INTRODUCTION:-

NAME:- UPLAKSH RAJORIA

SECTION:- AY

ROLL NO. :-63

//QUESTION:-1

```
#include <stdio.h>
#include<math.h>
int main()
{
   float originalPrice, taxRate, totalPrice;

   printf("Enter the original price of the product: rs_");
   scanf("%f", &originalPrice);
   printf("Enter the sales tax rate : ");
   scanf("%f", &taxRate);

   totalPrice = originalPrice + (originalPrice * (taxRate / 100.0));

   printf("The total price after adding %.2f%% sales tax: rs_%.2f\n", taxRate, totalPrice);


        return 0;
}
```

//QUESTION:-2

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

```
#include<math.h>

int main()
{

   float wagesPerHour, hoursWorked, weeklyWages;

   printf("Enter wages per hour: rs_");
   scanf("%f", &wagesPerHour);

   printf("Enter hours worked in the week: ");
   scanf("%f", &hoursWorked);

   if (hoursWorked <= 30) {
      weeklyWages = wagesPerHour * hoursWorked;
   } else {
      weeklyWages = wagesPerHour * 30;

      weeklyWages += (wagesPerHour * 2) * (hoursWorked - 30);
   }

   printf("Weekly wages: $%.2f\n", weeklyWages);


         return 0;
}
```

```
//QUESTION:-3
#include <stdio.h>
#include<math.h>

int main()
{

   float priceApple = 50.0;
   float priceMango = 35.0;
   float pricePotato = 10.0;
   float priceTomato = 15.0;

   float quantityApple = 2.0;
   float quantityMango = 1.5;
   float quantityPotato = 2.5;
   float quantityTomato = 1.0;

   float totalCost = (priceApple * quantityApple) + (priceMango * quantityMango) + (pricePotato *
quantityPotato) + (priceTomato * quantityTomato);

   float amountPaid = 500.0;

   float change = amountPaid - totalCost;

   if (change >= 0) {
      printf("Mr. X, your total cost is Rs. %.2f\n", totalCost);
      printf("You paid Rs. %.2f\n", amountPaid);
      printf("Your change is Rs. %.2f\n", change);
   } else {
```

```
        printf("Mr. X, you have not paid enough. Please provide more money.\n");
    }



            return 0;
}
```

//QUESTION:-4

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

int main()
{

    char name;
            char DOB;
            char MOBILE_NO;

    // Print each piece of information on separate lines
    printf("Name: ");
    scanf("%s\n",&name);

    printf("Date of Birth: ");
    scanf("%s\n",&DOB);

    printf("Mobile Number: ");
    scanf("%s\n",&MOBILE_NO);
```

```
    return 0;
}
```

```
//QUESTION:-5
#include <stdio.h>
#include<math.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("You entered:\n");
   printf("Integer: %d\n", integerNumber);
```

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


        return 0;
}
```

//QUESTION:-6

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

int main()
{

    float cost = 172.53;

    printf("The sales total is : $ %.2f\n", cost);


        return 0;
}
```

//QUESTION:-7

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

```
int main()
{

   float applesFromEach = 6.5;

   float totalApples = applesFromEach * 3;

   printf("Raju has %.1f apples in total without adding them.\n", totalApples);



        return 0;
}
```

//QUESTION:-8

```
#include <stdio.h>

int main()
 {
   float Value;

   printf("Enter a floating-point value: ");

   scanf("%f", &Value);

   printf("The value in exponential format: %.2ef\n",Value);

   return 0;
}
```

```
//QUESTION:-9
#include <stdio.h>
#include<math.h>

int main()
{

   char mobileNumber[10];

   printf("Enter your mobile number: ");
   scanf("%s", mobileNumber);

   printf("Your mobile number is: %s\n", mobileNumber);



        return 0;
}
```

```
//QUESTION:-10
#include <stdio.h>
#include<math.h>

int main()
{

   int initialPopulation = 30000;
   float growthRate1 = 0.20;
```

```
    float growthRate2 = 0.30;

    int populationAfterFirstYear = initialPopulation + (int)(initialPopulation * growthRate1);

    int populationAfterSecondYear = populationAfterFirstYear + (int)(populationAfterFirstYear *
growthRate2);

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



        return 0;
}
```

//QUESTION:-11

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

int main()
{

    char character;

    printf("Enter a character: ");
    scanf("%c", &character);

    printf("The ASCII value of '%c' is %d\n", character, character);
```

//QUESTION:-12

```c
#include <stdio.h>
#include<math.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("Basic Pay: %.2f\n", basicPay);
   printf("HRA: %.2f\n", hra);
   printf("TA: %.2f\n", ta);
   printf("Total Salary: %.2f\n", salary);



        return 0;
}
//QUESTION :13
#include <stdio.h>
#include <math.h>

int main()
```

```
{
   float xp, yp, xq, yq;

   printf("Enter the coordinates of point P (xp yp): ");
   scanf("%f%f",&xp,&yp);

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

   float slope = (yq - yp) / (xq - xp);

   float angle_rad = atan(slope);
   float angle_deg = angle_rad * 180.0 / M_PI;

   printf("The slope of the line passing through P and Q is: %.2f\n", slope);
   printf("The angle of inclination (in degrees) is: %.2f\n", angle_deg);

   return 0;
}
```

//QUESTION:-14

```
#include <stdio.h>

int main()
{

   float g1, g2, g3, g4, g5;
   float c1, c2, c3, c4, c5;
```

```
    printf("Enter grade points for each course (g1 g2 g3 g4 g5): ");
    scanf("%f%f%f%f%f",&g1,&g2,&g3,&g4,&g5);

    printf("Enter credits for each course (c1 c2 c3 c4 c5): ");
    scanf("%f%f%f%f%f",&c1,&c2,&c3,&c4,&c5);

    float spi = (g1 * c1 + g2 * c2 + g3 * c3 + g4 * c4 + g5 * c5) / (c1 + c2 + c3 + c4 + c5);

    printf("The Semester Performance Index (SPI) for k=5 is: %.2f\n", spi);

    return 0;
}
```

//QUESTION:-15

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

int main()
{

    float wavelength, speed, frequency;

    printf("Enter the wavelength (in meters): ");
    scanf("%f", &wavelength);

    printf("Enter the speed of the wave (in meters per second): ");
    scanf("%f", &speed);

    frequency = speed / wavelength;
```

```
    printf("The frequency of the wave is %.2f Hz\n", frequency);



        return 0;
}
```

//QUESTION:-16

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

int main()
{

    initialVelocity = 30.0;
    acceleration = 5.0;
    distance = 70.0;
    finalVelocity;

    finalVelocity = sqrt(initialVelocity * initialVelocity + 2 * acceleration * distance);

    printf("The final velocity of the car is %.2f m/s\n", finalVelocity);



        return 0;
}
```

//QUESTION:-17

```c
#include <stdio.h>

int main()
{

   float initialVelocity = 0.0;
   float acceleration = 4.0;
   float time = 3.0;

   float finalVelocity = initialVelocity + (acceleration * time);

   float distanceTraveled = (initialVelocity * time) + (0.5 * acceleration * time * time);

   printf("The final velocity is %.2f m/s\n", finalVelocity);
   printf("The distance traveled is %.2f meters\n", distanceTraveled);

   return 0;
}
```

```c
   //QUESTION:-18
#include <stdio.h>
#include<math.h>

int main()
{

   int rollNumber;
   int lastFourDigits;
```

```
    int sum = 0;

    printf("Enter your university roll number: ");
    scanf("%d", &rollNumber);

    lastFourDigits = rollNumber % 10000;



    while (lastFourDigits > 0)
            {
       sum += lastFourDigits % 10;
       lastFourDigits /= 10;
    }

    printf("The sum of the last four digits of your roll number is: %d\n", sum);



            return 0;
}
```

//QUESTION:-19

```
#include <stdio.h>

int main()
{

    float height_cm = 175.0;
    float weight_kg = 70.0;
```

```
    float CM_TO_FEET = 0.0328084;
    float KG_TO_POUNDS = 2.20462;

    float height_feet = height_cm * CM_TO_FEET;

    float weight_pounds = weight_kg * KG_TO_POUNDS;

    printf("Your height is %.2f cm, which is %.2lf feet.\n", height_cm, height_feet);
    printf("Your weight is %.2f kg, which is %.2lf pounds.\n", weight_kg, weight_pounds);

    return 0;
}
```

//QUESTION:-20

```
#include <stdio.h>

int main()
{

    char option;

    int sum = 0;

    float product = 1.0;

    return 0;
}
```

//QUESTION:-21

```
#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("The numbers in groups of three:\n");

   for (int i = 0; i < 9; i++) {
      printf("%d", numbers[i]);

      if ((i + 1) % 3 == 0) {
         printf("\n");
      } else {
         printf(", ");
      }
   }

   return 0;
}
```

//QUESTION:-22

Header files in C programming are files containing declarations of functions, variables, and other constructs used in C programs. These declarations provide information about these constructs to the compiler, allowing it to understand how to properly compile and link the code. Header files in C programming are essential for organizing and maintaining code, enabling code reuse,

and providing the necessary information to the compiler for correct compilation and linking of C programs.

//QUESTION:-23

56 70 38

//QUESTION:-24

GLA UNIVERSITY14

//QUESTION:-25

Library functions, also known as standard library functions or built-in functions, are predefined functions provided by programming language libraries or standard libraries. These functions perform common tasks and operations, and they are available for use in your programs without the need to implement them from scratch. In C programming, standard library functions are typically declared in header files and are part of the C Standard Library.

Here are four commonly used C Standard Library functions:

1. **printf**: Found in `<stdio.h>`, this function is used for formatted output. It allows you to print data to the standard output (usually the console) with various formatting options.
2. **scanf**: Also found in `<stdio.h>`, `scanf` is used for formatted input. It allows you to read and store data from the standard input (usually the keyboard) based on specified format specifiers.
3. **strlen**: Found in `<string.h>`, `strlen` is used to calculate the length of a null-terminated character string (a string with a null character `'\0'` at the end).
4. **rand**: Found in `<stdlib.h>`, `rand` is used to generate pseudo-random numbers. It allows you to generate random integers within a specified range.

//QUESTION:-26

C is placement oriented LanguageHi23 23

//QUESTION:-27

the statement `printf("%d", scanf("%d%d", &a, &b));` will first execute `scanf`, which reads integers from the user, and then it will print the result (the number of successfully scanned items) using `printf`. The

printed result will be either `0`, `1`, or `2`, depending on how many integers the user successfully entered and were stored in `a` and `b`.

//QUESTION:-28
"C % FOR % PLACEMENT"

//QUESTION:-29

```
#include <stdio.h>

int main()
{
   float distance, time;

   printf("Enter the distance between GLA University and Delhi (in km): ");
   scanf("%f", &distance);

   time = 4.0;

   float speed = distance / time;

   printf("The speed of the bus is %.2f km/h\n", speed);

   return 0;
}
```

//QUESTION:-30

```
#include <stdio.h>

int main()
{
   int satyamMarks = 50;
   int sumanMarks = 70;
   int shyamMarks = 80;

   int totalMarks = satyamMarks + sumanMarks + shyamMarks;

   float averageMarks = (float)totalMarks / 3;
```

```
   printf("Average marks of Satyam, Suman, and Shyam: %.2f\n", averageMarks);

   return 0;
}


//QUESTION:- 31

#include <stdio.h>

int main()
{
   int sauravMoney, sajalMoney, temp;

   printf("Enter the money given to Saurav: ");
   scanf("%d", &sauravMoney);

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

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

   printf("After the correction:\n");
   printf("Money given to Saurav: %d\n", sauravMoney);
   printf("Money given to Sajal: %d\n", sajalMoney);

   return 0;
}

//QUESTION:-32

#include <stdio.h>
int main()
 {
   float speed = 4.0;
   float timeInMinutes = 3.0;
   float timeInHours = timeInMinutes / 60.0;
```

```
    float distance;

    distance = speed * timeInHours;
    printf("Distance traveled: %.2f km\n", distance);

    return 0;
}
```

//QUESTION:-33

Yes, you can combine multiple escape sequences in a single line of program code in languages like C and C++. Escape sequences are special character combinations that are used to represent non-printable characters or to format the output.

For example,

```
#include <stdio.h>

int main() {
    printf("This is a line with a newline (\n) and a tab (\t) for formatting.\n");
    return 0;
}
```

//QUESTION:-34

Comments in a C program are annotations or explanatory notes that are ignored by the compiler. They are used to provide information, explanations, or documentation within the source code to make it more understandable to humans. Comments are not executed as part of the program and do not affect the program's functionality.

```
// This is a single-line comment
int x = 10; // This is a comment after code

/* This is a
   multi-line comment */
int y = 20;
```

//QUESTION:-35

The statement `scanf("%d", number);` has an issue with the way the `scanf` function is used. The issue is with the way the `number` variable is passed as an argument to `scanf`. In C, when you use `scanf` to read input and store it in a variable, you should pass the address of the variable using the `&` (address-of) operator. Here's the corrected statement:

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

//QUESTION:-36

The output of this program will be "Yes."

//QUESTION:-37

gross-salary - Variable names in C cannot contain hyphens (-). You should use underscores (_) or alphanumeric characters.
avg. - Variable names cannot contain a period (.) character. You should use underscores (_) or alphanumeric characters.

//QUESTION:-38

```
#include <stdio.h>

int main() {
    float tankCapacity = 175.0;
    float drainRate = 25.0;
    float timeRequired;

    timeRequired = tankCapacity / drainRate;

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

    return 0;
}
```

//QUESTION:-39

```
#include <stdio.h>

int main()
{
    float batteryPower = 0.75; // Battery power as a decimal (75%)
    float hours;
```

```
    hours = (1 - batteryPower) / (-0.2);
    printf("After %.2f hours, the battery power is at 75%%.\n", hours);
    return 0;
}
```

//QUESTION:-40

Compiler

//QUESTION:-41

%o

//QUESTION:-42

%.2e

QUESTION:-43

Array

QUESTION:-44

“hell”8

//QUESTION:-45

0, 5

//QUESTION:-46

Basic_pay

//QUESTION:-47

c1

//QUESTION:-48

a) Decimal to Binary:

```
 #include <stdio.h>

int main()
{
   int decimal = 365;
   int binary[32];
   int i = 0;

   while (decimal > 0)
{
      binary[i] = decimal % 2;
      decimal /= 2;
      i++;
   }

   printf("Binary equivalent: ");
   for (int j = i - 1; j >= 0; j--) {
      printf("%d", binary[j]);
   }

   return 0;
}
```

b) Decimal to Octal:

```
#include <stdio.h>

int main() {
   int decimal = 453;
   int octal[32];
   int i = 0;

   while (decimal > 0) {
      octal[i] = decimal % 8;
      decimal /= 8;
      i++;
   }

   printf("Octal equivalent: ");
   for (int j = i - 1; j >= 0; j--) {
      printf("%d", octal[j]);
```

```
    }

    return 0;
}
```

c) Decimal to Hexadecimal:

```
#include <stdio.h>

int main() {
    int decimal = 5164;
    char hexadecimal[32];
    int i = 0;

    while (decimal > 0) {
        int remainder = decimal % 16;
        if (remainder < 10) {
            hexadecimal[i] = '0' + remainder;
        } else {
            hexadecimal[i] = 'A' + (remainder - 10);
        }
        decimal /= 16;
        i++;
    }

    printf("Hexadecimal equivalent: ");
    for (int j = i - 1; j >= 0; j--) {
        printf("%c", hexadecimal[j]);
    }

    return 0;
}
```

d) Decimal to Base-5:

```
#include <stdio.h>

int main() {
    int decimal = 23;
    int base5[32];
    int i = 0;

    while (decimal > 0) {
```

```
        base5[i] = decimal % 5;
        decimal /= 5;
        i++;
    }

    printf("Base-5 equivalent: ");
    for (int j = i - 1; j >= 0; j--) {
        printf("%d", base5[j]);
    }

    return 0;
}
```

e) Decimal to Base-7:

```
#include <stdio.h>

int main() {
    int decimal = 772;
    int base7[32];
    int i = 0;

    while (decimal > 0) {
        base7[i] = decimal % 7;
        decimal /= 7;
        i++;
    }

    printf("Base-7 equivalent: ");
    for (int j = i - 1; j >= 0; j--) {
        printf("%d", base7[j]);
    }

    return 0;
}

//QUESTION:-49

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

float convertToDecimal(char* number, int base) {
    int integerPart = 0;
    float fractionalPart = 0.0;
```

```
    int decimalPosition = -1;
    int i = 0;

    while (number[i] != '.' && number[i] != '\0')
           {
        integerPart = integerPart * base + (number[i] - '0');
        i++;
    }

    if (number[i] == '.') {
        i++;
        while (number[i] != '\0') {
            fractionalPart = fractionalPart + ((number[i] - '0') / pow(base, decimalPosition));
            decimalPosition--;
            i++;
        }
    }

    return integerPart + fractionalPart;
}

int main()
{
    char number1[] = "325.54";  // Base-6
    char number2[] = "1001010110101.1110101";  // Base-2
    char number3[] = "742.72";  // Base-8
    char number4[] = "AC94.C5";  // Base-16

    float decimal1 = convertToDecimal(number1, 6);
    float decimal2 = convertToDecimal(number2, 2);
    float decimal3 = convertToDecimal(number3, 8);
    float decimal4 = convertToDecimal(number4, 16);

    printf("a) (%s)_6 = (%.4f)_10\n", number1, decimal1);
    printf("b) (%s)_2 = (%.7f)_10\n", number2, decimal2);
    printf("c) (%s)_8 = (%.2f)_10\n", number3, decimal3);
    printf("d) (%s)_16 = (%.2f)_10\n", number4, decimal4);

    return 0;
}

//QUESTION:-52

#include <stdio.h>
```

```
int main() {
   int a;

   // Equation 1: (23)10 = (17)A
   for (a = 2; a <= 16; a++) {
      if ((23 % 10) == (17 % a)) {
         printf("a) A = %d\n", a);
         break;
      }
   }

   // Equation 2: (21)16 = (41)A
   for (a = 2; a <= 16; a++) {
      if ((0x21) == (4 * 16 + 1 % a)) {
         printf("b) A = %d\n", a);
         break;
      }
   }

   // Equation 3: (32)8 = (101)A
   for (a = 2; a <= 8; a++) {
      if ((032) == (1 * a * a + 0 * a + 1)) {
         printf("c) A = %d\n", a);
         break;
      }
   }

   return 0;
}

//QUESTON:- 53
```

The range of a 16-bit signed integer in C is typically from -32,768 to 32,767. Therefore, attempting to store **`32770`** in a 16-bit signed integer will result in an overflow.