ASSIGNMENT FOR DATA STRUCTURES (EC2022E)

BADHON DATTA PROTTOY

ROLL: B230101EC

EC01

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

SET 1

QUESTION NO 1:

CODE:

```
#include <iostream>
#include <iomanip>
#include <string>

using namespace std;
int main(){
    string items[5]={"potatoes","tomatoes","onion","chilli","jackfruit"};
    double cost[5]={25.00, 50.00, 30.00, 70.00, 50.50};
    int quantities[5];


     for (int i = 0; i < 5; i++) {
        cout << "Enter quantity(KG) for " << items[i] << " (Price: RS" << cost[i] << "): ";
        cin >> quantities[i];
     }
     double totalbill=0.0;
     cout << fixed << setprecision(2);
    cout << endl << "Your Bill:" << endl;
    cout << setw(15) << "Item" << setw(25) << "Quantity(in kg)" << setw(15) << "cost" <<
setw(15) << "Total(rupees)" << endl;
    cout << setfill('-') << setw(80) << "-" << setfill(' ') << endl;

 for (int i = 0; i < 5; i++) {
        double totalPrice = cost[i] * quantities[i];
        totalbill += totalPrice;
        cout << setw(15) << items[i]
             << setw(25) << quantities[i]
             << setw(15) << cost[i]
             << setw(15) << totalPrice << endl;
    }

    cout << setfill('-') << setw(80) << "-" << setfill(' ') << endl;
    cout << setw(60) << "Total Bill:" << setw(15) << totalbill <<endl;
    return 0;

}
```

OUTPUT:

```
Enter quantity(KG) for potatoes (Price: RS25): 10
Enter quantity(KG) for tomatoes (Price: RS50): 8
Enter quantity(KG) for onion (Price: RS30): 3
Enter quantity(KG) for chilli (Price: RS70): 2
Enter quantity(KG) for jackfruit (Price: RS50.5): 3

Your Bill:
          Item          Quantity(in kg)          cost  Total(rupees)
-------------------------------------------------------------------------
------
       potatoes                       10         25.00         250.00
       tomatoes                        8         50.00         400.00
          onion                        3         30.00          90.00
         chilli                        2         70.00         140.00
      jackfruit                        3         50.50         151.50
-------------------------------------------------------------------------
------
                                                Total Bill:        1031.5
0
```

QUESTION NO 2:

CODE:

```
#include <iostream>
using namespace std;
int main(){
    int marks;
    cout<<"Enter your marks:"<<endl;
    cin>> marks;

   char grade;
   switch(marks/10){

       case 9:  // 90-100
          grade = 'S';
          break;
       case 8:  // 80-89
          grade = 'A';
          break;
       case 7:  // 70-79
          grade = 'B';
          break;
       case 6:  // 60-69
          grade = 'C';
```

```
            break;
        case 5:  // 50-59
            grade = 'D';
            break;
        case 4:  // 40-49
            grade = 'E';
            break;
        default: // 0-39
            grade ='F';
            break;

    }
    cout<<"your grade is :"<< grade <<endl;
    return 0;
}
```

OUTPUT:

```
Enter your marks:
91
your grade is :S
```

QUESTION NO 3:

CODE:

```
#include <iostream>
#include <vector>
using namespace std;
int main (){
    int choice;
    int numOperands;
    vector<double> operands;

    cout << "Simple Calculator" << endl;
    cout << "Select an operation:" << endl;
    cout << "1. Add" << endl;
    cout << "2. Subtract" << endl;
    cout << "3. Multiply" << endl;
    cout << "4. Divide" << endl;
    cout << "Enter your choice (1-4): ";
    cin >> choice;
```

```
cout << "Enter the number of operands: ";
cin >> numOperands;

operands.resize(numOperands);
cout << "Enter " << numOperands << " operands:" << endl;
for (int i = 0; i < numOperands; i++) {
    cout << "Operand " << (i + 1) << ": ";
    cin >> operands[i];
}
double result=operands[0];
switch (choice){
    case 1:
     for(int i=1;i<numOperands;i++){
        result+=operands[i];
    }
    cout<<"Result(addition) is:"<<result<<endl;
    break;

    case 2:
     for(int i=1;i<numOperands;i++){
        result-=operands[i];
    }
    cout<<"Result(substraction) is:"<<result<<endl;
    break;

    case 3:
     for(int i=1;i<numOperands;i++){
        result*=operands[i];
    }
    cout<<"Result(multiplication) is:"<<result<<endl;
    break;

    case 4:
     for(int i=1;i<numOperands;i++){
        if (operands[i] == 0) {
                cout << "Error: Division by zero!" << endl;
                return 1;
        }
            result /= operands[i];
    }
    cout<<"Result(division) is:"<<result<<endl;
    break;
    default:
    cout<<"invalid choice"<<endl;
```

```
        break;


    }

    return 0;
}
```

OUTPUT:

```
Simple Calculator
Select an operation:
1. Add
2. Subtract
3. Multiply
4. Divide
Enter your choice (1-4): 3
Enter the number of operands: 4
Enter 4 operands:
Operand 1: 44
Operand 2: 3
Operand 3: 54
Operand 4: 6
Result(multiplication) is:42768
```

QUESTION NO 4:

CODE:

```
#include<iostream>
using namespace std;
int main(){
    const int rows=3;
    const int colms=3;
    int matrix [rows][colms];

    for(int i=0; i<rows; i++){
        for(int j=0; j<colms; j++){
            cout<<"enter the matrix value <<[" << i+1 << "][" << j+1 << "]:";
            cin>>matrix[i][j];


        }
    }

   cout<< "The matrix is:"<<endl;
   for(int i=0; i<rows; i++) {
```

```
    for (int j=0; j<colms; j++ ){
        cout<<matrix[i][j]<<" ";

    }
    cout<<endl;
   }
   return 0;

}
```

OUTPUT:

```
enter the matrix value <<[1][1]:45
enter the matrix value <<[1][2]:65
enter the matrix value <<[1][3]:85
enter the matrix value <<[2][1]:37
enter the matrix value <<[2][2]:25
enter the matrix value <<[2][3]:73
enter the matrix value <<[3][1]:78
enter the matrix value <<[3][2]:53
enter the matrix value <<[3][3]:88
The matrix is:
45 65 85
37 25 73
78 53 88
```

QUESTION NO 5:

CODE:

```
#include<iostream>
using namespace std;
int main(){
    int rows=6;
    for (int i=1; i<=rows; i++){
        for(int j=1; j<=i; j++){
            cout<<"*";
        }
        cout<<endl;
    }
   return 0;
}
```

OUTPUT:

```
*
**
***
****
*****
******
```

QUESTION NO 6:

CODE:

```
#include <iostream>
#include <iomanip>
using namespace std;

bool isLeapYear(int year) {
    return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
}

int getStartDay(int month, int year) {
    int m = month;
    int y = year;

    if (month < 3) {
        m += 12;
        y--;
    }

    int k = y % 100;
    int j = y / 100;

    int startDay = (1 + ((13 * (m + 1)) / 5) + k + (k / 4) + (j / 4) + (5 * j)) % 7;

    return (startDay + 6) % 7;
}

int main() {
    int month = 8;
    int year = 2014;

    int daysInMonth[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
```

```
    if (month == 2 && isLeapYear(year)) {
        daysInMonth[1] = 29;
    }


    int startDay = getStartDay(month, year);

    cout << "Sun Mon Tue Wed Thu Fri Sat" << endl;

    for (int i = 0; i < startDay; i++) {
        cout << "    ";
    }


    for (int day = 1; day <= daysInMonth[month - 1]; day++) {
        cout << setw(3) << day << " ";


        if ((day + startDay) % 7 == 0) {
            cout << endl;
        }
    }

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

OUTPUT:

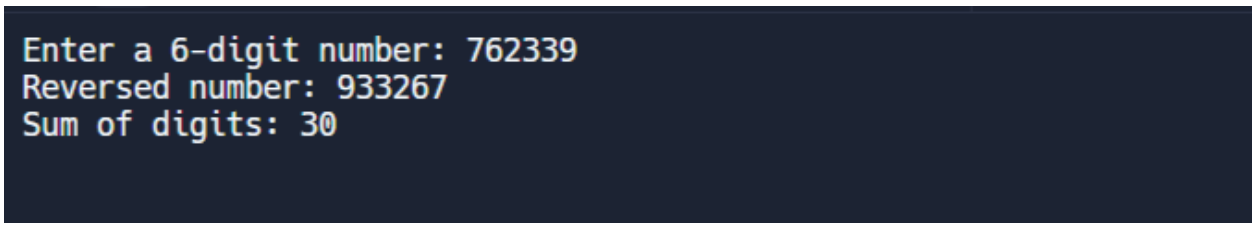
```
Sun Mon Tue Wed Thu Fri Sat
                        1   2
  3   4   5   6   7   8   9
 10  11  12  13  14  15  16
 17  18  19  20  21  22  23
 24  25  26  27  28  29  30
 31
```

QUESTION NO 7:

CODE:

```
#include <iostream>
using namespace std;

int main() {
    int number;
    int sum = 0;
    int reversedNumber = 0;

    cout << "Enter a 6-digit number: ";
    cin >> number;

    if (number < 100000 || number > 999999) {
        cout << "Please enter a valid 6-digit number." << endl;
        return 1;
    }


    int temp = number;
    while (temp > 0) {
        int digit = temp % 10;
        sum += digit;
        reversedNumber = reversedNumber * 10 + digit;
        temp /= 10;
    }

    cout << "Reversed number: " << reversedNumber << endl;
    cout << "Sum of digits: " << sum << endl;

    return 0;
}
```

OUTPUT:



SET 2

QUESTION NO 1:

CODE:

```
#include <iostream>
#include <cmath>
using namespace std;

// Function to calculate factorial
long long factorial(int n) {
    if (n == 0 || n == 1)
        return 1;
    return n * factorial(n - 1);
}

int main() {
    double x, sum = 0.0;
    int terms;


    cout << "Enter the value of x: ";
    cin >> x;

    cout << "Enter the number of terms: ";
    cin >> terms;


    for (int i = 1; i <= terms; i++) {
        double term = pow(x, 2 * i - 1) / factorial(2 * i - 2);
        if (i % 2 == 0)
            term = -term;

        sum += term;
    }

    cout << "The sum of the series is: " << sum << endl;

    return 0;
}
```

OUTPUT:

```
Enter the value of x: 6
Enter the number of terms: 4
The sum of the series is: -166.8
```

QUESTION NO 2:
CODE:

```
#include <iostream>
#include <iomanip>
using namespace std;

void printPattern() {
    const int width = 8;

    for (int i = 1; i <= 5; ++i) {

        cout << setw((5 - i) * width) << "";

        // nested loop for pattern elements
        for (int j = 1; j <= i; ++j) {
            cout << setw(width) << "x^" << i << "+" << j;
        }

        cout << endl;
    }
}

int main() {
    printPattern();
    return 0;
}
```

OUTPUT:

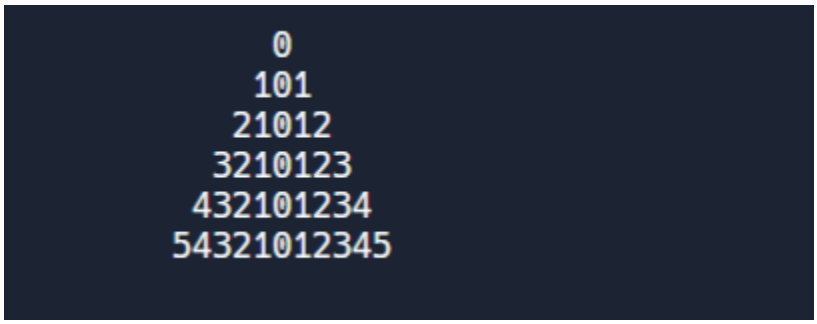
```
                                x^1+1
                            x^2+1       x^2+2
                        x^3+1       x^3+2       x^3+3
                    x^4+1       x^4+2       x^4+3       x^4+4
                x^5+1       x^5+2       x^5+3       x^5+4       x^5+5
```

QUESTION NO 3:
CODE:

```
#include <iostream>
#include <iomanip>
using namespace std;

void printPattern() {
```

```
    const int width = 7;

    for (int i = 0; i < 6; ++i) {
        cout << setw((2 * width - 1) - i) << "";
        for (int j = i; j >= 0; --j)
            cout << j;
        for (int j = 1; j <= i; ++j)
            cout << j;
        cout << endl;
    }
}

int main() {
    printPattern();
    return 0;
}
```

OUTPUT:



QUESTION NO 4:

CODE:

```
#include <iostream>
#include <algorithm>
#include <string>
using namespace std;
int rearrangeNumber(int num) {
    string digits = to_string(num);
    sort(digits.rbegin(), digits.rend());
    return stoi(digits); // Convert back to integer
}

int main() {
    int num1, num2, num3;

    // Input three 6-digit numbers
```

```
    cout << "Enter three 6-digit numbers: ";
    cin >> num1 >> num2 >> num3;


    int modNum1 = rearrangeNumber(num1);
    int modNum2 = rearrangeNumber(num2);
    int modNum3 = rearrangeNumber(num3);

    int largest = max(modNum1, max(modNum2, modNum3)); // Find the largest number

    cout << "Modified numbers are: " << modNum1 << ", " << modNum2 << ", " << modNum3 <<
endl;
    cout << "The largest modified number is: " << largest << endl;

    return 0;
}
```

OUTPUT:

```
Enter three 6-digit numbers: 987245
984765
984357
Modified numbers are: 987542, 987654, 987543
The largest modified number is: 987654
```

QUESTION NO 5:
CODE:

```
#include <iostream>
using namespace std;

// Function to calculate factorial
long long factorial(int num) {
    long long fact = 1;
    for (int i = 1; i <= num; i++) {
        fact *= i;
    }
    return fact;
}

// Function to generate Fibonacci numbers
void fibonacciFactorial(int n) {
    int a = 0, b = 1;
    cout << "Num (n)\tFactorial (n!)\n";
```

```
    for (int i = 1; i <= n; i++) {
        cout << b << "\t\t" << factorial(b) << "\n";
        int next = a + b;
        a = b;
        b = next;
    }
}

int main() {
    int n;
    cout << "Enter the number of Fibonacci numb to show: ";
    cin >> n;
    fibonacciFactorial(n);
    return 0;
}
```

OUTPUT:

```
Enter the number of Fibonacci numb to show: 6
Num (n) Factorial (n!)
1       1
1       1
2       2
3       6
5       120
8       40320
```

QUESTION NO 6:
CODE:

```
#include <iostream>
#include <cctype>
using namespace std;


bool isVowel(char ch) {

    ch = tolower(ch);

    // Check if character is a vowel
    if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
        return true;
    }
    return false;
```

```
}

int main() {
    string sentence;
    int vowelCount = 0;
    int nonAlphaCount = 0;
    char firstVowel = '\0'; // To store the first vowel ,if found

    // Ask user for input
    cout << "Enter a sentence: ";
    getline(cin, sentence);

    // Loop through each character in the sentence
    for (int i = 0; i < sentence.length(); i++) {
        char ch = sentence[i];

        // Check if the character is alphabetic
        if (isalpha(ch)) {
            if (isVowel(ch) && firstVowel == '\0') {
                firstVowel = ch; // Store the first vowel
            }

            if (isVowel(ch)) {
                vowelCount++;
            }
        } else {
            // If it is not alphabetic, increment the non-alphabetic count
            nonAlphaCount++;
        }
    }

    // Print the results
    cout << "Number of vowels: " << vowelCount << endl;
    cout << "Number of non-alphabetic characters: " << nonAlphaCount << endl;

    // Check if a vowel was found
    if (firstVowel != '\0') {
        cout << "First vowel: " << firstVowel << endl;
    } else {
        cout << "No vowels found." << endl;
    }

    return 0; // End the program
}
```

OUTPUT:

```
Enter a sentence: My name is Badhon Datta Prottoy
Number of vowels: 9
Number of non-alphabetic characters: 5
First vowel: a
```

QUESTION NO 7:
CODE:

```
#include <iostream>
#include <string>
using namespace std;

string removeConsecutiveDuplicates(const string &str) {
    string result = "";
    for (size_t i = 0; i < str.size(); i++) {
        if (i == 0 || str[i] != str[i - 1]) {
            result += str[i];
        }
    }
    return result;
}

int main() {
    string input;
    cout << "Enter a string: ";
    getline(cin, input);

    int originalLength = input.length();
    string processed = removeConsecutiveDuplicates(input);
    int processedLength = processed.length();

    cout << "Original string length: " << originalLength << "\n";
    cout << "Processed string length: " << processedLength << "\n";
    cout << "Processed string: " << processed << "\n";
    return 0;
}
```

OUTPUT:

```
Enter a string: Programming is FUN
Original string length: 18
Processed string length: 17
Processed string: Programing is FUN
```