



## LABORATORY 3

March 21, 2021

### Abstract

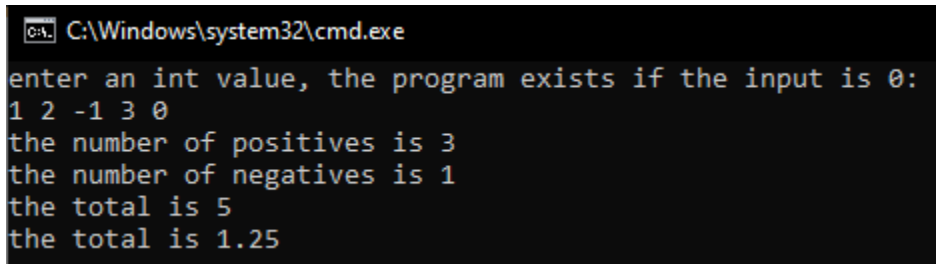
We learned to use for and while loops to execute math calculations and formatting styles

Jordan Xu

Ece 1310 C for Engineers

1. Write a program that reads an unspecified number of integers, determines how many positive and negative values have been read, and computes the total and average of the input values (not counting zeros). Your program ends with the input 0. Display the average as a floating-point number.

```
// ECE 1310-04
// Author: Jordan Xu
// Description: determine amount of positive and negative numbers, then find
// sum and average
// Date: 3/21/2021
#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;
const int sentinel = 0;
int main()
{
    int num, sum, pNum, nNum;
    sum = pNum = nNum = 0;
    cout << "enter an int value, the program exists if the input is 0: " << endl;
    cin >> num;
    while (num != sentinel)
    {
        num < 0 ? nNum++ : pNum++; // determines no. of + and - values
        sum += num;
        cin >> num;
    }
    cout << "The number of positives is " << pNum << endl;
    cout << "The number of negatives is " << nNum << endl;
    cout << "The total is " << sum << endl;
    cout << fixed << setprecision(2);
    if (sum == 0) // prevents error if only input is 0
        cout << "The average is " << 0.00 << endl;
    else
        cout << "The total is " << sum / double(pNum + nNum) << endl;
    cout << endl;
    return 0;
}
```

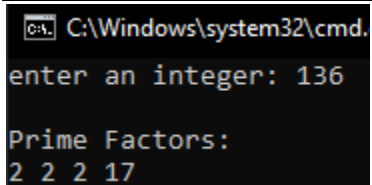


```
C:\Windows\system32\cmd.exe
enter an int value, the program exists if the input is 0:
1 2 -1 3 0
the number of positives is 3
the number of negatives is 1
the total is 5
the total is 1.25
```

2. Write a program that reads an integer and displays all its smallest factors, also known as prime factors.

```
// ECE 1310-04
// Author: Jordan Xu
// Description: integer broken down to prime factors
// Date: 3/21/2021
#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;
int main()
{
    int num, temp, prod, k;
    prod = 1, k = 2;
    cout << "enter an integer: ";
    cin >> num;
    temp = num;
    cout << endl;
    cout << "Prime Factors: " << endl;
    while (num != prod)
    {
        if (temp % k == 0)
        {
            cout << k << " ";
            prod *= k;
            temp /= k;
        }
        else k++;
    }
    cout << endl;

    return 0;
}
```



```
C:\Windows\system32\cmd.
enter an integer: 136

Prime Factors:
2 2 2 17
```

3. Use nested loops that print the following patterns in four separate programs:

```
// ECE 1310-04
// Author: Jordan Xu
// Description: integer broken down to prime factors
// Date: 3/21/2021
#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;
int main()
{
    cout << "3a" << endl;
    cout << endl;
    for (int a = 1; a <= 6; a++)
    {
        for (int b = 1; b <= a; b++)
            cout << b << " ";
        cout << endl;
    }
    cout << endl;
    cout << endl;

    cout << "3b" << endl;
    cout << endl;
    for (int a = 6; a >= 1; a--)
    {
        for (int b = 1; b <= a; b++)
            cout << b << " ";
        cout << endl;
    }
    cout << endl;
    cout << endl;

    cout << "3c" << endl;
    cout << endl;
    int w = 6; // formula won't work if w >= 10
    int width = w;
    for (int a = 1; a <= w; a++)
    {
        for (int b = 1; b <= 2 * (width - 1); b++) // adjusts spacing
            cout << " ";
        for (int b = a; b >= 1; b--)
            cout << b << " ";
        width--;
        cout << endl;
    }
    cout << endl;
```

```

cout << endl;

cout << "3d" << endl;
cout << endl;
int s = 6;
int space = 1;
for (int a = s; a >= 1; a--)
{
    for (int b = 1; b <= 2 * (space - 1); b++) // adjusts spacing
        cout << " ";
    for (int b = 1; b <= a; b++)
        cout << b << " ";
    width++;
    cout << endl;
}
cout << endl;
return 0;
}

```

```

C:\Windows\system32\cmd.exe
3a
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6

3b
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1

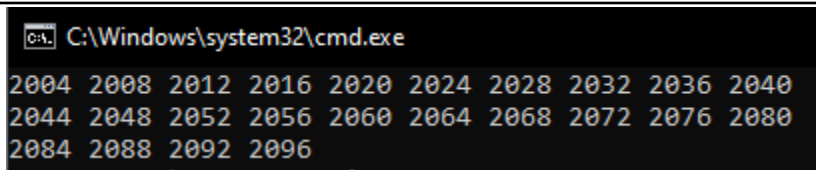
3c
      1
     2 1
    3 2 1
   4 3 2 1
  5 4 3 2 1
 6 5 4 3 2 1

3d
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1

```

4. Write a program that displays, ten per line, all the leap years in the twenty-first century (from year 2001 to 2100).

```
// ECE 1310-04
// Author: Jordan Xu
// Description: integer broken down to prime factors
// Date: 3/21/2021
#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;
int main()
{
    bool isLeap = false;
    int line = 0;
    for (int year = 2001; year <= 2100; year++)
    {
        isLeap = !(year % 4) ? (!(year % 100) ? (!(year % 400) ? true : false) : true) : false;
        if (isLeap)
        {
            line++;
            cout << year;
            if (line % 10 != 0)
                cout << " ";
            else cout << endl;
        }
    }
    cout << endl;
    return 0;
}
```



```
C:\Windows\system32\cmd.exe
2004 2008 2012 2016 2020 2024 2028 2032 2036 2040
2044 2048 2052 2056 2060 2064 2068 2072 2076 2080
2084 2088 2092 2096
```

5. Prove that the following summation is 24.

$$\frac{1}{1 + \sqrt{2}} + \frac{1}{\sqrt{2} + \sqrt{3}} + \frac{1}{\sqrt{3} + \sqrt{4}} + \cdots + \frac{1}{\sqrt{624} + \sqrt{625}}$$

```
// ECE 1310-04
// Author: Jordan Xu
// Description: integer broken down to prime factors
// Date: 3/21/2021
#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;
int main()
{
    double sum = 0;
    for (int i = 1; i <= 624; i++)
        sum += 1 / (pow(i, 0.5) + pow(i + 1, 0.5));
    cout << "sum = " << sum << endl;
    return 0;
}
```

```
C:\Windows\system32\cmd.exe
sum = 24
```