Homework 1

- Read 2 integers A, B and print based on following cases:
  - if both are odd print their product A*B
  - if both are even print their division A/B
  - if the first is odd and the second is even then find their sum A+B
  - if the first is even and the second is odd then find their subtraction A-B
- Inputs ⇒ outputs
  - 5 7 => 35
  - 12 2 => 6
  - 5 6 => 11
  - 12 3 => 9

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

int main()
{
    int A, B;
    cout << "Enter two numbers: ";
    cin >> A >> B;

    if (A % 2 != 0 && B % 2 != 0)
    {
        cout << "Product: " << A * B << endl;
    }
  else if (A % 2 == 0 && B % 2 == 0)
    {
        cout << "Division: " << A / B << endl;
    }
  else if (A % 2 != 0 && B % 2 == 0)
    {
        cout << "Sum: " << A + B << endl;
    }
  else if (A % 2 == 0 && B % 2 != 0)
    {
        cout << "Subtraction: " << A - B << endl;
    }
 return 0;
}
```

Homework 2: Sort 3 numbers

- Given 3 integers, sort (order) them in ascending order and print them .
- Inputs
  - 1 2 3 ⇒ 1 2 3
  - 1 3 2 ⇒ 1 2 3
  - 2 1 3 ⇒ 1 2 3
  - 2 3 1 ⇒ 1 2 3
  - 3 1 2 ⇒ 1 2 3
  - 3 2 1 ⇒ 1 2 3
- Do you notice there are only 6 ways to permutate 3 numbers!

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

int main()
{
    int a, b, c;
    cout << "Enter three numbers: ";
    cin >> a >> b >> c;

    if (a > b) swap(a, b);
    if (a > c) swap(a, c);
    if (b > c) swap(b, c);

    cout << a << " " << b << " " << c << endl;

    return 0;
}
```

Homework 3: Maximum but constrained

- Given 3 integers, you have to find the biggest one of them which is < 100.
  - Print -1 if no such number
- Inputs
  - 22 90 115 ⇒ 90
    - Here [20 90] are only < 100. Maximum (20, 90) = 90
  - 200 300 400 ⇒ -1
    - All of them are > 100, so no answer
  - 50 100 150 ⇒ 50
    - Only 50 is < 100.
  - 10 30 20 ⇒ 30
    - The 3 numbers < 100, so their max is 30

```cpp
#include<iostream>

using namespace std;

int main()
{
    int a, b, c;
    cout << "Enter three numbers: ";
    cin >> a >> b >> c;

    int maxNum = -1;
    if (a < 100 && a > maxNum ) maxNum  = a;
    if (b < 100 && b > maxNum ) maxNum  = b;
    if (c < 100 && c > maxNum ) maxNum  = c;

    cout << maxNum << endl;

    return 0;
}
```

Homework 4: Conditional Count

- Write a program that reads number X, then other 5 numbers. Print 2 values:
  - How many numbers <= X
  - How many numbers > X
  - Any relation between these 2 outputs?
- Inputs
  - 10 **300 1 5 100 200**
  - Output: 2 3
  - Explantation
  - 2 numbers (1, 5) are <= 10
  - 3 numbers (100, 200, 300) are > 10

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

int main()
{
    int X, num, countLess = 0, countMore = 0;
    cout << "Enter X: ";
    cin >> X;

    for (int i = 0; i < 5; i++)
    {
        cout << "Enter number " << i + 1 << ": ";
        cin >> num;

        if (num <= X) countLess++;
        else countMore++;
    }

    cout << "Numbers <= X: " << countLess << endl;
    cout << "Numbers > X: " << countMore << endl;

    return 0;
}
```

Homework 5: Find Maximum of 10

- Read 10 integers, find which of them has the biggest value and print it.
- Inputs
  - 1 67 -9 88 -45 129 90 65 77 34 ⇒ 129
- Restriction: In your whole code there should be 2 integer variables defined ONLY
  - If hard constraint; code it in whatever easier way for you

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

int main()
{
    int maxNum, num;
    cout << "Enter 10 numbers: ";
    cin >> maxNum;

    for (int i = 1; i < 10; i++)
    {
        cin >> num;
        if (num > maxNum) maxNum = num;
    }

    cout << "Maximum value: " << maxNum << endl;

    return 0;
}
```

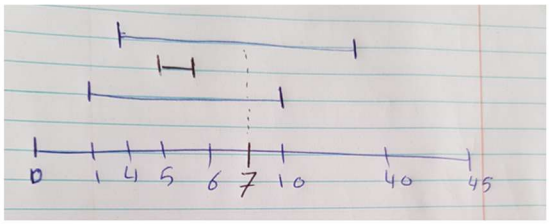
Homework 6: Find Maximum up to 10

- Read an integer N (2 <= N <= 10)
- Then read N integers, find which of them has the biggest value and print it.
- Inputs
  - 5 **1 3 2 4 2** ⇒ 4
    - 5 means read 5 integers
    - Then we read them [**1 3 2 4 2**]. Their maximum is 4
  - **10** 1 67 -9 88 -45 129 90 65 77 34 ⇒ 129
    - Same as last homework. This time we are given first N (10)
-

```cpp
#include <iostream>

using namespace std;

int main()
{
    int N, maxNum, num;
    cin >> N >> maxNum;

    for (int i = 1; i < N; i++)
    {
        cin >> num;
        if (num > maxNum) maxNum = num;
    }

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

Homework 7: Intervals

- Read number X then read 6 numbers s1, e1, s2, e2, s3, e3
  - These 6 numbers are for 3 interval
  - Each Interval is a range [start, end]
  - Number X in a range if start <= X <= end
  - E.g 7 in range [5, 12] but not in range [10, 20]
- Print how many digits X is part of it
- Inputs
  - 7 1 10 5 6 4 40 ⇒ 2
    - Number 7 exists in 2 intervals [1, 10] and [4, 40]
  - 10 5 15 6 100 3 30 ⇒ 3
    - 1o exists in the 3 intervals [5 15], [6 100], [3 30]
  - 10 100 200 100 101 120 170 ⇒ 0




```cpp
#include <iostream>

using namespace std;

int main() {
    int X, s1, e1, s2, e2, s3, e3, count = 0;
    cout << "Enter X followed by three intervals the start and the end for
each:\n";
    cin >> X >> s1 >> e1 >> s2 >> e2 >> s3 >> e3;

    if (s1 <= X && X <= e1) count++;
    if (s2 <= X && X <= e2) count++;
    if (s3 <= X && X <= e3) count++;

    cout << " X is in " << count << " intervals" << endl;
    return 0;
}
```

Homework 8: Two Intervals Intersection

- Read 4 numbers representing 2 intervals and print their intersection interval. If they don't intersect, print -1
- Inputs
  - 1 6 3 8 ⇒ 3 6
    - Interval [1 6] and [3 8] only intersects at [3, 6]
    - Why: interval [1, 6] has numbers: {1, 2, **3, 4, 5, 6**}
    - And: interval [3, 8] has numbers: {**3, 4, 5, 6**, 7, 8}
    - So the intersection is {**3, 4, 5, 6**} = [3, 6]
  - 1 15 20 30 ⇒ -1

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

int main()
{
    int s1, e1, s2, e2;

    cout << "Enter the start and end of the intervals:\n";
    cin >> s1 >> e1 >> s2 >> e2;

    int start = max(s1, s2);
    int end = min(e1, e2);

    if (start <= end)
    {
        cout << "The intervals overlap in [" << start << ", " << end << "]\n";
    }
    else
    {
        cout << " No overlap, output: -1\n";
    }

    return 0;
}
```