The horse has traveled a distance of %.2f meters\n", s);
    return 0;
}

```

Q 18. Write a program to find the sum of your four last digit of your university roll number .

```

#include <stdio.h>

int main() {
    char rollNumber[] = "12345678";
    int sum = 0;
    if (strlen(rollNumber) >= 4) {
        int length = strlen(rollNumber);
        for (int i = length - 1; i >= length - 4; i--) {
            if (isdigit(rollNumber[i])) {
                sum += rollNumber[i] - '0';
            }
        }
    }
}

```

```

        }
    }
    printf("Sum of the last four digits: %d\n", sum);
    } else {
    printf("Invalid roll number. It should have at least four digits.\n");
    }

    return 0;
}

```

Q19. Write a program to initialize your height and weight in cm. and kgs respectively demonstrating compile time initialization and convert them in feet

```

#include <stdio.h>

int main() {
    double heightCm = 175.0;
    double weightKg = 70.0;

    double heightFeet = heightCm * CM_TO_FEET;
    double weightPounds = weightKg * KG_TO_POUNDS;
    printf("Height in feet: %.2f ft\n", heightFeet);
    printf("Weight in pounds: %.2f lbs\n", weightPounds);

    return 0;
}

```

Q 20 .Code the variable declarations for each of following:

- a) **A character variable named option.**
- b) **An integer variable sum initialized to 0**
- c) **A floating point variable, product, initialized to 1**

```
#include <stdio.h>
```

```
int main() {  
    char option;  
    int sum = 0;  
  
    float product = 1.0;  
  
    return 0;  
}
```

Q21. Write a program that reads nine integers. Display these numbers by printing three numbers in a line separated by commas.

```
#include <stdio.h>
```

```
int main() {  
    int numbers[9];  
  
    printf("Enter nine integers:\n");  
    for (int i = 0; i < 9; i++) {  
        scanf("%d", &numbers[i]);  
    }
```

```
    printf("Numbers:\n");  
    for (int i = 0; i < 9; i++) {  
        printf("%d", numbers[i]);  
        if ((i + 1) % 3 == 0) {  
            printf("\n");  
        } else {  
            printf(", ");  
        }  
    }
```

```
}  
  
return 0;  
}
```

Q22. What are header files and what are its uses in C programming?

Header files in C programming are files that contain declarations of functions, variables, and other constructs along with necessary preprocessor directives. These files are typically used to provide the necessary information to the compiler about the functions and data types used in a program without having to provide their complete definitions. Header files have the file extension ".h" and are included in C programs using the #include preprocessor directive.

Answer: The primary uses of header files in C programming are:

Declaration of Functions and Data Types: Header files declare the functions, data types, and macros that can be used in a C program. This allows you to use these functions and data types without needing to define them in every source file that uses them. For example, the <stdio.h> header file provides declarations for functions like printf and data types like int and char.

Code Organization: Header files help in organizing the code into manageable and modular components. By separating function declarations and data type definitions into header files, you can have a clean and structured codebase.

Encapsulation: Header files allow you to encapsulate the interface of a module or library, hiding the implementation details from the users of the module. This concept of information hiding is essential for building robust and maintainable software.

Code Reusability: Header files promote code reusability. If you have a set of functions or data types that are commonly used in multiple programs, you can create a header file and include it wherever needed, avoiding redundant code.

Interoperability: Header files are used for interfacing with external libraries and system functions. For example, when you use libraries like the Standard C Library (<stdio.h>, <stdlib.h>, etc.), you include their respective header files to access their functionality.

Commonly used standard header files in C programming include <stdio.h>, <stdlib.h>, <string.h>, <math.h>, and many others. Additionally, you can create your own header files to organize your code and define custom functions and data types.

Q23. What will be the output of following program?

```
#include<stdio.h>
int main()
{ int num=070;
printf(“%d\t%o\t%x”,num,num,num);
}
```

OUTPUT = 56 70 38

Q 24. What will be the output of following program?

```
#include <stdio.h>
void main()
{
int x = printf("GLA UNIVERSITY");
printf("%d", x);
}
```

OUTPUT = GLA UNIVERSITY15

Q25. What are library functions? List any four library functions.

Library functions in programming are predefined functions provided by a programming language's standard library or external libraries. These functions serve specific purposes and can be used by developers to perform common tasks without the need to write the code for those tasks from scratch. Library functions save time and effort and promote code reusability.

Answer:Here are four commonly used library functions in C programming:

printf: The printf function is used for formatted output. It allows developers to print data to the standard output (usually the console) in a specified format. This function is part of the C Standard Library and provides a wide range of formatting options.

scanf: The scanf function is used for formatted input. It enables developers to read data from the standard input (typically the keyboard) and store it in variables. Like printf, scanf offers various format specifiers for different data types.

strlen: The strlen function is used to find the length of a string. It calculates and returns the number of characters in a given null-terminated string. This function is especially useful for working with strings and character arrays.

rand: The rand function is used to generate pseudo-random numbers. It produces a sequence of random integer values within a specified range. While the numbers generated are not truly random, they can serve various purposes, such as simulations or games.

These library functions are just a small subset of the many functions available in the C Standard Library, which offers a comprehensive set of tools for performing a wide range of tasks in C programming. Developers can include the appropriate header files, such as `<stdio.h>` or `<stdlib.h>`, to access these functions and leverage their functionality in their C programs.

Q26. What will be the output of following program?

```
#include <stdio.h>
void main()
{
    int x = printf("C is placement oriented Language") – printf("Hi");
    printf(" %d %o %x", x,x,x);
}
```

OUTPUT= 32 40 20

Q27. What is the meaning of following statement?

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

REASON;

The scanf function is used for formatted input. In this case, it attempts to read two integer values from the standard input (usually the keyboard) using the format specifier %d%d and stores them in the variables a and b. The return value of scanf is the number of successfully matched and assigned input items. If it successfully reads two integers, it will return 2.

The printf function is used for formatted output. It takes the return value of the scanf function, which is 2, and prints it using the format specifier %d. Therefore, it prints the value 2 as an integer.

In summary, the statement printf("%d", scanf("%d%d", &a, &b)); reads two integers into variables a and b and then prints the value 2 to the standard output as an integer.

Q28. What will be the output of following program?

```
#include <stdio.h>
void main()
{
    printf(" \"C %% FOR %% PLACEMENT\"");
}
```

OUTPUT = "C % FOR % PLACEMENT"

Q29. Suppose distance between GLA University and Delhi is m km (to be entered by user), by BUS you can reach Delhi in 4 hours. Develop a 'C' program to calculate speed of bus.

```
#include <stdio.h>
```

```
int main() {
```

```
    double distance, time, speed;
```

```
    printf("Enter the distance between GLA University and Delhi (in km): ");
```

```
    scanf("%lf", &distance);
```

```
    time = 4.0;
```

```
    speed = distance / time;
```

```
    printf("The speed of the bus is %.2lf km/hr\n", speed);
```

```
    return 0;
```

```
}
```

Q30. In an exam Satyam got 50 marks, Suman got 70 marks and Shyam got 80 marks, Write a 'C' program to find average marks of these three participants.

```
#include <stdio.h>
```

```
int main() {
```

```
    int satyamMarks = 50;
```

```
int sumanMarks = 70;
int shyamMarks = 80;
double average;
average = (satyamMarks + sumanMarks + shyamMarks) / 3.0;
printf("Average marks of Satyam, Suman, and Shyam: %.2lf\n", average);

return 0;
}
```

Q31. One day, Mohan called Saurav and Sajal and gave some money to them, later he realized that money that was given to Saurav should be given to Sajal and vice-versa. Develop a ‘C’ program to help Mohan so that he can rectify his mistake.

```
#include <stdio.h>

int main() {
    double sauravMoney, sajalMoney, temp;
    printf("Enter the amount of money given to Saurav: ");
    scanf("%lf", &sauravMoney);

    printf("Enter the amount of money given to Sajal: ");
    scanf("%lf", &sajalMoney);

    temp = sauravMoney;
    sauravMoney = sajalMoney;
    sajalMoney = temp;

    printf("After exchanging the money:\n");
    printf("Money given to Saurav: %.2lf\n", sauravMoney);
    printf("Money given to Sajal: %.2lf\n", sajalMoney);

    return 0;
}
```

```
}
```

Q32. One day when I was going for a lunch, suddenly rain started, I was very hungry so started running with speed of 4km/h and it took 3 min to reach mess. Help me to develop a 'C' program to calculate distance travelled by me.

```
#include <stdio.h>

int main() {
    double speed_kph = 4.0; // Speed in kilometers per hour
    double time_min = 3.0; // Time in minutes

    double time_hr = time_min / 60.0;
    double distance_km = speed_kph * time_hr;

    printf("Distance traveled: %.2lf kilometers\n", distance_km);

    return 0;
}
```

Q33. Can two or more escape sequences such as \n and \t be combined in a single line of program code?

Answer: Yes, you can combine multiple escape sequences in a single line of program code in many programming languages that support escape sequences. Escape sequences are special character sequences that are used to represent characters that are difficult or impossible to include directly in a string or character literal. Common escape sequences include \n for a newline and \t for a tab.

Q34. What are comments and how do you insert it in a C program?

Comments in a C program are text notations that are used to provide explanations, documentation, or annotations within the source code. Comments are ignored by the C compiler and do not have any effect on the program's execution. They are solely intended for human readers to better understand the code.

```
// This is a single-line comment
```

```
int main() {  
    // This is a comment within a function  
    return 0;  
}
```

Q35. What is wrong in this statement? `scanf("%d",number);`

The issue in the statement `scanf("%d", number);` is that you need to provide the address of the variable where you want to store the input integer value, but you are passing the variable itself instead of its address.

Q36. What will be the output?

```
#include <stdio.h>  
int main()  
{  
    if (sizeof(int) > -1)  
        printf("Yes");  
    else  
        printf("No");  
    return 0;  
}
```

OUTPUT = The output of this program will be "Yes."

Q37. Point out which of the following variable names are invalid:

gross-salary INTEREST , salary of emp , avg. , thereisbookinmysoup

1. **gross-salary:** Valid - Variable names can include hyphens or underscores, so "gross-salary" is a valid variable name.

2. **INTEREST: Valid** - Variable names are case-sensitive in many programming languages, so "INTEREST" is a valid variable name.
3. **salary of emp:** Invalid - Variable names cannot have spaces in them. You can use underscores or camelCase to separate words, like "salary_of_emp" or "salaryOfEmp."
4. **avg.: Invalid** - Variable names cannot contain periods (dots) in most programming languages. You can use underscores or letters and numbers to create valid variable names, like "avg" or "average."

Q38. Tom works at an aquarium shop on Saturdays. One Saturday, when Tom gets to work, he is asked to clean a 175-gallon reef tank. His first job is to drain the tank. He puts a hose into the tank and starts a siphon. Tom wonders if the tank will finish draining before he leaves work. He measures the amount of water that is draining out and finds that 12.5 gallons drain out in 30 minutes. So, he figures that the rate is 25 gallons per hour. Develop a 'C' program to help Tom to calculate time required to completely clean tank.

```
#include <stdio.h>
```

```
int main() {
```

```
    double flowRate = 25.0; // gallons per hour
```

```
    double tankSize = 175.0; // gallons
```

```
    double timeRequired = tankSize / flowRate; // hours
```

```
    printf("Time required to completely clean the tank: %.2f hours\n", timeRequired);
```

```
    return 0;
```

```
}
```

Q39. The percent y (in decimal form) of battery power remaining x hours after you turn on a laptop computer is $y = -0.2x + 1$. Develop a 'C' program to calculate after how many hours the battery power is at 75%?

```
#include <stdio.h>

int main() {
    // Define the desired battery power (75%)
    double desiredPower = 0.75;

    // Calculate the number of hours required using the given equation
    //  $y = -0.2x + 1$ , where  $y$  is the desiredPower (0.75)
    // Solve for  $x$ :  $0.75 = -0.2x + 1$ 
    //  $-0.2x = 0.75 - 1$ 
    //  $-0.2x = -0.25$ 
    //  $x = -0.25 / -0.2$ 
    double hoursRequired = (-0.25) / (-0.2);

    // Display the result
    printf("The battery power will be at 75%% after %.2f hours\n", hoursRequired);

    return 0;
}
```

language in a single go?

a. Compiler b. Interpreter

c. Linker Q40. Which of the following is used to convert the high level language in machine d. Assembler

The correct answer is:

a. Compiler

Q 41. What is the format specifier for an Octal Number?

- a. %0 b. %d
- c. %o d. %e

c. %o

Q 42. Which format specifier is used to print the exponent value upto 2 decimal places.

- a. %e b. %.2f c. %f d. %.2e

ANSWER d. %.2e

Q 43. Which of the following is not a basic data type?

- a. char
- b. array
- c. float
- d. int

ANSWER B. ARRAY

Q 44. What is the output of following code?

```
#include<stdio.h>
void main()
{
    int x=0;
    x= printf("\"hello\b\"");
    printf("%d",x);
}
```

- a. hello7 b. "hello"7 c. "hell"8 d. hell8

ANSWER = c. "hell"8

Q 45. What is the output of following code?

```
#include<stdio.h>
void main()
{
    int b,c=5 ;
    int("%d , %d", b,c);
}
```


}
a. 5, 5 b. 5, 5.000000
c. Garbage, 5.000000 d. Garbage, 5

ANSWER = b. 5, 5.000000

Q46. Which of the following is an identifier?

a. &fact b. Basic_pay c. enum d. lsum

ANSWER = b. Basic_pay

Q 47. What is the output of the following program?

```
#include<stdio.h>
void main()
{
    char x, a='c';
    x=printf("%c",a);
    printf("%d",x);
}
```

a. c1 b. cgarbage
c. 1 c. c

ANSWER = a. c1

Q48. Perform the following conversion from Decimal to other number as directed-

- a) $(365.55)_{10} = (?)_2$
- b) $(453.65)_{10} = (?)_8$
- c) $(5164.12)_{10} = (?)_{16}$
- d) $(23.65)_{10} = (?)_5$
- e) $(772)_{10} = (?)_7$

SLOUTIONS:

convert the integer part (365) to binary:
 $365 \text{ (decimal)} = 101101101 \text{ (binary)}$

Convert the fractional part (0.55) to binary:
 $0.55 \text{ (decimal)} = 0.10001100110011... \text{ (binary, recurring)}$

So, $(365.55)_{10}$ in binary is approximately $(101101101.10001100110011...)_{2}$.

b) $(453.65)_{10} = (?)_8$ (Octal):

Convert the integer part (453) to octal:

453 (decimal) = 705 (octal)

Convert the fractional part (0.65) to octal:

0.65 (decimal) = 0.52 (octal)

So, $(453.65)_{10}$ in octal is approximately $(705.52)_8$.

c) $(5164.12)_{10} = (?)_{16}$ (Hexadecimal):

Convert the integer part (5164) to hexadecimal:

5164 (decimal) = 1434 (hexadecimal)

Convert the fractional part (0.12) to hexadecimal:

0.12 (decimal) = $0.1A3D70A3D70A...$ (hexadecimal, recurring)

So, $(5164.12)_{10}$ in hexadecimal is approximately $(1434.1A3D70A3D70A...)_{16}$.

d) $(23.65)_{10} = (?)_5$ (Quinary):

Convert the integer part (23) to quinary:

23 (decimal) = 43 (quinary)

Convert the fractional part (0.65) to quinary:

0.65 (decimal) = $0.13030303...$ (quinary, recurring)

So, $(23.65)_{10}$ in quinary is approximately $(43.13030303...)_{5}$.

e) $(772)_{10} = (?)_7$ (Septenary):

Convert the integer part (772) to septenary:

772 (decimal) = 1442 (septenary)

Since there is no fractional part, the conversion is straightforward.

So, $(772)_{10}$ in septenary is equal to $(1442)_7$.

Q49. Covert the following numbers to decimal number system-

a) $(325.54)_6 = (?)_{10}$

b) $(1001010110101.1110101)_2 = (?)_{10}$

c) $(742.72)_8 = (?)_{10}$

d) $(AC94.C5)_{16} = (?)_{10}$

a) $(325.54)_6 = (?)_{10}$ (Hexadecimal to Decimal):

Integer part:

$$3 * 6^2 + 2 * 6^1 + 5 * 6^0 = 108 + 12 + 5 = 125$$

Fractional part:

$$5 * 6^{-1} + 4 * 6^{-2} = 5/6 + 4/36 = 5/6 + 1/9 = 15/18 + 2/18 = 17/18$$

So, $(325.54)_6$ in decimal is $(125.9444...)_{10}$, or approximately 125.9444... in decimal.

b) $(1001010110101.1110101)_2 = (?)_{10}$ (Binary to Decimal):

Integer part:

$$1 * 2^{12} + 0 * 2^{11} + 0 * 2^{10} + \dots + 1 * 2^0 = 8192 + 0 + 0 + \dots + 1 = 8193$$

Fractional part:

$$1 * 2^{-1} + 1 * 2^{-2} + 1 * 2^{-3} + \dots + 1 * 2^{-7} = 0.5 + 0.25 + 0.125 + \dots + 0.0078125 = 0.9999389648\dots$$

So, $(1001010110101.1110101)_2$ in decimal is approximately 8193.9999389648...

c) $(742.72)_8 = (?)_{10}$ (Octal to Decimal):

Integer part:

$$7 * 8^2 + 4 * 8^1 + 2 * 8^0 = 448 + 32 + 2 = 482$$

Fractional part:

$$7 * 8^{-1} + 2 * 8^{-2} = 7/8 + 2/64 = 7/8 + 1/32 = 35/32 + 1/32 = 36/32 = 9/8 = 1.125$$

So, $(742.72)_8$ in decimal is 482.125.

d) $(AC94.C5)_{16} = (?)_{10}$ (Hexadecimal to Decimal):

Integer part:

$$A * 16^3 + C * 16^2 + 9 * 16^1 + 4 * 16^0 = 40960 + 3072 + 144 + 4 = 44280$$

Fractional part:

$$C * 16^{-1} + 5 * 16^{-2} = 12/16 + 5/256 = 12/16 + 5/256 = 48/64 + 5/256 = 53/64 = 0.828125$$

So, $(AC94.C5)_{16}$ in decimal is 44280.828125.

Q50. Perform the following conversion from Hexadecimal to other number as directed-

$$(DB56.CD4)_{16} = (?)_2, (?)_8, (?)_4$$

Given hexadecimal number: $(DB56.CD4)_{16}$

SOLUTIONS:

1. Convert to Binary: To convert a hexadecimal number to binary, you can convert each hexadecimal digit to its 4-bit binary equivalent.

$$D = 1101 \quad B = 1011 \quad 5 = 0101 \quad 6 = 0110 \quad C = 1100 \quad D = 1101 \quad 4 = 0100$$

Now, put the binary digits together along with the decimal point:

$$(DB56.CD4)_{16} = 110110110110010101101100110100$$

2. Convert to Octal: To convert binary to octal, group the binary digits into sets of three, starting from the binary point. Add leading zeros if necessary.

$$11 \ 011 \ 011 \ 011 \ 001 \ 010 \ 110 \ 110 \ 011 \ 010 \ 0$$

Now, convert each group to its octal equivalent:

$$(11 \ 011 \ 011 \ 011 \ 001 \ 010 \ 110 \ 110 \ 011 \ 010 \ 0)_2 = (3353136320)_8$$

3. Convert to Decimal: To convert binary to decimal, multiply each digit by the corresponding power of 2 and sum the results.

$$\begin{aligned}(\text{DB56.CD4})_{16} &= (110110110110010101101100110100)_2 \\&= 1 * 2^{35} + 1 * 2^{34} + 0 * 2^{33} + 1 * 2^{32} + 1 * 2^{31} + 0 * 2^{30} + 1 * 2^{29} + 1 * 2^{28} + \\&0 * 2^{27} + 1 * 2^{26} + 1 * 2^{25} + 0 * 2^{24} + 1 * 2^{23} + 0 * 2^{22} + 0 * 2^{21} + 1 * 2^{20} + 1 \\&* 2^{19} + 0 * 2^{18} + 1 * 2^{17} + 1 * 2^{16} + 0 * 2^{15} + 1 * 2^{14} + 1 * 2^{13} + 0 * 2^{12} + 0 * \\&2^{11} + 1 * 2^{10} + 1 * 2^9 + 0 * 2^8 + 0 * 2^7 + 1 * 2^6 + 0 * 2^5 + 1 * 2^4 + 1 * 2^3 + 0 * \\&2^2 + 0 * 2^1 + 1 * 2^0\end{aligned}$$

Now, calculate the decimal value:

$$(\text{DB56.CD4})_{16} = 1,939,610,276$$

So, the conversions are as follows:

- Binary: $(\text{DB56.CD4})_{16} = (110110110110010101101100110100)_2$
- Octal: $(\text{DB56.CD4})_{16} = (3353136320)_8$
- Decimal: $(\text{DB56.CD4})_{16} = 1,939,610,276$

Q51. Perform the following conversion from octal to other number as directed-

$$(473.42)_8 = (?)_2, (?)_{10}, (?)_{16}, (?)_5$$

- Binary: $(473.42)_8 = (1001110010110010.010)_2$
- Decimal: $(473.42)_8 = 4002.25_{10}$
- Hexadecimal: $(473.42)_8 = (9CB2.4)_{16}$
- Base-5: $(473.42)_8 = 13002.4_5$

Q52. Find the value of A?

a) $(23)_{10} = (17)_A$

b) $(21)_{16} = (41)_A$

c) $(32)_8 = (101)_A$

ANSWERS = A) $A = 4$

B) $A = 0$

C) $A_1 = (3 + \sqrt{5}) / 2$

$$A_2 = (3 - \sqrt{5}) / 2$$

Q53: What will be the output of following program? Assume integer is of 2 bytes

```
void main(){  
int a=32770;  
printf("%d",a);  
}
```

Q54: #include <stdio.h>

```
int main()  
{  
float c = 5.0;  
printf ("Temperature in Fahrenheit is %.2f", (9/5)*c + 32);  
return 0;  
}
```

#include <stdio.h>

```
int main() {  
float c = 5.0;  
printf("Temperature in Fahrenheit is %.2f", (9.0 / 5.0) * c + 32);  
return 0;  
}
```

OUTPUT = Temperature in Fahrenheit is 41.00

