

Assignment 01

1.1 WAP to create an enum (Day) containing 7 day names as its attributes. Prompt the user to enter a day number as int type. Assign this day number to a variable of enum Day. Using switch display the corresponding day name of the given day number. Pass the variable of Day in switch and make the cases as attributes of Day.

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

enum Day
{
    Sunday,
    Monday,
    Tuesday,
    Wednesday,
    Thursday,
    Friday,
    Saturday
};

int main()
{
    int dayNumber;

    cout << "Enter a day number (0 for Sunday, 1 for Monday, ..., 6 for Saturday):
";
    cin >> dayNumber;

    if (dayNumber < 0 || dayNumber > 6)
    {
        cout << "Invalid day number. Please enter a number between 0 and 6." <<
endl;
        return 1;
    }

    Day day = static_cast<Day>(dayNumber);

    switch (day)
```

```

{
case Sunday:
    cout << "The day is Sunday." << endl;
    break;
case Monday:
    cout << "The day is Monday." << endl;
    break;
case Tuesday:
    cout << "The day is Tuesday." << endl;
    break;
case Wednesday:
    cout << "The day is Wednesday." << endl;
    break;
case Thursday:
    cout << "The day is Thursday." << endl;
    break;
case Friday:
    cout << "The day is Friday." << endl;
    break;
case Saturday:
    cout << "The day is Saturday." << endl;
    break;
default:
    cout << "Unexpected error." << endl;
    break;
}

return 0;
}

```

1.2 Write a function using reference variables as arguments to swap the values of a pair of integers.

```

#include <iostream>
using namespace std;

void swapValues(int &a, int &b)
{
    int temp = a;

```

```

    a = b;
    b = temp;
}

int main()
{
    int x = 5;
    int y = 10;

    cout << "Before swapping: x = " << x << ", y = " << y << endl;

    swapValues(x, y);

    cout << "After swapping: x = " << x << ", y = " << y << endl;

    return 0;
}

```

1.3 Write a function that creates a vector of user given size M using new operator.

```

#include <iostream>
using namespace std;

int *createArray(int M)
{
    int *arr = new int[M];
    return arr;
}

int main()
{
    int user_size;
    cout << "Enter the size of the array: ";
    cin >> user_size;

    int *myArr = createArray(user_size);

    for (int i = 0; i < user_size; ++i)
    {

```

```

        myArr[i] = i;
        cout << myArr[i] << " ";
    }

    delete[] myArr;
    return 0;
}

```

1.4 Write a program to read a matrix of size m*n from the keyboard using new operator.

```

#include <iostream>
using namespace std;

void readMatrix(int **matrix, int m, int n)
{
    for (int i = 0; i < m; ++i)
    {
        for (int j = 0; j < n; ++j)
        {
            cout << "Enter element [" << i + 1 << "][" << j + 1 << "]: ";
            cin >> matrix[i][j];
        }
    }
}

int main()
{
    int m, n;
    cout << "Enter the number of rows (m): ";
    cin >> m;
    cout << "Enter the number of columns (n): ";
    cin >> n;

    int **matrix = new int *[m];
    for (int i = 0; i < m; ++i)
    {
        matrix[i] = new int[n];
    }
}

```

```

readMatrix(matrix, m, n);

for (int i = 0; i < m; ++i)
{
    for (int j = 0; j < n; ++j)
    {
        cout << matrix[i][j] << " ";
    }
    cout << "\n";
}

for (int i = 0; i < m; ++i)
{
    delete[] matrix[i];
}
delete[] matrix;

return 0;
}

```

1.5 Write a program to read a matrix of size $m \times n$ from the keyboard and display the same on the screen using function

```

#include <iostream>
using namespace std;

void readMatrix(int **matrix, int m, int n)
{
    for (int i = 0; i < m; ++i)
    {
        for (int j = 0; j < n; ++j)
        {
            cout << "Enter element [" << i + 1 << "][" << j + 1 << "]: ";
            cin >> matrix[i][j];
        }
    }
}

void displayMatrix(int **matrix, int m, int n)

```

```

{
    cout << "Matrix:\n";
    for (int i = 0; i < m; ++i)
    {
        for (int j = 0; j < n; ++j)
        {
            cout << matrix[i][j] << " ";
        }
        cout << "\n";
    }
}

int main()
{
    int m, n;
    cout << "Enter the number of rows (m): ";
    cin >> m;
    cout << "Enter the number of columns (n): ";
    cin >> n;

    int **matrix = new int *[m];
    for (int i = 0; i < m; ++i)
    {
        matrix[i] = new int[n];
    }

    readMatrix(matrix, m, n);
    displayMatrix(matrix, m, n);

    for (int i = 0; i < m; ++i)
    {
        delete[] matrix[i];
    }
    delete[] matrix;

    return 0;
}

```

1.6 Rewrite the Problem no. 1.5 to make the row parameter of the matrix as a default argument.

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

void readMatrix(int **matrix, int m, int n)
{
    for (int i = 0; i < m; ++i)
    {
        for (int j = 0; j < n; ++j)
        {
            cout << "Enter element [" << i + 1 << "][" << j + 1 << "]: ";
            cin >> matrix[i][j];
        }
    }
}

void displayMatrix(int **matrix, int m, int n)
{
    cout << "Matrix:\n";
    for (int i = 0; i < m; ++i)
    {
        for (int j = 0; j < n; ++j)
        {
            cout << matrix[i][j] << " ";
        }
        cout << "\n";
    }
}

void createAndDisplayMatrix(int n, int m = 3)
{
    int **matrix = new int *[m];
    for (int i = 0; i < m; ++i)
    {
        matrix[i] = new int[n];
    }

    readMatrix(matrix, m, n);
}
```

```

    displayMatrix(matrix, m, n);

    for (int i = 0; i < m; ++i)
    {
        delete[] matrix[i];
    }
    delete[] matrix;
}

int main()
{
    int n, m;
    cout << "Enter the number of columns (n): ";
    cin >> n;
    cout << "Enter the number of rows (m), default is 3: ";
    if (!(cin >> m))
    {
        m = 3;
        cin.clear();
        cin.ignore(10000, '\n');
    }

    createAndDisplayMatrix(n, m);
    return 0;
}

```

1.7 Write a macro that obtains the largest of the three numbers.

```

#include <iostream>
using namespace std;

// Correct macro definition with parentheses around each argument
#define MAX_OF_THREE(a, b, c) (((a) > (b)) ? (((a) > (c)) ? (a) : (c)) : (((b) > (c)) ? (b) : (c)))

int main()
{
    int x, y, z;
    cout << "Enter three numbers: ";
}

```



```

cin >> x >> y >> z;

// Pass correct variables to the macro
cout << "Largest number: " << MAX_OF_THREE(x, y, z) << endl;

return 0;
}

```

1.8 Write a function power() to raise a number m to power n. The function takes a double value for m and int value for n and returns the result correctly. Use a default value of 2 for n to make the function to calculate the squares when this argument is omitted. Write a main that gets the values of m and n from the user to test the function.

```

#include <iostream>
using namespace std;

// Function to raise a number m to power n
double power(double m, int n = 2) {
    double result = 1;
    for (int i = 0; i < n; i++) result *= m;
    return result;
}

int main() {
    double m;
    int n;

    // Prompt user for base value
    cout << "Enter a base and an exponent value (Press enter after just the base value to calculate square by default): ";
    cin >> m;

    if (cin.peek() == '\n') {
        cin.ignore(); // Ignore the newline character
        cout << "Result (square): " << power(m) << endl; // Calculate square
    } else {
        cin >> n;
        cout << "Result (" << m << " raised to the power of " << n << "): " <<

```

```

power(m, n) << endl; // Calculate with provided exponent
}

return 0;
}

```

1.9 Write a function that performs the same operation as that of Problem no. 1.8 but takes an int value for m. Both the functions should have the same name. Write a main that calls both the functions. Use the concept of function overloading.

```

#include <iostream>
using namespace std;

// Function to raise an integer to a power
int power(int m, int n = 2) {
    int result = 1;
    for (int i = 0; i < n; i++) result *= m;
    return result;
}

// Function to raise a double to a power
double power(double m, int n = 2) {
    double result = 1;
    for (int i = 0; i < n; i++) result *= m;
    return result;
}

int main() {
    int m1;
    double m2;
    int n;

    // Prompt user for integer base and exponent
    cout << "Enter an integer base and an exponent value (Press enter after just
the base value to calculate square by default): ";
    cin >> m1;

    if (cin.peek() == '\n') {
        cin.ignore(); // Ignore the newline character
    }
}

```

```

        cout << "Result (square): " << power(m1) << endl; // Calculate square
    } else {
        cin >> n;
        cout << "Result (" << m1 << " raised to the power of " << n << "): " <<
power(m1, n) << endl; // Calculate with provided exponent
    }

    // Prompt user for double base and exponent
    cout << "Enter a double base and an exponent value (Press enter after just the
base value to calculate square by default): ";
    cin >> m2;

    if (cin.peek() == '\n') {
        cin.ignore(); // Ignore the newline character
        cout << "Result (square): " << power(m2) << endl; // Calculate square
    } else {
        cin >> n;
        cout << "Result (" << m2 << " raised to the power of " << n << "): " <<
power(m2, n) << endl; // Calculate with provided exponent
    }

    return 0;
}

```