



Flow of Control

Week 2

2017 Fall

Problem 1

My little calculator

My little calculator

- Put the formula to calculate
 - Formula format: [operand1] [operator] [operand2]
 - Operand type: integer
 - Operator: + - * / %
- Print the result of input formula
 - Use switch statement rather than if-else
 - Print “OP” instead for unexpected operands
 - Print “D0” instead for ‘Divide by 0’ condition
- Repeat the above for the result is non-zero
 - Terminate if the result is zero

My little calculator

> 1+2

3

5-3

2

3*2

6

7/2

3

7%2

1

3/0

D0

3%0

D0

3#3

0P

1*0

0

My little calculator

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(){
6     char op;
7     int a,b;
8
9     int res = 1;
10
11    for(;;){
12        cin >> a;
13        cin >> op;
14        cin >> b;
15
16        switch (op) {
17            case '+':
18                res = a+b;
19                break;
20            case '-':
21                res = a-b;
22                break;
23            case '*':
24                res = a*b;
25                break;
```

6-7: use appropriate datatype for 'cin'

9: declare variable to store the result
(Initialization is to prevent unexpected behavior, because there are branches that do not assign values to res below.)

11: use 'for(;;)' to start infinite loop
using 'for(; res;)' is also nice

12-14: read inputs with 'cin'
(Using 'cin >> a >> op >> b;' is not recommended, because blanks between inputs are needed, and the readability goes poor.)

16-25: use switch-case to store the appropriate result according to the operator

My little calculator

```
26     case '/':
27         if(!b){
28             cout << "D0" << endl;
29             continue;
30         }
31         res = a/b;
32         break;
33     case '%':
34         if(!b){
35             cout << "D0" << endl;
36             continue;
37         }
38         res = a%b;
39         break;
40     default:
41         cout << "OP" << endl;
42         continue;
43 }
44
45 cout << res << endl;
46 if(!res)
47     break;
48 }
49 }
50 }
```

26-39: (continued) use 'switch-case' to store the appropriate result according to the operator

27-30, 34-37: handle 'Divide by 0' condition.

40-42: handle unexpected operator

36, 39, 42: use 'continue' to skip codes below and get back to the front of the loop.

45: print the result.

46-47: get out from the loop to terminate the program if the result is zero (unnecessary if 'for(; res;)' is used at line 11)

Problem 2

Patterns of stars

Patterns of stars

- Put an integer, N
- Print 3 patterns of stars as follows
 - Triangle (▲)
 - Diagonal (\)
 - Cross (X)
- The sizes of the patterns are same as N
- Right-side blanks are arbitrary

Patterns of stars

> 3

```
*  
**  
***
```

```
*  
 *  
  *
```

```
* *  
 *  
* *
```

> 4

```
*  
**  
***  
****
```

```
*  
 *  
  *  
   *
```

```
* *  
 **  
 **  
* *
```

Patterns of stars

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(){
6     int N;
7
8     cin >> N;
9
10    for(int i=1; i<=N; i++){
11        for(int j=0; j<i; j++){
12            cout << '*';
13        }
14        cout << endl;
15    }
16    cout << endl;
17
```

8: read input with 'cin'

10: iterate printing i^{th} line. ($i=1\cdots N$)

11-14: print stars of i as the i^{th} line

Patterns of stars

```
18     for(int i=0; i<N; i++){
19         for(int j=0; j<i; j++){
20             cout << ' ';
21         }
22         cout << '*' << endl;
23     }
24     cout << endl;
25
26     for(int i=0; i<N; i++){
27         for(int j=0; j<N; j++){
28             if(j==i || j==N-i-1)
29                 cout << '*';
30             else
31                 cout << ' ';
32         }
33         cout << endl;
34     }
35     cout << endl;
36 }
37
```

18: iterate printing i^{th} line. ($i=0 \cdots N-1$)

19-22: print spaces of $i-1$ and a star as the i^{th} line

26-27: iterate printing (i, j) position

28-33: print star if the position is at diagonal($i==j$), or space otherwise

※ 'Diagonal' is much easier to print using 'if'.