COMP2004 Programming Practice 2002 Summer School

Kevin Pulo
School of Information Technologies
University of Sydney

Assignment 1

- Example input/output
- How to find roman numerals
- Worked examples
- Different symbols
- Different number of symbols

Example Input/Output

Input	Output
IVXLCDM	XXXIV
34	LXXXVII
87	MMCCCLXIV
2364	CCXCIX
299	MMMMCCXCIX
4299	XLIV
44	

- Numbers will always fit into an unsigned long

Example main()

```
int main() {
   std::string symbols;
   unsigned long number;
   std::cin >> symbols;
   while (std::cin >> number) {
      std::cout << roman(number,
                  symbols) << std::endl;
   }
}
```

Symbol values

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

First digit

Number	Roman	(IVXLCDM)
1	I	
2	II	
3	III	
4	IV	
5	V	
6	VI	
7	VII	
8	VIII	
9	IX	

Second digit

N	R	N	R (IVXLCDM)
1	I	10	X
2	II	20	XX
3	III	30	XXX
4	IV	40	XL
5	V	50	L
6	VI	60	LX
7	VII	70	LXX
8	VIII	80	LXXX
9	IX	90	XC

Third digit

N	R	N	R (IVXLCDM)
1	I	100	C
2	II	200	CC
3	III	300	CCC
4	IV	400	CD
5	V	500	D
6	VI	600	DC
7	VII	700	DCC
8	VIII	800	DCCC
9	IX	900	CM

Last digit

N	R	N	R (IVXLCDM)
1	I	1000	M
2	II	2000	MM
3	III	3000	MMM
4	IV	4000	M?
5	V		
6	VI		
7	VII		
8	VIII		
9	IX		

Last digit

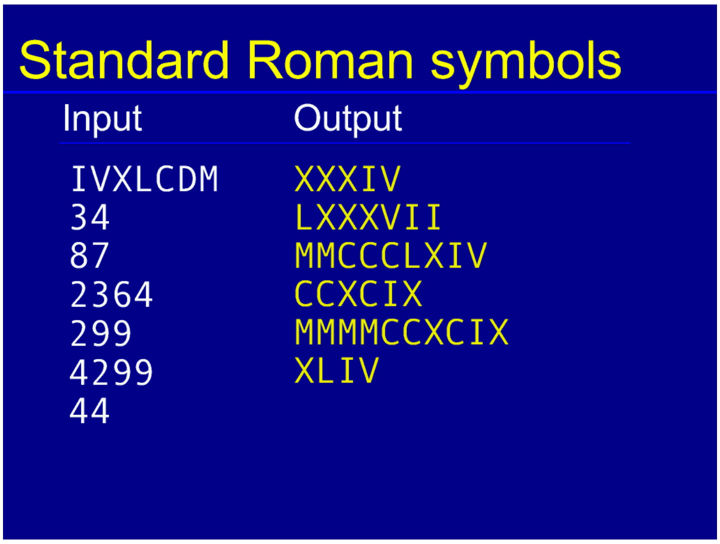
N	R	N	R (IVXLCDM)
1	I	1000	M
2	II	2000	MM
3	III	3000	MMM
4	IV	4000	MMMM
5	V	5000	MMMMM
6	VI	6000	MMMMMM
7	VII	7000	MMMMMMM
8	VIII	8000	MMMMMMMM
9	IX	9000	MMMMMMMMM

Worked Example 1

- Input: 2364
- Digits: 2,3,6,4
- Components: 2000,300,60,4
  - 2000: MM
  - 300: XXX
  - 60: LX
  - 4: IV
- Concatenating: MM XXX LX IV
- Without spaces: MMXXXLXIV

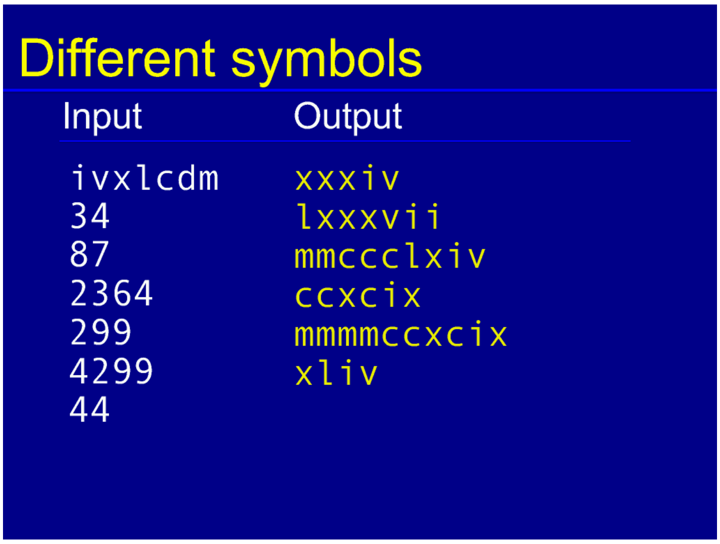
Worked Example 2

- Input: 999
- Digits: 9,9,9
- Components: 900,90,9
  - 900: CM
  - 90: XC
  - 9: IX
- Concatenating: CM XC IX
- Without spaces: CMXCIX



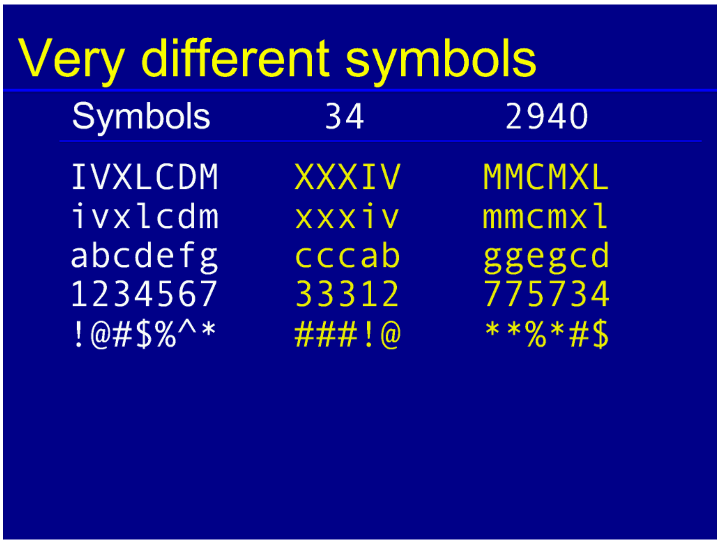
Standard Roman symbols

Input	Output
IVXLCDM	XXXIV
34	LXXXVII
87	MMCCCLXIV
2364	CCXCIX
299	MMMMCCXCIX
4299	XLIV
44	



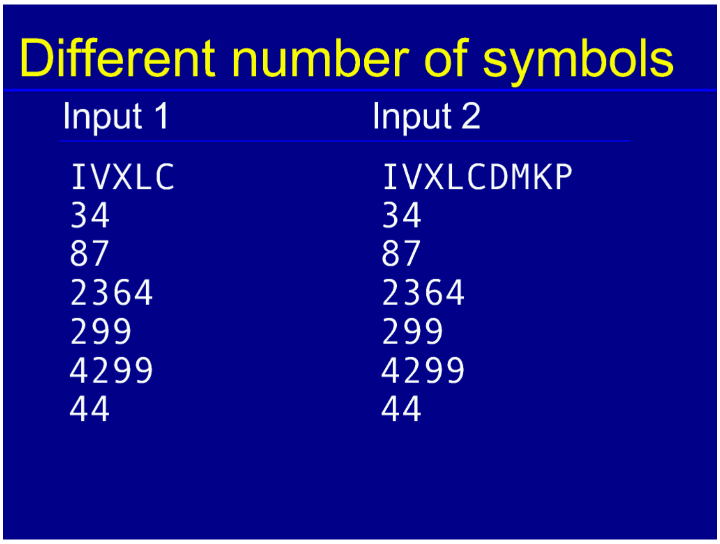
Different symbols

Input	Output
ivxlcdm	xxxiv
34	lxxxvii
87	mmccclxiv
2364	ccxcix
299	mmmmccxcix
4299	xliv
44	



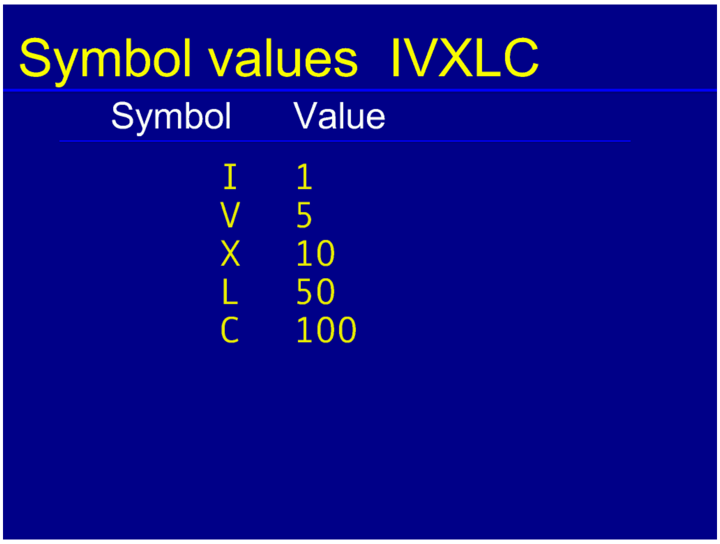
Very different symbols

Symbols	34	2940
IVXLCDM	XXXIV	MMCMXL
ivxlcdm	xxxiv	mmcmxl
abcdefg	cccab	ggegcd
1234567	33312	775734
!@#$%^*	###!@	**%*#$



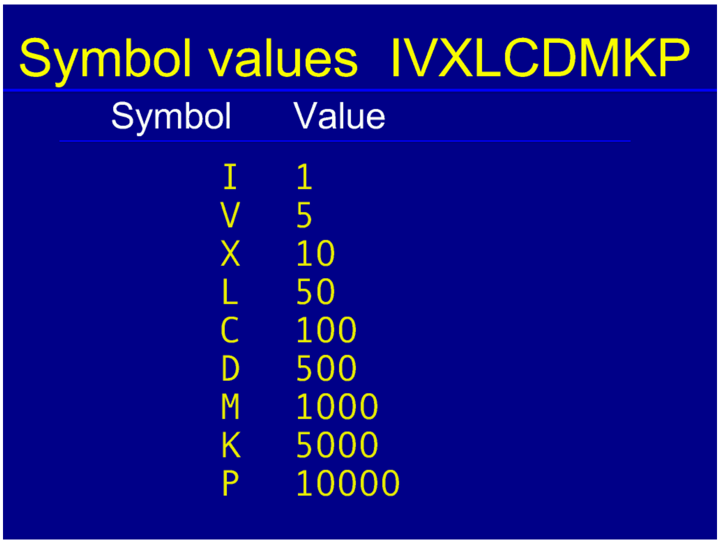
Different number of symbols

Input 1	Input 2
IVXLC	IVXLCDMKP
34	34
87	87
2364	2364
299	299
4299	4299
44	44



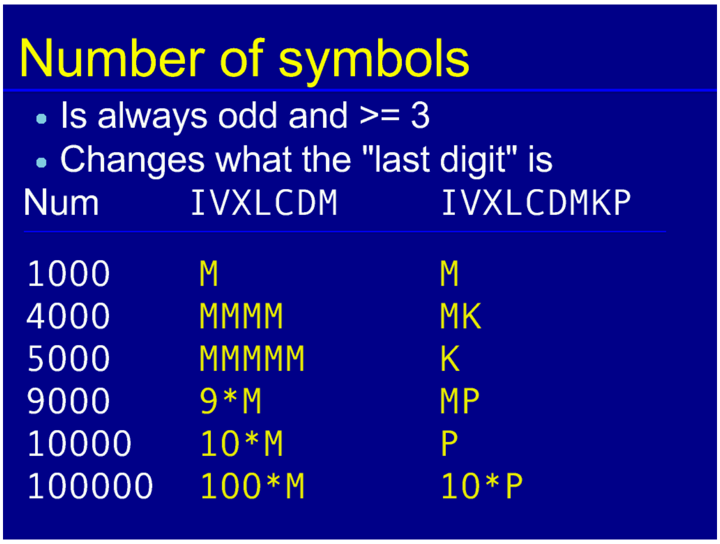
Symbol values IVXLC

Symbol	Value
I	1
V	5
X	10
L	50
C	100



Symbol values IVXLCDMKP

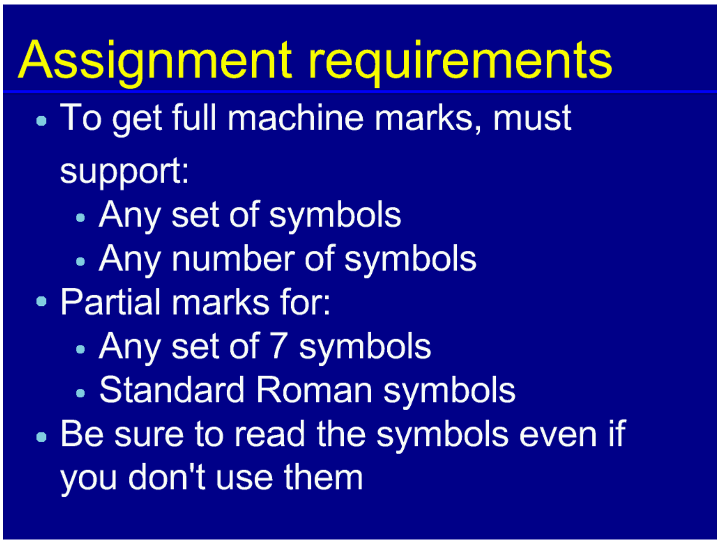
Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000
K	5000
P	10000



Number of symbols

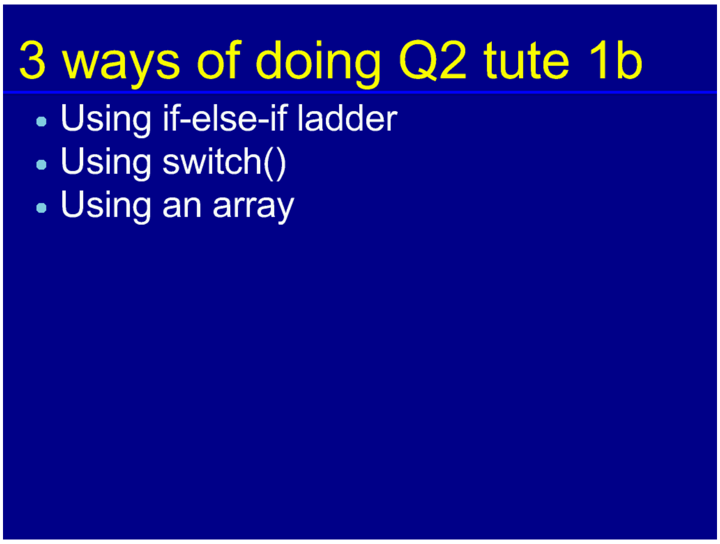
- Is always odd and >= 3
- Changes what the "last digit" is

Num	IVXLCDM	IVXLCDMKP
1000	M	M
4000	MMMM	MK
5000	MMMMM	K
9000	9*M	MP
10000	10*M	P
100000	100*M	10*P



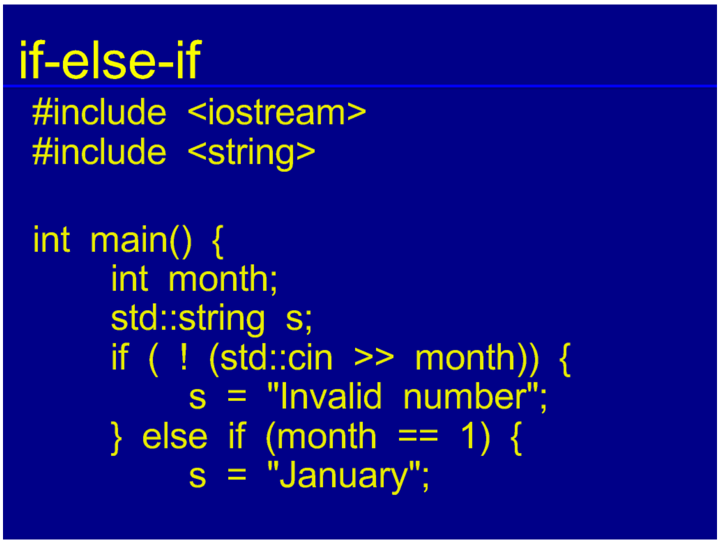
Assignment requirements

- To get full machine marks, must support:
  - Any set of symbols
  - Any number of symbols
- Partial marks for:
  - Any set of 7 symbols
  - Standard Roman symbols
- Be sure to read the symbols even if you don't use them



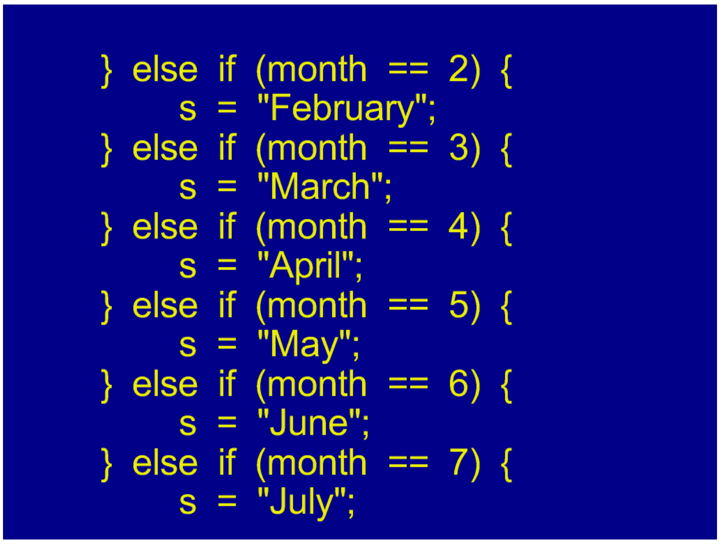
3 ways of doing Q2 tute 1b

- Using if-else-if ladder
- Using switch()
- Using an array

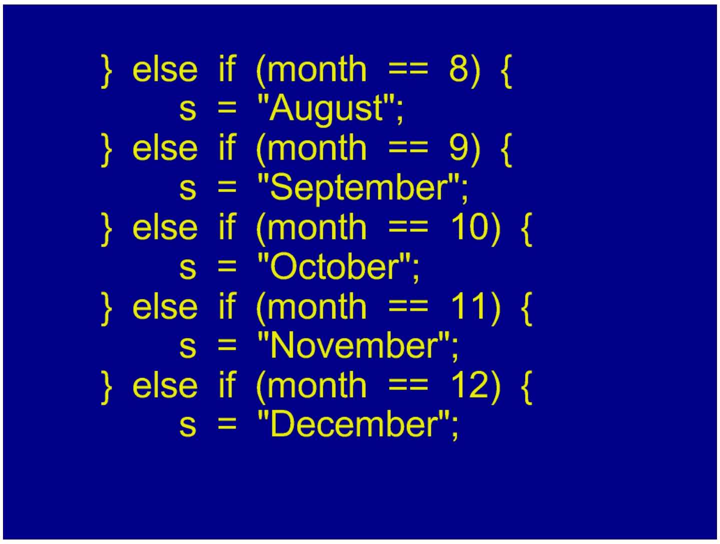


if-else-if

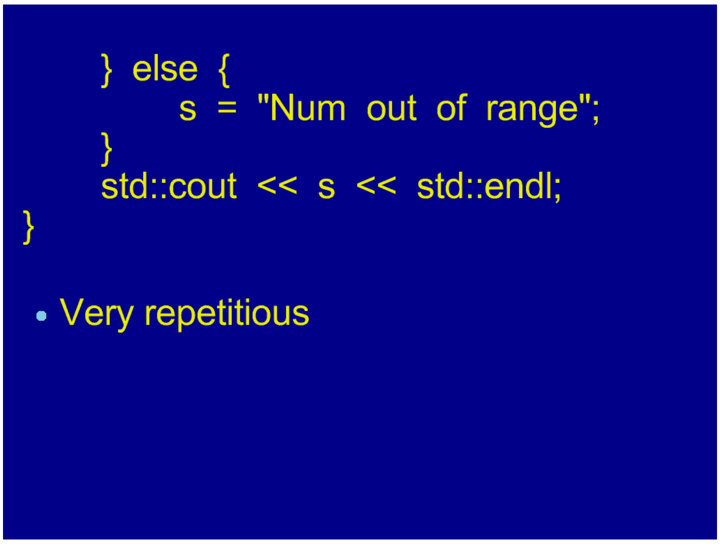
```
#include <iostream>
#include <string>

int main() {
    int month;
    std::string s;
    if ( ! (std::cin >> month)) {
        s = "Invalid number";
    } else if (month == 1) {
        s = "January";
```



```
    } else if (month == 2) {
        s = "February";
    } else if (month == 3) {
        s = "March";
    } else if (month == 4) {
        s = "April";
    } else if (month == 5) {
        s = "May";
    } else if (month == 6) {
        s = "June";
    } else if (month == 7) {
        s = "July";
```

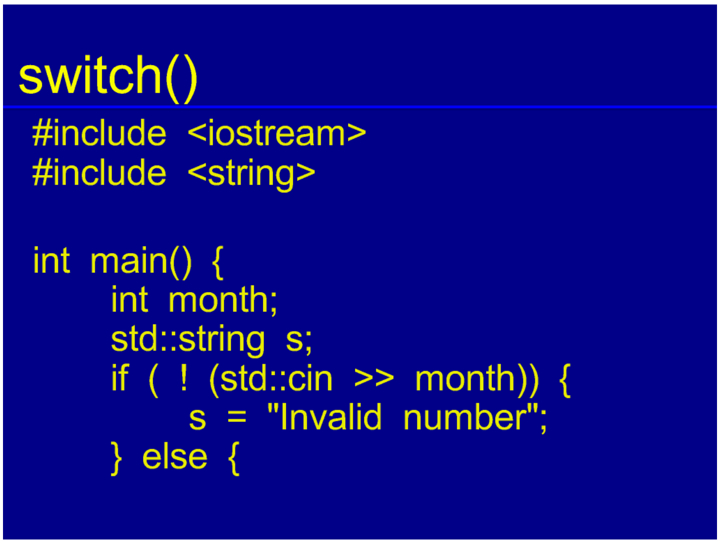


```
    } else if (month == 8) {
        s = "August";
    } else if (month == 9) {
        s = "September";
    } else if (month == 10) {
        s = "October";
    } else if (month == 11) {
        s = "November";
    } else if (month == 12) {
        s = "December";
```



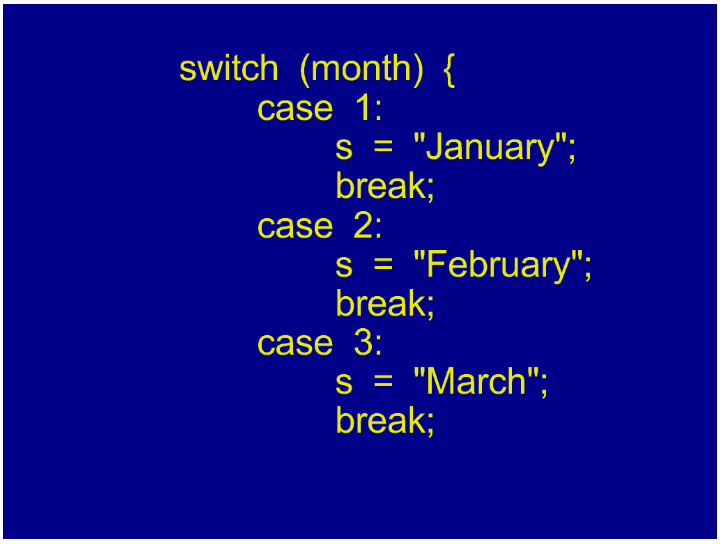
```
        } else {
                s = "Num out of range";
        }
        std::cout << s << std::endl;
}
```

- Very repetitious

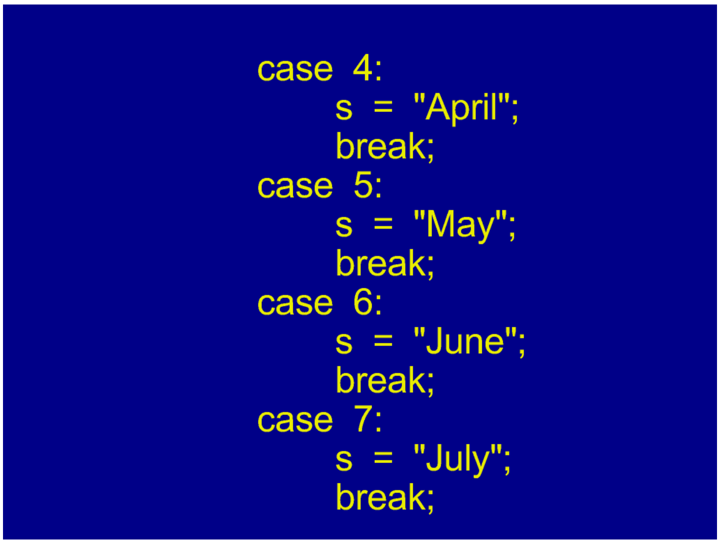


switch()

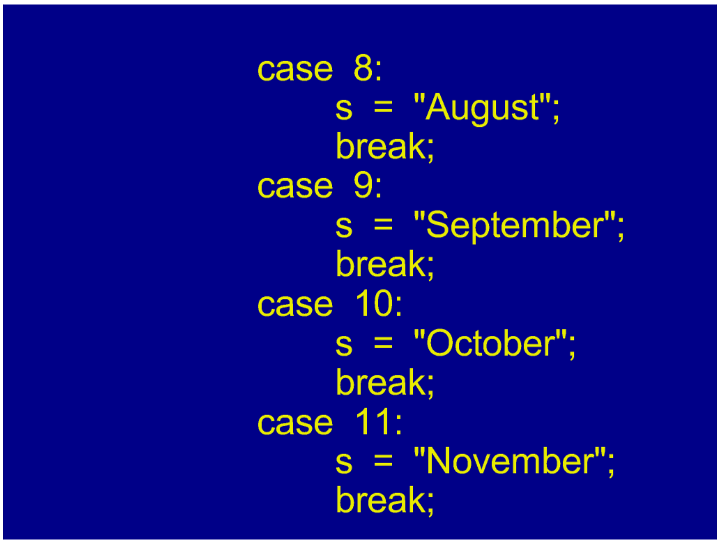
```
#include <iostream>
#include <string>

int main() {
        int month;
        std::string s;
        if ( ! (std::cin >> month)) {
                s = "Invalid number";
        } else {
```



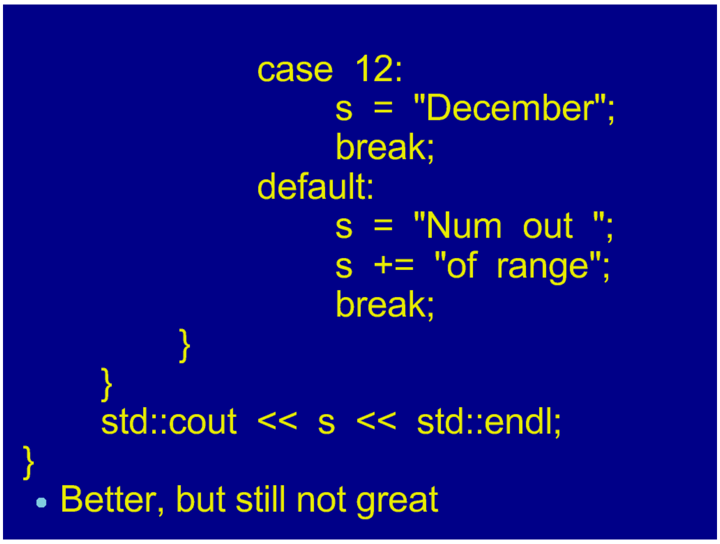
```
                switch (month) {
                        case 1:
                                s = "January";
                                break;
                        case 2:
                                s = "February";
                                break;
                        case 3:
                                s = "March";
                                break;
```



```
                        case 4:
                                s = "April";
                                break;
                        case 5:
                                s = "May";
                                break;
                        case 6:
                                s = "June";
                                break;
                        case 7:
                                s = "July";
                                break;
```

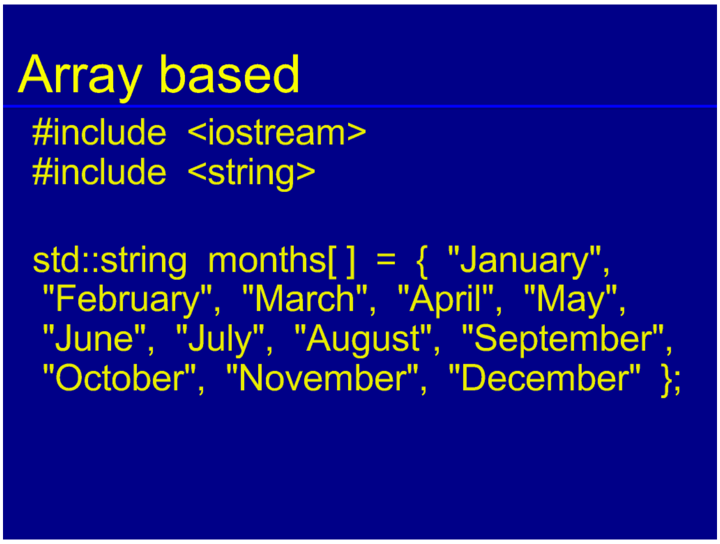


```
                        case 8:
                                s = "August";
                                break;
                        case 9:
                                s = "September";
                                break;
                        case 10:
                                s = "October";
                                break;
                        case 11:
                                s = "November";
                                break;
```



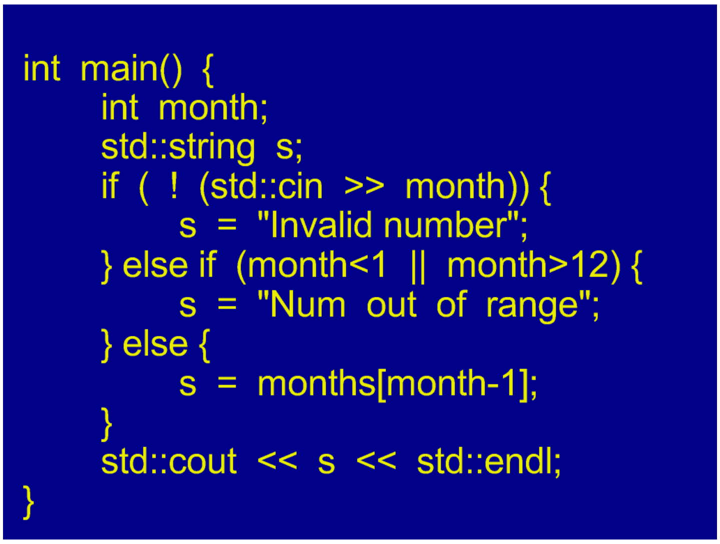
```
                        case 12:
                                s = "December";
                                break;
                        default:
                                s = "Num out ";
                                s += "of range";
                                break;
                }
        }
        std::cout << s << std::endl;
}
```

- Better, but still not great



Array based

```
#include  <iostream>
#include  <string>

std::string  months[ ]  =  {  "January",
 "February",  "March",  "April",  "May",
 "June",  "July",  "August",  "September",
 "October",  "November",  "December"  };
```



```
int  main()  {
        int  month;
        std::string  s;
        if  (  !  (std::cin  >>  month)) {
                s  =  "Invalid number";
        } else if  (month<1  ||  month>12) {
                s  =  "Num  out  of  range";
        } else {
                s  =  months[month-1];
        }
        std::cout  <<  s  <<  std::endl;
}
```