```
//1)


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


int main() {
	int i, iInt;
	char a, aChar;
	long g;
	float f;
	double d, dDouble;


	// Assign values
	i = 65;
	a = 'r';
	g = 241;
	f = 2.06;
	d = 122.448;


	cout << "The integer " << i << " converted to a character "
		<< "is " << static_cast<char>(i) << "\n";
```

```
cout << "The character " << a << " converted to an integer "
        << "is " << static_cast<int>(a) << "\n";
cout << "The long integer " << g << " converted to a character "
        << "is " << static_cast<char>(g) << "\n";
cout << "The float value " << f << " converted to an integer "
        << "is " << static_cast<int>(f) << "\n";
cout << "The double value " << d << " converted to a character "
        << "is " << static_cast<char>(d) << "\n";
cout << "\n\n";

iInt = g + i;
aChar = i + a;
dDouble = d + g;

cout << "The long integer " << g << " + " << " the integer "
        << i << " converted to an integer " << "is "
        << static_cast<int>(iInt) << "\n";

cout << "The long integer " << g << " + " << " the integer "
        << i << " converted to a character " << "is "
        << static_cast<char>(iInt) << "\n";
cout << "The integer " << i << " + " << " the character "
```

```
        << a << " converted to an integer " << "is "
        << static_cast<int>(aChar) << "\n";
    cout << "The integer " << i << " + " << " the character "
        << a << " converted to a character " << "is "
        << static_cast<char>(aChar) << "\n";
    cout << "The integer " << i << " * " << " the character "
        << a << " converted to an integer " << "is "
        << static_cast<double>(d * a) << "\n";
    cout << "The double " << d << " / " << " the character "
        << a << " converted to an integer " << "is "
        << static_cast<int>(d / a) << "\n";
    cout << "The double " << d << " + " << " the long integer "
        << g << " converted to an integer " << "is "
        << static_cast<int>(dDouble) << "\n";
    cout << "\n\n";

    return 0;
}

//2) a.
#include <iostream>
using namespace std;
```

```
// Function to read array elements from the keyboard
void ReadArray(int a[], int size) {
	cout << "Enter " << size << " elements: " << endl;
	for (int i = 0; i < size; i++) {
		cout << "-> ";
		cin >> a[i];
	}
}

int main() {
	const int size = 8;    // Define the size of the array
	int Numbers[size];      // Declare the array

	// Read elements from the keyboard
	ReadArray(Numbers, size);

	int Minimum = Numbers[0];

	// Compare the members
	for (int i = 1; i < size; ++i) {
		if (Numbers[i] < Minimum)
```

```
            Minimum = Numbers[i];
        }

        // Announce the result
        cout << "The lowest member value of the array is " << Minimum << "." << endl;

        return 0;
}



//b.
#include <iostream>
using namespace std;

// Function to get the maximum value from the array
int getMax(int a[], int size) {
        int Maximum = a[0];
        for (int i = 1; i < size; ++i) {
                if (a[i] > Maximum)
                        Maximum = a[i];
        }
        return Maximum;
```

```
}

// Function to read elements into the array
void ReadArray(int a[], int size) {
	for (int i = 0; i < size; ++i) {
		cout << "Enter element " << i + 1 << ": ";
		cin >> a[i];
	}
}

int main() {
	const int size = 8;	// Define the size of the array
	int Numbers[size];	// Declare the array

	// Read elements from the keyboard
	ReadArray(Numbers, size);

	// Find the minimum value
	int Minimum = Numbers[0];
	for (int i = 1; i < size; ++i) {
		if (Numbers[i] < Minimum)
			Minimum = Numbers[i];
```

```
    }

    cout << "The lowest member value of the array is " << Minimum << "." << endl;

    // Find the maximum value and display it
    int Maximum = getMax(Numbers, size);
    cout << "The highest member value of the array is " << Maximum << "." <<
endl;

    return 0;
}

//c.
#include <iostream>
using namespace std;

// Function to swap two elements
void swap(int &x, int &y) {
    int t = x;
    x = y;
    y = t;
}
```

```
// Function to read elements into the array
void ReadArray(int a[], int size) {
	for (int i = 0; i < size; ++i) {
		cout << "Enter element " << i + 1 << ": ";
		cin >> a[i];
	}
}

// Function to sort the array in ascending order
void sortArray(int a[], int size) {
	for (int i = 0; i < size - 1; i++) {
		for (int j = i + 1; j < size; j++) {
			if (a[i] > a[j]) {
				swap(a[i], a[j]);
			}
		}
	}
}

int main() {
	const int size = 8;    // Define the size of the array
```

```
	int Numbers[size];	// Declare the array

	// Read elements from the keyboard
	ReadArray(Numbers, size);

	// Sort the array in ascending order
	sortArray(Numbers, size);

	// Display the sorted array
	cout << "The array in ascending order is: ";
	for (int i = 0; i < size; ++i) {
		cout << Numbers[i] << " ";
	}
	cout << endl;

	return 0;
}

//d.
#include <iostream>
using namespace std;
```

```
// Function to read array elements from the keyboard
void ReadArray(int a[], int size) {
	cout << "Enter " << size << " elements: " << endl;
	for (int i = 0; i < size; i++) {
		cout << "-> ";
		cin >> a[i];
	}
}

// Function to get the maximum value from the array
int getMax(int a[], int size) {
	int Maximum = a[0];
	for (int i = 1; i < size; ++i) {
		if (a[i] > Maximum)
			Maximum = a[i];
	}
	return Maximum;
}

// Function to swap two elements
void swap(int &x, int &y) {
	int t = x;
```

```
    x = y;
    y = t;
}

// Function to sort the array in ascending order
void sortArray(int a[], int size) {
    for (int i = 0; i < size - 1; i++) {
        for (int j = i + 1; j < size; j++) {
            if (a[i] > a[j]) {
                swap(a[i], a[j]);
            }
        }
    }
}

// Function to print the array
void printArray(int a[], int size) {
    cout << "The array contents are: ";
    for (int i = 0; i < size; i++) {
        cout << a[i] << " ";
    }
    cout << endl;
```

```
}

int main() {
	const int size = 8;    // Define the size of the array
	int Numbers[size];      // Declare the array

	// Read elements from the keyboard
	ReadArray(Numbers, size);

	// Find and display the minimum value
	int Minimum = Numbers[0];
	for (int i = 1; i < size; ++i) {
		if (Numbers[i] < Minimum)
			Minimum = Numbers[i];
	}
	cout << "The lowest member value of the array is " << Minimum << "." << endl;

	// Find and display the maximum value
	int Maximum = getMax(Numbers, size);
	cout << "The highest member value of the array is " << Maximum << "." <<
endl;
```

```
    // Sort the array
    sortArray(Numbers, size);

    // Display the sorted array
    printArray(Numbers, size);

    return 0;
}

//4.
#include <iostream>    // Use <iostream> instead of <iostream.h>

using namespace std;    // Required to avoid std:: prefix

enum TEmploymentStatus { esFullTime, esPartTime, esContractor, esNS };

int main()
{
    int EmplStatus;    // Added missing semicolon

    cout << "Employee's Contract Status: ";
    cout << "\n0 - Full Time | 1 - Part Time"
```

```
        << "\n2 - Contractor | 3 - Other"
        << "\nStatus: ";
cin >> EmplStatus;
cout << endl;

switch( EmplStatus )
{
case esFullTime:
    cout << "Employment Status: Full Time\n";
    cout << "Employee's Benefits: Medical Insurance\n"
            << " Sick Leave\n"
            << " Maternal Leave\n"
            << " Vacation Time\n"
            << " 401K\n";
    break;    // Fixed missing semicolon here

case esPartTime:    // Added colon to fix syntax
    cout << "Employment Status: Part Time\n";
    cout << "Employee's Benefits: Sick Leave\n"
            << " Maternal Leave\n";
    break;
```

```
        case esContractor:
            cout << "Employment Status: Contractor\n";
            cout << "Employee's Benefits: None\n";    // Fixed missing semicolon here
            break;

        case esNS:
            cout << "Employment Status: Other\n";
            cout << "Status Not Specified\n";
            break;

        default:
            cout << "Unknown Status\n";
        }

        return 0;
}

//5)
#include <iostream>
#include <limits.h>    // For INT_MAX and INT_MIN
using namespace std;
```

```
void odd(int a, int &oddSum, int &minOdd);
void even(int a, int &evenSum, int &maxEven);

int main()
{
	int i;
	int oddSum = 0, evenSum = 0;	// To store sums of odd and even numbers
	int minOdd = INT_MAX;	// Start with a large value for the minimum odd
	int maxEven = INT_MIN; // Start with a small value for the maximum even

	do
	{
		cout << "Type a number: (0 to exit): ";
		cin >> i;

		// If the number is not 0, process it as either odd or even
		if (i != 0)
		{
			odd(i, oddSum, minOdd);
			even(i, evenSum, maxEven);
		}
	}
```

```
    while(i != 0);

    // Display results
    cout << "\nSum of Odd Numbers: " << oddSum << endl;
    cout << "Sum of Even Numbers: " << evenSum << endl;

    if (minOdd != INT_MAX)    // If at least one odd number was entered
        cout << "Minimum Odd Number: " << minOdd << endl;
    else
        cout << "No Odd Numbers Entered\n";

    if (maxEven != INT_MIN)    // If at least one even number was entered
        cout << "Maximum Even Number: " << maxEven << endl;
    else
        cout << "No Even Numbers Entered\n";

    return 0;
}

void odd(int a, int &oddSum, int &minOdd)
{
    if ((a % 2) != 0)    // Check if the number is odd
```

```
    {
        cout << "Number is odd.\n";
        oddSum += a;    // Add to the odd sum

        if (a < minOdd)    // Update the minimum odd number
            minOdd = a;
    }
}

void even(int a, int &evenSum, int &maxEven)
{
    if ((a % 2) == 0)    // Check if the number is even
    {
        cout << "Number is even.\n";
        evenSum += a;    // Add to the even sum

        if (a > maxEven)    // Update the maximum even number
            maxEven = a;
    }
}

//6)
```

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
	vector<int> g1;

	for (int i = 1; i <= 5; i++)
		g1.push_back(i);

	cout << "Output of begin and end: ";
	for (auto i = g1.begin(); i != g1.end(); ++i)
		cout << *i << " ";

	cout << "\nOutput of cbegin and cend: ";
	for (auto i = g1.cbegin(); i != g1.cend(); ++i)
		cout << *i << " ";

	cout << "\nOutput of rbegin and rend: ";
```

```
    for (auto ir = g1.rbegin(); ir != g1.rend(); ++ir)
        cout << *ir << " ";

    cout << "\nOutput of crbegin and crend : ";
    for (auto ir = g1.crbegin(); ir != g1.crend(); ++ir)
        cout << *ir << " ";

    return 0;
}
```