Ridoy Karmakar

24103031

Section: B

Program: BCSE

1. The Fibonacci Series is the series where the next term is the sum of the previous two terms. For

Example: 0, 1, 1, 2, 3, 5, 5, , n

Write a program to print the Fibonacci series using call by reference.

N.B. The output will be written in the main function

```
// ----------------Ridoy karmakar-------------
//-----------------24103031-------------------
#include <iostream>
using namespace std;
void fibonacci(int n, int &a, int &b)
{
   for (int i = 0; i < n; i++)
   {
      cout << a << " ";
      int temp = a + b;
      a = b;
      b = temp;
   }
}
int main()
{
   int n;
   cin >> n;
   int f = 0, s = 1;
   fibonacci(n, f, s);
   return 0;
}
```

2. Rahul loves to play with numbers; he challenges his friend Ankush with a number- related problem in which he has to reverse the order of two digits and then swap them. Let us take the two digits as a and b.

Complete the functions reverse_dig() and swap() with arguments as a and b references. Don't return anything to the function.

```
// ----------------Ridoy karmakar-------------
//-----------------24103031-------------------

#include <iostream>
using namespace std;
void reverse_dig(int &num)
{
   int rev = 0;
   while (num > 0)
   {
      rev = rev * 10 + (num % 10);
      num /= 10;
   }
   num = rev;
}
void swap(int &a, int &b)
{
   int temp = a;
   a = b;
   b = temp;
}
```

```
int main()
{
   int a, b;
   cin >> a >> b;

   reverse_dig(a);
   reverse_dig(b);

   swap(a, b);

   cout << "After reversing and swapping:\n";
   cout << "a = " << a << "\n";
   cout << "b = " << b << "\n";

   return 0;
}
```

3. Create a C++ program that maintains employee records as arrays. Write a function that accepts an array of employee records and updates their salaries based on a given percentage increase.

```
// ----------------Ridoy karmakar-------------
//-----------------24103031-------------------

#include <iostream>
using namespace std;

struct Employee
{
```

```
    int id;
    string name;
    float salary;
};
void updateSalary(Employee emp[], int n, float percentage)
{
    for (int i = 0; i < n; i++)
    {
        emp[i].salary += emp[i].salary * (percentage / 100);
    }
}
void outputEmp(Employee emp[], int n)
{
    cout << "\nUpdated Employee Records:\n";
    for (int i = 0; i < n; i++)
    {
        cout << "ID: " << emp[i].id << ", Name: " << emp[i].name << ", Salary: " << emp[i].salary <<
endl;
    }
}
int main()
{
    int n;
    cin >> n;

    Employee emp[n];

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

```
        cout << "Enter details for employee " << i + 1 << ":\n";
        cout << "ID: ";
        cin >> emp[i].id;
        cout << "Name: ";
        cin >> emp[i].name;
        cout << "Salary: ";
        cin >> emp[i].salary;
    }
    float percentage;
    cout << "Enter your salary :" << endl;
    cin >> percentage;

    updateSalary(emp, n, percentage);

    outputEmp(emp, n);

    return 0;
}
```

4. In a retail store inventory system, implement a program using arrays that take the current stock quantity of products and update it after a customer purchases a certain quantity.

```
// ----------------Ridoy karmakar-------------
//-----------------24103031-------------------

#include <iostream>
using namespace std;

int main()
{
   int n;
   cout << "Enter number of products ";
   cin >> n;
   string p[n + 1];
   int stock[n + 1];
   for (int i = 1; i <= n; i++)
   {
      cout << "Enter product name: ";
      cin >> p[i];
      cout << "Enter stock for " << p[i] << ": ";
      cin >> stock[i];
   }

   cout << "\nAvailable products:\n";
   for (int i = 1; i <= n; i++)
   {
      cout << i << ". " << p[i] << " - " << stock[i] << " units\n";
   }
```

```
    int index, quantity;
    cout << "\nEnter product index to buy: ";
    cin >> index;
    cout << "Enter quantity to buy: ";
    cin >> quantity;

    if (quantity <= stock[index])
    {
        stock[index] -= quantity;
        cout << "Purchase successful!\n";
    }
    else
    {
        cout << "Not enough stock!\n";
    }
    cout << "\nUpdated stock:\n";
    for (int i = 1; i <= n; i++)
    {
        cout << p[i] << " - " << stock[i] << " units\n";
    }
}
```

5. Build a C++ program to manage bank accounts using arrays. Write a function that accepts an array of account balances and updates the account balance after a withdrawal or deposit operation.

```
// ----------------Ridoy karmakar-------------
//-----------------24103031-------------------

#include <iostream>
using namespace std;
void updateBalance(float balances[], int index, float amount, char type)
{
   if (type == 'd')
   {
      balances[index] += amount;
      cout << "Deposit successful!\n";
   }
   else if (type == 'w')
   {
      if (amount <= balances[index])
      {
         balances[index] -= amount;
         cout << "Withdrawal successful!\n";
      }
      else
      {
         cout << "Ei Modon balance nai\n";
      }
   }
   else
```

```
    {
        cout << "Invalid operation type!\n";
    }
}
int main()
{
    int n;
    cout << "Enter number of bank accounts: ";
    cin >> n;

    string name[100];
    float balances[100];

    for (int i = 1; i <= n; i++)
    {
        cin >> name[i];
        cin >> balances[i];
    }
    cout << "\nAccount Balances:\n";
    for (int i = 1; i <= n; i++)
    {
        cout << name[i] << " : " << balances[i] << " /-" << endl;
    }
    int index;
    float amount;
    char type;

    cout << "\nEnter account index: ";
    cin >> index;
```

```
    cout << "Enter operation type (d for deposit, w for withdraw): ";
    cin >> type;
    cout << "Enter amount: ";
    cin >> amount;

    updateBalance(balances, index, amount, type);

    cout << "\nUpdated Account Balances:\n";

    for (int i = 1; i <= n; i++)
    {
        cout << name[i] << " : " << balances[i] << " /-" << endl;
    }

    return 0;
}
```

6. Create a program using arrays to calculate students' final grades. Implement a function that takes an array of test scores and calculates the average score.

```
// ----------------Ridoy karmakar-------------
//-----------------24103031-------------------

#include <iostream>
using namespace std;
float calculateAverage(float scores[], int n)
{
    float sum = 0;
    for (int i = 0; i < n; i++)
```

```
    {
        sum += scores[i];
    }
    return sum / n;
}

int main()
{
    int n;
    cin >> n;
    float scores[100];
    for (int i = 0; i < n; i++)
    {
        cin >> scores[i];
    }

    float average = calculateAverage(scores, n);

    cout << "\n Final average score: " << average << endl;

    return 0;
}
```

7. Develop a temperature conversion program using arrays. Write a function that accepts an array of temperatures in Celsius and converts them to Fahrenheit.

```
// ----------------Ridoy karmakar-------------
//-----------------24103031-------------------

#include <iostream>
using namespace std;

void convertToFahrenheit(float celsius[], float fahrenheit[], int n)
{
   for (int i = 0; i < n; i++)
   {
      fahrenheit[i] = (celsius[i] * 9 / 5) + 32;
   }
}

int main()
{
   int n;
   cout << "Enter number of temperatures: ";
   cin >> n;

   float celsius[100], fahrenheit[100];

   for (int i = 0; i < n; i++)
   {
      cin >> celsius[i];
   }
```

```
    convertToFahrenheit(celsius, fahrenheit, n);

    cout << "\nTemperatures in Fahrenheit:\n";

    for (int i = 0; i < n; i++)
    {
        cout << "Celsius: " << celsius[i] << " => Fahrenheit: " << fahrenheit[i] << endl;
    }

    return 0;
}
```

8. Create a program using arrays to manipulate vectors of integers. Implement a function that takes an array of integers and squares each element in the array.

```
// ----------------Ridoy karmakar-------------
//-----------------24103031-------------------

#include <iostream>
using namespace std;

void squareElements(int arr[], int n)
{
    for (int i = 0; i < n; i++)
    {
        arr[i] = arr[i] * arr[i];
    }
}
```

```
int main()
{
    int n;
    cin >> n;

    int arr[100];

    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }

    squareElements(arr, n);

    cout << "\nSquared elements:\n";
    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }

    cout << endl;
    return 0;
}
```

9. Build a library inventory management system using arrays. Write a function that accepts an array of book records and updates the number of copies available after a book is borrowed.

```
// ----------------Ridoy karmakar-------------
//-----------------24103031-------------------

#include <iostream>
using namespace std;

void borrowBook(string books[], int copies[], int index, int n)
{
   if (index >= 1 && index <= n)
   {
      if (copies[index] > 0)
      {
         copies[index]--;
         cout << "You borrowed: " << books[index] << endl;
      }
      else
      {
         cout << "Sorry! No copies available for " << books[index] << endl;
      }
   }
   else
   {
      cout << "Invalid book index!" << endl;
   }
}
```

```
int main()
{
    int n;
    cin >> n;
    string books[100];
    int copies[100];

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

        // cin.ignore();
        // getline(cin, books[i]);
        cin >> books[i];
        cin >> copies[i];
    }

    cout << "\nLibrary Inventory:\n";
    for (int i = 1; i <= n; i++)
    {
        cout << i << ". " << books[i] << " - " << copies[i] << " copies\n";
    }

    int index;
    cin >> index;

    borrowBook(books, copies, index, n);

    cout << "\nUpdated Inventory:\n";
    for (int i = 1; i <= n; i++)
```

```
    {
        cout << books[i] << " - " << copies[i] << " copies\n";
    }

    return 0;
}
```

10. Design an online shopping cart program using arrays. Implement a function that takes an array of product prices and applies a discount based on a given percentage.

```
// ----------------Ridoy karmakar-------------
//-----------------24103031-------------------

#include <iostream>
using namespace std;

void applyDiscount(float prices[], int n, float discountPercent)
{
    for (int i = 1; i <= n; i++)
    {
        prices[i] = prices[i] - (prices[i] * discountPercent / 100);
    }
}

int main()
{
    int n;
    cin >> n;
    float prices[100];
```

```
    for (int i = 1; i <= n; i++)
    {
        cin >> prices[i];
    }

    float discount;
    cin >> discount;

    applyDiscount(prices, n, discount);

    cout << "\nPrices after discount:\n";
    for (int i = 1; i <= n; i++)
    {
        cout << "Product " << i << ": " << prices[i] << "/-" << endl;
    }

    return 0;
}
```

11. Create a program to track car maintenance using arrays. Write a function that accepts an array of car mileages and updates them after a service is performed.

```
// ----------------Ridoy karmakar-------------
//-----------------24103031-------------------

#include <iostream>
using namespace std;

void updateMileage(int mileages[], int n, int serviceMileage)
{
   for (int i = 0; i < n; i++)
   {
      mileages[i] += serviceMileage;
   }
}

int main()
{
   int n;
   cin >> n;

   int mileages[100];

   for (int i = 0; i < n; i++)
   {
      cin >> mileages[i];
   }
```

```
    int serviceMileage;

    cin >> serviceMileage;

    updateMileage(mileages, n, serviceMileage);

    cout << "\nUpdated mileages after service:\n";
    for (int i = 0; i < n; i++)
    {
        cout << "Car " << i + 1 << ": " << mileages[i] << " km" << endl;
    }

    return 0;
}
```

12. Develop a customer loyalty points system using arrays. Implement a function that takes an array of customer points balance and updates them based on their recent purchase amounts and bonus points.

```
// ----------------Ridoy karmakar-------------
//-----------------24103031-------------------

#include <iostream>
using namespace std;
void updatePoints(int points[], float purchases[], int bonus[], int n)
{
    for (int i = 0; i < n; i++)
    {
```

```
        int earnedPoints = (int)(purchases[i] * 0.10);
        points[i] += earnedPoints + bonus[i];
    }
}

int main()
{
    int n;
    cin >> n;
    int points[100], bonus[100];
    float purchases[100];

    for (int i = 0; i < n; i++)
    {
        cin >> points[i];
        cin >> purchases[i];
        cin >> bonus[i];
    }

    updatePoints(points, purchases, bonus, n);

    cout << "\nUpdated Customer Points:\n";
    for (int i = 0; i < n; i++)
    {
        cout << "Customer " << i + 1 << ": " << points[i] << " points\n";
    }

    return 0;
}
```