C++ Programming

While Loops Practice

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Practice: Numbers divisible by 3

```
4  int main() {
5      int end;
6      cin >> end;
7
8      int start = 1;
9
10     while (start <= end) {
11         if (start % 3 == 0)
12             cout << start << "\n";
13         start += 1;
14     }
15     return 0;
16 }
17
```

Problems | Console | Tasks | Propert

<terminated> ztemp [C/C++ Application] /home/m

```
12
3
6
9
12
```

- Read an integer X, find all numbers divisible by 3 from 1 to X.
  - These are 3, 6, 9, 12, 15, 18, ….. (multiple of 3)

Practice: Power Function

```
4  int main() {
5      int num, pow;
6      cin >> num >> pow;
7
8      int result = 1;
9
10     while (pow >= 1) {
11         result *= num;
12         pow--;
13     }
14     cout<<result;
15     return 0;
16 }
17
```

Problems | Console | Tasks

<terminated> ztemp [C/C++ Application]

```
2 5
32
```

- Read 2 integers X and Y and compute X^Y.
  - This means X * X * X ….. Y times
  - E.g = 2^5= 2 * 2 * 2 * 2 * 2

Practice: Number of digits

```
4  int main() {
5      int num;
6      cin >> num;
7
8      int digits = 0;
9
10     while (num > 0) {
11         digits += 1;
12         num = num / 10;
13     }
14     cout<<digits;
15     return 0;
16 }
17
```

Problems | Console | Tasks

<terminated> ztemp [C/C++ Application]

```
123
3
```

- Read a C++ integer and count its number of digits
- There are 2 bugs in this code
  - Find 2 test cases to find them!

Practice: Number of digits - Fixing bugs!

```
3
4  int main() {
5      int num;
6      cin >> num;
7
8      int digits = 0;
9
10     if (num == 0)
11         digits = 1;
12     else {
13         while (num > 0) {
14             digits += 1;
15             num = num / 10;
16         }
17     }
18     cout << digits;
19     return 0;
20 }
21
```

Problems | Console | Tasks

<terminated> ztemp [C/C++ Application] /h

```
0
1
```

- Our first bug is: the previous code fails for input 0
  - The loop won't be accessed as num > 0 condition
- Solution:
  - Special if condition for this special case
- Other bug?

Practice: Number of digits - Fixing bugs!

```
3
4  int main() {
5      int num;
6      cin >> num;
7
8      int digits = 0;
9
10     if (num == 0)
11         digits = 1;
12     else {
13         if (num < 0)
14             num = -num;
15
16         while (num > 0) {
17             digits += 1;
18             num = num / 10;
19         }
20     }
21     cout << digits;
22     return 0;
23 }
24
```

Problems | Console | Tasks

<terminated> ztemp [C/C++ Application] /h

```
-1234
4
```

- Our previous code will fail for negative numbers
- E.g. if we feed -123, the condition fails!
- Simple trick: if it is negative, multiply by -1
- Works well!
- Except a single tricky case. What is it?

Practice: Number of digits - Fixing bugs!

```
4  int main() {
5      int num;
6      cin >> num;
7
8      int digits = 0;
9
10     if (num == 0)
11         digits = 1;
12     else if (num == -2147483648)
13         digits = 10;
14     else {
15         if (num < 0)
16             num = -num;
17
18         while (num > 0) {
19             digits += 1;
20             num = num / 10;
21         }
22     }
23     cout << "# of digits of "<<num<<" is "<<digits;
24     return 0;
25 }
26
```

- How could -num be wrong?
- We said last time integer limits are:
  - -2147483648 to 2147483647
- If we did -num on the max number it become: 2147483648
  - But this is > 2147483647 ⇒ Overflow
  - Solution: Special case handling
- We wanted to change output message as line 23. What is wrong?

Practice: Number of digits

```
int main() {
    int num;
    cin >> num;

    int tem_num = num;
    int digits = 0;

    if (num == 0)
        digits = 1;
    else if (num == -2147483648)
        digits = 10;
    else {
        if (num < 0)
            num = -num;

        while (num > 0) {
            digits += 1;
            num = num / 10;
        }
    }
    cout << "# of digits of "<<tem_num<<" is "<<digits;
    return 0;
}
```

Problems | Console | Tasks | Properties | Call Graph

<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces/

```
-54321
# of digits of -54321 is 5
```

- Num, the input, was divided till be zero. So we lost its **original** value!
- Solution: Make a copy
- **Lesson**
  - It takes time to be a strong programmer
    - Clean readable code
    - Short code
    - Well tested code

Nested loop

```
5     int T;
6     cin >> T;
7
8     while (T > 0) {
9         int num;
10        cin >> num;
11
12        int sum = 0;
13        int start = 1;
14
15        while (start <= num) {
16            sum += start;
17            start++;
18        }
19        --T;
20        cout << "Sum from 1 to " << num << " = " << sum << "\n";
21    }
22
23    return 0;
24 }
```

Problems | Console | Tasks | Properties | Call Graph | Search

```
<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces/eclipse_cpp
3
3
Sum from 1 to 3 = 6
4
Sum from 1 to 4 = 10
5
Sum from 1 to 5 = 15
```

- Write a program that reads integer T for T test cases.
- Then read T numbers: for each number N **print sum of 1 to N**
- Remember, we can replace the sum with formula N * (N+1) / 2
  - Which is more efficient?

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."