Homework 3

1 - SUMMING NUMBERS

1.1 - A Naive Approach

CODE:

```
#include <iostream>

using namespace std;

int main() {

    int total = 0;
    total = total + 1;  // total = 1
    total = total + 2;  // total = 3
    total = total + 3;  // total = 6
    total = total + 4;  // total = 10
    total = total + 5;  // total = 15
    total = total + 6;  // total = 16
    total = total + 7;  // total = 21
    total = total + 8;  // total = 28
    total = total + 9;  // total = 36
    total = total + 10; // total = 55

    cout << total << endl;

    return 0;
 }
```

The output of running this program is: 55

1.2 - Using A For-Loop

CODE:

```
#include <iostream>

using namespace std;

int main() {

    int total = 0;

    for(int i = 1; i <= 10; i++) {
        total = total +i;
    }

    cout << total << endl;

    return 0;
}
```

MODIFIED CODE:

```
#include <iostream>

using namespace std;

int main() {

    int total = 0;

    for(int i = 1; i <= 10; i++) {
        total = total +i;
        cout << "i = " << i << endl;
        cout << "total = " << total << endl;
    }

    cout << total << endl;

    return 0;
}
```

Terminal Output:

```
    i = 1
    total = 1
    i = 2
    total = 3
    i = 3
    total = 6
    i = 4
    total = 10
    i = 5
    total = 15
    i = 6
    total = 21
    i = 7
    total = 28
    i = 8
    total = 36
    i = 9
    total = 45
    i = 10
    total = 55
    55
```

The output of "MODIFIED CODE", as seen above, shows the value of the internal variables 'i' and 'total' for each iteration in the `For Loop`.

At the start of the `For Loop` (first iteration), the value of 'total' is 0, and the value of 'i' starts at 1. Then, the value of 'i' is added to 'total', and then both of the variables' values are printed to the screen.

Once the first iteration ends, the second begins. At the start of each iteration, the value of 'i' is checked to make sure it is less than or equal to 10, and if it is, then the value of 'i' is incremented by 1. Now, the value of 'i' is 2. In the next line, the value of 'i' is added to the value of 'total'. Since 'i' is 1 and total is '2', the value of 'total' is now 3. Both 'i' and 'total' are printed to the screen, the iteration ends, and the next one begins. 'i' is checked to see if it is less than 10 (which it is), and then it gets incremented by 1. The value of 'i' is now 3. Then, the value of 'i' gets added to 'total'. Since 'i' is 3 and 'total' is also 3, the new value of 'total' is 6. This process repeats until the value of 'i' hits 11. Once the value of 'i' is 11, the for-loop’s conditions are not satisfied, and the program exits the loop and continues with the next line of code which says to print the value of 'total' to the screen.

1.3 - User Defined Sum

CODE:

```
#include <iostream>

using namespace std;

int main() {

    int total = 0, n;

    cout << "What Number Do You Want To Sum To?\n" << "> ";
    cin >> n;

    //cout << n;

    for(int i = 1; i <= n; i++) {
        total = total +i;
        //cout << "i = " << i << endl;
        //cout << "total = " << total << endl;
    }

    cout << "The value of 'n' is --> " << n << endl;
    cout << "The final sum is    --> " << total << endl;

    return 0;
}
```

The final output of this program, when 'n' = 101, is:

```
  The value of 'n' is --> 101
  The final sum is    --> 5151
```

2) IF STATEMENTS

```
#include <iostream>

using namespace std;

int main() {

    int user_input;

    cout << "Please enter an Integer --> ";
    cin >> user_input;

    if (user_input % 2 == 0) cout << "True" << endl;
    else                     cout << "False" << endl;

    return 0;
}
```

3) PRACTICE

```
#include <iostream>

using namespace std;

int main() {

    int user_input;
    bool isPrime = true;

    cout << "Please Enter An Integer :: ";
    cin >> user_input;

    if (user_input <= 1) isPrime = false;

    for (int strt = 2; (strt < user_input); strt++)
        if (user_input % strt == 0) isPrime = false;

    if (isPrime) cout << "True" << endl;
    else         cout << "False" << endl;

    return 0;
}
```

The output for the program above, on the numbers 1 to 10 is:

```
$ ./Prime
Please Enter An Integer :: 1
False

$ ./Prime
Please Enter An Integer :: 2
True

$ ./Prime
Please Enter An Integer :: 3
True

$ ./Prime
Please Enter An Integer :: 4
False

$ ./Prime
Please Enter An Integer :: 5
True

$ ./Prime
Please Enter An Integer :: 6
False

$ ./Prime
Please Enter An Integer :: 7
True

$ ./Prime
Please Enter An Integer :: 8
False

$ ./Prime
Please Enter An Integer :: 9
False

$ ./Prime
Please Enter An Integer :: 10
False
```