

## Programmers Notes and Homework #1, 11/1/06

**From:** Buis, Paul E.

**Subject:** Robotics programming homework

### Homework #1

Download the devC++ programming environment. The site (<http://www.bloodshed.net/devcpp.html>) is down as I write this. You might also try [http://www.download.com/Bloodshed-Dev-C-/3000-2069\\_4-10019857.html](http://www.download.com/Bloodshed-Dev-C-/3000-2069_4-10019857.html)

1. Create a C (the default is C++) "console" project.
2. Write a program to add the integers from 1 to 100 inclusive.
3. Extra credit: read in an integer, n, with the scanf function and add the integers from 1 to n inclusive

Example C program from 11/1/2006 tutorial session (guess the number between 0 and 999:

```
#include <stdio.h>

#include <stdlib.h>

int main(void)
{
    srand();

    int answer = 1000*((float)rand()/(2<<15));

    int guess;
    int i;
    for (i=0; i<10; i++) {
        printf("new guess: ");
        scanf("%d", &guess);
        if (guess == answer) {
            printf("correct!\n");
            break;
        }
    }
```

```

    else if (guess < answer) {
        printf("too low\n");
    }
    else {
        printf("too high\n");
    }
}

if (i==10)
    printf("too many guesses, answer was %d\n", answer);

system("PAUSE");

return 0;
}

```

---

### Homework #1 Solution:

The answer for the sum of integers from 1 to 100 is 5050. The key to doing it in you head is to pair up numbers starting at the beginning of the sequence with numbers from the end of the sequence: (1,100), (2,99), (3,98), ... and then ask yourself how many pairs there will be. With 50 pairs of numbers, and each pair totaling to 101, the answer is  $50 \cdot 101 = 5050$ . For the sum of integers from 1 to  $n$ , the answer will always be  $n/2 \cdot (n+1)$ . One might ask, what if  $n$  is odd? Then you get  $(n-1)/2$  pairs and the value  $n/2$  as the odd value to add in at the end. For  $n=5$ , for example, you get (1,5), (2,4), 3 . So, the answer will be  $(n-1)/2 \cdot (n+1) + n/2$  and when you simplify that mess, you end up with  $n \cdot (n+1)/2$  which like the previous formula can be written  $(n^2+n)/2$ .

I incorrectly stated the kid who got punished by being told to add the first 100 integers was Newton. It was Gauss. <http://www.daviddarling.info/encyclopedia/G/Gauss.html>  
<http://www.americanscientist.org/template/AssetDetail/assetid/50686?&print=yes>

Here is the solution for the homework with a few lines of extra code to time how long the loop takes. Note that for large input values the answer overflows (the correct answer requires more than 32-bits). On my machine, even for an input value of a million, the clock doesn't usually tick during the loop. For an input value of a billion, it takes about a half second. The robot microcontroller differs from a PC in that it will be much slower and an "int" is only 16-bits so overflow is a much more likely problem.

The problem I've seen with code you guys have sent me is variable name choice. In general, you should choose names that are meaningful rather than i, j, and k.

```

#include <stdio.h>

#include <time.h>

```

```

int main(void)
{
    int i;           // loop index
    int sum = 0;     // sum of loop indices
    int n;           // max loop index
    clock_t tBefore; // clock_t defined as appropriate unsigned
integer for CPU
    clock_t tAfter;

    printf ("Please enter a number ");
    scanf("%d", &n);
    tBefore = clock();

    for (i=1; i<=n; i++) {
        sum = sum + i;
    }

    tAfter = clock();
    printf ("The answer is %d\n", sum);

    double ticks = tAfter - tBefore;           // how many times the
clock has ticked
    printf("Elapsed Time: %f\n", ticks/CLOCKS_PER_SEC);

    system ("pause");
    return 0;
}

```