Q1-: Suppose you own a beer distributorship that sells Piels (ID number 1), Coors (ID number 2), Bud (ID number 3), and Iron City (ID number 4) by the case. Write a complete C program to a. Get the case inventory for each brand for the start of the week. b. Process all weekly sales and purchase records for each brand. c. Display out the final This problem has been solved! You'll get a detailed solution from a subject-matter expert that helps you learn core concepts. See answer Question: In C language Suppose you own a beer distributorship that sells Piels (ID number 1), Coors (ID number 2), Bud (ID number 3), and Iron City (ID number 4) by the case. Write a complete C program to a. Get the case inventory for each brand for the start of the week. b. Process all weekly sales and purchase records for each brand. c. Display out the final In C language Suppose you own a beer distributorship that sells Piels (ID number 1), Coors (ID number 2), Bud (ID number 3), and Iron City (ID number 4) by the case. Write a complete C program to a. Get the case inventory for each brand for the start of the week. b. Process all weekly sales and purchase records for each brand. c. Display out the final inventory. Each transaction will consist of two data items. The first item will be the brand ID number (an integer). The second will be the amount purchased (a positive integer value) or the amount sold (a negative integer value). For now you may assume that you alway have sufficient foresight to prevent depletion of your inventory for any brand.

Ans:- #include <stdio.h>

```
#define NUM_BRANDS 4

const char *brand_names[] = {"Piels", "Coors", "Bud", "Iron City"};

void display_inventory(int inventory[]) {
    printf("\nFinal Inventory:\n");
    for (int i = 0; i < NUM_BRANDS; i++) {
        printf("%s: %d cases\n", brand_names[i], inventory[i]);
    }
}

int main() {
    int inventory[NUM_BRANDS] = {0};
    int brand_id, quantity;

    printf("Enter the starting inventory for each brand:\n");
    for (int i = 0; i < NUM_BRANDS; i++) {
```

```
        printf("%s: ", brand_names[i]);
        scanf("%d", &inventory[i]);
    }

    printf("\nEnter transactions (brand ID and quantity):\n");
    printf("(Enter -1 for brand ID to stop)\n");
    while (1) {
        printf("Brand ID (1-4): ");
        scanf("%d", &brand_id);

        if (brand_id == -1) {
            break;
        }

        if (brand_id < 1 || brand_id > NUM_BRANDS) {
            printf("Invalid Brand ID! Please enter a number between 1 and 4.\n");
            continue;
        }

        printf("Quantity (positive for purchase, negative for sale): ");
        scanf("%d", &quantity);

        inventory[brand_id - 1] += quantity;
    }

    display_inventory(inventory);

    return 0;
}
```

Q-2:- Determine the following information about each value in a list of positive integers. (a) Is the value a multiple of 3, 5, or 8? (b) Is the sum of the digits divisible by 4? (c) Is the value a perfect square? You should write a function with three type int output parameters that send back the answers to these three questions. Some sample input data might be: 15 27 81 100 49 153

Ans:-

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

void check_properties(int num, int *is_multiple, int *is_sum_divisible, int *is_perfect_square);
int sum_of_digits(int num);
bool is_perfect_square(int num);

int main() {
   int numbers[] = {15, 27, 81, 100, 49, 153};
   int size = sizeof(numbers) / sizeof(numbers[0]);

   printf("Number  Multiple(3,5,8)  SumDiv4  PerfectSquare\n");
   printf("-------------------------------------------------\n");

   for (int i = 0; i < size; i++) {
      int is_multiple, is_sum_divisible, is_perfect_square;

      check_properties(numbers[i], &is_multiple, &is_sum_divisible, &is_perfect_square);

      printf("%-7d  %-15s  %-8s  %-13s\n",
           numbers[i],
           (is_multiple ? "Yes" : "No"),
           (is_sum_divisible ? "Yes" : "No"),
           (is_perfect_square ? "Yes" : "No"));
   }
```

```
    return 0;
}

void check_properties(int num, int *is_multiple, int *is_sum_divisible, int *is_perfect_square) {
    *is_multiple = (num % 3 == 0 || num % 5 == 0 || num % 8 == 0);
    *is_sum_divisible = (sum_of_digits(num) % 4 == 0);
    *is_perfect_square = is_perfect_square(num);
}

int sum_of_digits(int num) {
    int sum = 0;
    while (num > 0) {
        sum += num % 10;
        num /= 10;
    }
    return sum;
}

bool is_perfect_square(int num) {
    int sqrt_num = sqrt(num);
    return (sqrt_num * sqrt_num == num);
}
```

Set-A

Q1-:Create a C program to determine the real roots of a quadratic equation, if they exist, based on user input. Prompt the user to enter three coefficients (a, b, and c) for the equation ax2+bx+c=0. Write a function solve_quadratic that takes the coefficients as input arguments and has three output arguments: a flag indicating if real roots exist (1 for real roots, 0 for no real roots), and the two root values if they exist. In the main function, display the results with a message indicating whether real roots were found and, if so, display their values

Ans:-
```
#include <stdio.h>
#include <math.h>
```

```
void solve_quadratic(double a, double b, double c, int *has_real_roots, double *root1, double *root2) {
    double discriminant = b * b - 4 * a * c;

    if (discriminant < 0) {
        *has_real_roots = 0;
    } else {
        *has_real_roots = 1;
        *root1 = (-b + sqrt(discriminant)) / (2 * a);
        *root2 = (-b - sqrt(discriminant)) / (2 * a);
    }
}

int main() {
    double a, b, c;
    int has_real_roots;
    double root1, root2;

    printf("Enter coefficients a, b, and c for the equation ax^2 + bx + c = 0:\n");
    printf("a: ");
    scanf("%lf", &a);
    printf("b: ");
    scanf("%lf", &b);
    printf("c: ");
    scanf("%lf", &c);

    if (a == 0) {
        printf("This is not a quadratic equation (a cannot be 0).\n");
        return 1;
    }
```

```
    solve_quadratic(a, b, c, &has_real_roots, &root1, &root2);

    if (has_real_roots) {
        printf("Real roots found:\n");
        printf("Root 1: %.2lf\n", root1);
        printf("Root 2: %.2lf\n", root2);
    } else {
        printf("No real roots exist for the given equation.\n");
    }

    return 0;
}
```

Q-2:- You have saved $500 to use as a down payment on a car. Before beginning your car shopping, you decide to write a program to help you figure out what your monthly payment will be, given the car's purchase price, the monthly interest rate, and the time period over which you will pay back the loan. The formula for calculating your monthly payment is given by: payment = iP 1-(1+i)-n Where: • P Principal (the amount you borrow, which is the car's purchase price minus the down payment) i = Monthly interest rate (this is of the annual interest rate) n = Total number of payments (usually the loan term in months) Your program should prompt the user for the purchase price, the down payment, the annual interest rate and the total number of payments (usually 36, 48, or 60). It should then display the amount borrowed and the monthly payment including a dollar sign and two decimal places.

Ans:-

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

int main() {
    double purchase_price, down_payment, annual_interest_rate, monthly_interest_rate, principal, monthly_payment;
    int total_payments;

    printf("Enter the car's purchase price: $");
    scanf("%lf", &purchase_price);
```

```
    printf("Enter the down payment: $");
    scanf("%lf", &down_payment);
    printf("Enter the annual interest rate (as a percentage): ");
    scanf("%lf", &annual_interest_rate);
    printf("Enter the total number of payments (e.g., 36, 48, 60): ");
    scanf("%d", &total_payments);

    principal = purchase_price - down_payment;
    monthly_interest_rate = annual_interest_rate / 12 / 100;
    monthly_payment = (monthly_interest_rate * principal) / (1 - pow(1 + monthly_interest_rate, -
total_payments));

    printf("\nAmount borrowed: $%.2lf\n", principal);
    printf("Monthly payment: $%.2lf\n", monthly_payment);

    return 0;
}
```