1)Pattern printing using Nested Loops

ANSWER:

```
#include <stdio.h>
int main() {
   int rows, i = 1, j;
   printf(“Enter the number of rows: “);
   scanf(“%d”, &rows);

   while (i <= rows) {     // Outer loop for rows
      j = 1;
      While (j <= i) {     // Inner loop for printing stars
         printf(“* “);
         j++;
      }
      printf(“\n”);
      i++;
   }
   return 0;
}
```

2)Pyramid Pattern

ANSWER:

```
#include <stdio.h>
int main() {
```

```
   int rows, i = 1, j, k;
   printf("Enter the number of rows: ");
   scanf("%d", &rows);
 while (i <= rows) {
      j = 1;
      while (j <= rows - i) {
         printf(" ");
         j++;
      }
      k = 1;
      while (k <= 2 * i - 1) {
         printf("*");
         k++;
      }
      printf("\n");
      i++;
   }
   return 0;
}
```

3).Multiplication table

ANSWER:

```
#include <stdio.h>
int main() {
   int num, i = 1;
```

```
   printf("Enter a number to display its multiplication table: ");
   scanf("%d", &num);
  do {
      printf("%d x %d = %d\n", num, i, num * i);
      i++;
   } while (i <= 10);
  return 0;
}
```

4.Sum of N natural numbers using for loop

ANSWER:

```
#include<stdio.h>
int main()
{
int num,sum=0;
printf("Enter the number:");
scanf("%d",&num);
for(int i=0;i<=num;i++)
{
sum +=i;
}
printf("=%d",sum);
```

5)Reverse a number

ANSWER:

```
#include <stdio.h>
int main() {
   int num, reversed = 0, remainder;
   printf("Enter an integer: ");
   scanf("%d", &num);
   for (; num != 0; num /= 10) {  // Continue until num becomes 0
      remainder = num % 10;          // Get the last digit
      reversed = reversed * 10 + remainder;  // Build the reversed number
   }
   printf("Reversed Number: %d\n", reversed);
   return 0;
}
```

6)Fibonacci Series

ANSWER:

```
#include <stdio.h>
int main() {
   int n, first = 0, second = 1, next;
   printf("Enter the number of terms: ");
   scanf("%d", &n);
   printf("Fibonacci Series up to %d terms: \n", n);
   for (int i = 1; i <= n; i++) {
      if (i == 1) {
         printf("%d ", first);
```

```
        } else if (i == 2) {
            printf("%d ", second);
        } else {
            next = first + second;
            printf("%d ", next);
            first = second;
            second = next;
        }
    }
    printf("\n");
    return 0;
}
```

7)Pascals Triangle

ANSWER:

```
#include <stdio.h>
int main() {
    int n = 8, i, j, number;
    printf("Pascal's Triangle with 8 rows: \n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n - i - 1; j++) {
            printf(" ");
        }
        number = 1;
        for (j = 0; j <= i; j++) {
            printf("%d ", number);
```

```
        number = number * (i - j) / (j + 1);
      }
      printf("\n");
   }
   return 0;
}
```

8)Requirements

•In this challenge, you are going to create a "Guess the Number" C program

•Your program will generate a random number from 0 to 20

You will then ask the user to guess it

•User should only be able to enter numbers from 0-20

•The program will indicate to the user if each guess is too high or too low

The player wins the game if they can guess the number within five tries

Sample Output

This is a guessing game.

I have chosen a number between 0 and 20 which you must guess.

You have 5 tries left.

Enter a guess: 12

Sorry, 12 is wrong. My number is less than that.

You have 4 tries left.

Enter a guess: B Sorry, 8 is wrong. My number is less than that.

You have 3 tries left. Enter a guess: 4

Sorry, 4 is wrong. My number is less than that.

You have 2 tries left.

Enter a guess: 2

Congratulations. You guessed it!

ANSWER:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main() {
   int secret_number, guess, tries_left = 5;
   srand(time(0));
   secret_number = rand() % 21;
   printf("This is a guessing game.\n");
   printf("I have chosen a number between 0 and 20 which you must guess..\n");
   printf("You have 5 tries left.\n");
   while (tries_left > 0) {
      printf("Enter a guess: ");
      scanf("%d", &guess);
      if (guess < 0 || guess > 20) {
         printf("Please enter a number between 0 and 20.\n");
         continue;
      }
      if (guess == secret_number) {
         printf("Congratulations. You guessed it!\n");
         break;
      } else {
```

```
        if (guess < secret_number) {
            printf("Sorry, %d is wrong. My number is greater than that.\n", guess);
        } else {
            printf("Sorry, %d is wrong. My number is less than that.\n", guess);
        }
        tries_left--;
        printf("You have %d tries left.\n", tries_left);
      }
      if (tries_left == 0) {
        printf("Sorry, you've used all your tries. The number was %d.\n", secret_number);
      }
    }
    return 0;
}
```

9.Program to find sum of all even numbers entered by the user

ANSWER:

```
#include <stdio.h>
int main() {
    int num, sum = 0, count = 0;
    printf("Enter up to 20 integers (enter -1 to stop):\n");
    while (count < 20) {
        printf("Enter a number: ");
        scanf("%d", &num);
        if (num == -1) {
```

```
        break;
      }
      if (num < 0) {
        continue;
      }
      if (num % 2 == 0) {
        sum += num;
      }
      count++;
    }
    printf("Sum of even numbers: %d\n", sum);
    return 0;
}
```

10)Problem Statement 1: Banking System Simulation

Description: Create a simple banking system simulation that allows users to create an account, deposit money, withdraw money, and check their balance. The program should handle multiple accounts and provide a menu-driven interface.

Requirements:

1. Use appropriate data types for account balance (e.g., float for monetary values)

and user input (e.g., int for account numbers).

2. Use variables to hold(account number, account holder name, balance).

3. Use control statements to navigate through the menu options:

i. Create Account

ii. Deposit Money

iii. Withdraw Money

iv. Check Balance

4. Ensure that the withdrawal does not exceed the available balance and handle invalid

inputs gracefully.

Example Input/Output:

Welcome to the Banking System

1. Create Account

2. Deposit Money

3. Withdraw Money

4. Check Balance

5. Exit

Choose an option: 1

Enter account holder name: John Doe

Account created successfully! Account Number: 1001

Choose an option: 2

Enter account number: 1001

Enter amount to deposit: 500

Deposit successful! New Balance: 500.0

Choose an option: 3

Enter account number: 1001

Enter amount to withdraw: 200

Withdrawal successful! New Balance: 300.0

Choose an option: 4

Enter account number: 1001

Current Balance: 300.0

Choose an option: 5

Exiting the system.

ANSWER:

```
#include <stdio.h>
int main() {
   int account_number;
   char account_holder_name[50];
   float account_balance = 0.0;
   int option;
   printf("Welcome to the Banking System\n");
   while (1) {
      printf("1. Create Account\n");
      printf("2. Deposit Money\n");
      printf("3. Withdraw Money\n");
      printf("4. Check Balance\n");
      printf("5. Exit\n");
      printf("Choose an option: ");
      scanf("%d", &option);
      switch (option) {
         case 1:
            printf("Enter account number: ");
            scanf("%d", &account_number);
            printf("Enter account holder name: ");
            scanf("%s", account_holder_name);
            printf("Account created successfully!\n");
```

```
    break;
  case 2: {
    float deposit_amount;
    printf("Enter amount to deposit: ");
    scanf("%f", &deposit_amount);
    if (deposit_amount > 0) {
      account_balance += deposit_amount;
      printf("Deposit successful! New Balance: %.2f\n", account_balance);
    } else {
      printf("Invalid deposit amount.\n");
    }
    break;
  }
  case 3: {
    float withdraw_amount;
    printf("Enter amount to withdraw: ");
    scanf("%f", &withdraw_amount);
    if (withdraw_amount > 0 && withdraw_amount <= account_balance) {
      account_balance -= withdraw_amount;
      printf("Withdrawal successful! New Balance: %.2f\n", account_balance);
    } else if (withdraw_amount > account_balance) {
      printf("Insufficient balance.\n");
    } else {
      printf("Invalid withdrawal amount.\n");
    }
    break;
```

```
        }
        case 4:
            printf("Current Balance: %.2f\n", account_balance);
            break;
        case 5:
            printf("Exiting the system\n");
            return 0;
        default:
            printf("Invalid option. Please try again.\n");
    }
  }
}
```

11) Problem Statement 4: Weather Data Analysis

Description: Write a program that collects daily temperature data for a month and analyzes it to find the average temperature, the highest temperature, the lowest temperature, and how many days were above average.

Requirements:

1. Use appropriate data types (float for temperatures and int for days).

2. Store temperature data in an array.

3. Use control statements to calculate:

i. Average Temperature of the month.

ii. Highest Temperature recorded.

iii. Lowest Temperature recorded.

iv. Count of days with temperatures above average.

4. Handle cases where no data is entered.

Example Input/Output:

Enter temperatures for each day of the month (30 days):

Day 1 temperature: 72.5

Day 2 temperature: 68.0

...

Day 30 temperature: 75.0

Average Temperature of Month: XX.X

Highest Temperature Recorded: YY.Y

Lowest Temperature Recorded: ZZ.Z

Number of Days Above Average Temperature: N

ANSWER:

```
#include <stdio.h>
#define DAYS_IN_MONTH 30
int main() {
   float temperatures[DAYS_IN_MONTH];
   float total = 0, average, highest, lowest;
   int days_above_average = 0;
   printf("Enter temperatures for each day of the month (%d days):\n", DAYS_IN_MONTH);
   for (int i = 0; i < DAYS_IN_MONTH; i++) {
      printf("Day %d temperature: ", i + 1);
      scanf("%f", &temperatures[i]);
      total += temperatures[i];
      if (i == 0) {
         highest = temperatures[i];
         lowest = temperatures[i];
      } else {
```

```
            if (temperatures[i] > highest) {
                highest = temperatures[i];
            }
            if (temperatures[i] < lowest) {
                lowest = temperatures[i];
            }
        }
    }
    average = total / DAYS_IN_MONTH;
    for (int i = 0; i < DAYS_IN_MONTH; i++) {
        if (temperatures[i] > average) {
            days_above_average++;
        }
    }
    printf("\nAverage Temperature of Month: %.1f\n", average);
    printf("Highest Temperature Recorded: %.1f\n", highest);
    printf("Lowest Temperature Recorded: %.1f\n", lowest);
    printf("Number of Days Above Average Temperature: %d\n", days_above_average);

    return 0;
}
```

12) Problem Statement : Inventory Management System

Description: Create an inventory management system that allows users to manage products in a store. Users should be able to add new products, update existing product quantities, delete products, and view inventory details.

Requirements:

1. Use appropriate data types for product details (e.g., char arrays for product names, int

for quantities, float for prices).

2. Implement a structure to hold product information.

3. Use control statements for menu-driven operations:

i. Add Product

ii. Update Product Quantity

iii. Delete Product

iv. View All Products in Inventory

4. Ensure that the program handles invalid inputs and displays appropriate error

messages.

Example Input/Output:

Inventory Management System

1. Add Product

2. Update Product Quantity

3. Delete Product

4. View All Products in Inventory

5. Exit

Choose an option: 1

Enter product name: Widget A

Enter product quantity: 50

Enter product price: 19.99

Choose an option: 4

Product Name: Widget A, Quantity: 50, Price: $19.99

Choose an option: 5

Exiting the system.

ANSWER:

```
#include <stdio.h>
#include <string.h>
int main() {
   char product_name[50];
   int product_quantity = 0;
   float product_price = 0.0;
   int option;
   int product_exists = 0;
   printf("Welcome to the Inventory Management System\n");
   while (1) {
      printf("\n1. Add Product\n");
      printf("2. Update Product Quantity\n");
      printf("3. Delete Product\n");
      printf("4. View All Products in Inventory\n");
      printf("5. Exit\n");
      printf("Choose an option: ");
      scanf("%d", &option);
      switch (option) {
         case 1:
            printf("Enter product name: ");
            scanf("%s", product_name);
            printf("Enter product quantity: ");
            scanf("%d", &product_quantity);
```

```
    printf("Enter product price: ");
    scanf("%f", &product_price);
    product_exists = 1; // Product has been added
    printf("Product added successfully!\n");
    break;
  case 2:
    if (product_exists) {
      int new_quantity;
      printf("Enter new quantity for %s: ", product_name);
      scanf("%d", &new_quantity);
      product_quantity = new_quantity;
      printf("Product quantity updated successfully!\n");
    } else {
      printf("No product found to update. Please add a product first.\n");
    }
    break;
  case 3:
    if (product_exists) {
      product_exists = 0;
      printf("Product deleted successfully!\n");
    } else {
      printf("No product found to delete.\n");
    }
    break;
  case 4:
    if (product_exists) {
```

```
            printf("Product Name: %s, Quantity: %d, Price: $%.2f\n", product_name,
product_quantity, product_price);
        } else {
            printf("No products in inventory.\n");
        }
        break;
      case 5:
        printf("Exiting the system\n");
        return 0;
      default:
        printf("Invalid option. Please try again.\n");
    }
  }
}
```