

HW8 Part 3 coding

108072147 林汶螢

1.

```
G++ 1.cpp > Number > ty
1  #include <iostream>
2  #include <string.h>
3  using namespace std;
4  class Number{
5  private:
6      int number;
7      string lessThan20[20] = {
8          "zero", "one", "two", "three", "four", "five", "six", "seven",
9          "eight", "nine", "ten", "eleven", "twelve", "thirteen", "fourteen",
10         "fifteen", "sixteen", "seventeen", "eighteen", "nineteen" };
11     string ty[10] = {"", "",
12         "twenty", "thirty", "forty", "fifty",
13         "sixty", "seventy", "eighty", "ninety" };
14     string hundred = "hundred";
15     string thousand = "thousand";
16 public:
17     Number(int num){
18         this->number = num;
19     }
20     string translate(int num){
21         string rtn;
22         int S = num%10, T = (num%100 - S)/10;
23         int Th = (num%10000 - num%1000)/1000, H = (num%1000 - T)/100;
24         if( Th > 0){
25             rtn += this->lessThan20[Th];
26             rtn += " ";
27             rtn += this->thousand;
28             rtn += " ";
29         }
30         if( H > 0){
31             rtn += this->lessThan20[H];
32             rtn += " ";
33             rtn += this->hundred;
34             rtn += " ";
35         }
36         if( T > 0){
37             if( T >= 2){
38                 rtn += this->ty[T];
39                 rtn += " ";
40             }
41             else {
42                 rtn += this->lessThan20[num%100];
43                 rtn += " ";
44                 return rtn;
45             }
46         }
47         if(S > 0){
48             rtn += lessThan20[S];
49             rtn += " ";
50         }
51         return rtn;
52     }
53 };
54
55 int main(){
56     int num = -1;
57     while(num<0||num>9999){
58         cout<<"Please enter a number. (0~9999)  ";
59         cin>>num;
60         cout<<endl;
61     }
62     Number number(num);
63     cout<<number.translate(num)<<endl;
64     return 0;
65 }
```

Output

```
Please enter a number. (0~9999)  1853
one thousand eight hundred fifty three
```

2.

```
G++ 2.cpp > main(int, char * [])
1  #include <iostream>
2  #include <string>
3  #include <sstream>
4  using namespace std;
5
6  class DayOfYear{
7  private:
8      int day;
9      const string month[12]={"January", "February", "March",
10                             "April", "May", "June", "July",
11                             "August", "September", "October",
12                             "November", "December" };
13 public:
14     DayOfYear (int num){
15         this -> day = num;
16     }
17     void print(int num){
18         cout<<"Day "<<this->day<<" would be "<<this->transition(num)<<endl;
19     }
20     string transition(int num){
21         string rtn;
22         if (num<=31){
23             stringstream ss;
24             ss<<num;
25             string date;
26             ss>>date;
27             ss.clear();
28             rtn += month[0] + " " + date;
29             return rtn;
30         }
31         if (num>31 && num<=59){
32             stringstream ss;
33             int cal = (num-31);
34             ss<<cal;
35             string date;
36             ss>>date;
37             ss.clear();
38             rtn += month[1] + " " + date;
39             return rtn;
40         }
41         if (num>59 && num<=90){
42             stringstream ss;
43             int cal = (num-59);
44             ss<<cal;
45             string date;
46             ss>>date;
47             ss.clear();
48             rtn += month[2] + " " + date;
49             return rtn;
50         }
51         if (num>90 && num<=120){
52             stringstream ss;
53             int cal = (num-90);
54             ss<<cal;
55             string date;
```

```
56         ss>>date;
57         ss.clear();
58         rtn += month[3] + " " + date;
59         return rtn;
60     }
61     if (num>120 && num<=151){
62         stringstream ss;
63         int cal = (num-120);
64         ss<<cal;
65         string date;
66         ss>>date;
67         ss.clear();
68         rtn += month[4] + " " + date;
69         return rtn;
70     }
71     if (num>151 && num<=181){
72         stringstream ss;
73         int cal = (num-151);
74         ss<<cal;
75         string date;
76         ss>>date;
77         ss.clear();
78         rtn += month[5] + " " + date;
79         return rtn;
80     }
81     if (num>181 && num<=212){
82         stringstream ss;
83         int cal = (num-181);
84         ss<<cal;
85         string date;
86         ss>>date;
87         ss.clear();
88         rtn += month[6] + " " + date;
89         return rtn;
90     }
91     if (num>212 && num<=243){
92         stringstream ss;
93         int cal = (num-212);
94         ss<<cal;
95         string date;
96         ss>>date;
97         ss.clear();
98         rtn += month[7] + " " + date;
99         return rtn;
100    }
101    if (num>243 && num<=273){
102        stringstream ss;
103        int cal = (num-243);
104        ss<<cal;
105        string date;
106        ss>>date;
107        ss.clear();
108        rtn += month[8] + " " + date;
109        return rtn;
```

```

109         return rtn;
110     }
111     if (num>273 && num<=304){
112         stringstream ss;
113         int cal = (num-273);
114         ss<<cal;
115         string date;
116         ss>>date;
117         ss.clear();
118         rtn += month[9] + " " + date;
119         return rtn;
120     }
121     if (num>304 && num<=334){
122         stringstream ss;
123         int cal = (num-304);
124         ss<<cal;
125         string date;
126         ss>>date;
127         ss.clear();
128         rtn += month[10] + " " + date;
129         return rtn;
130     }
131     if (num>334 && num<=365){
132         stringstream ss;
133         int cal = (num-334);
134         ss<<cal;
135         string date;
136         ss>>date;
137         ss.clear();
138         rtn += month[11] + " " + date;
139         return rtn;
140     }
141 }
142 };
143 int main(int argc, char* argv[]) {
144     int times,num;
145     cout<<"Please enter how many times u want to transit. ";
146     cin>>times;
147     while(times--){
148         cout<<"Enter the day u want to know: ";
149         cin>>num;
150         DayOfYear dayofyear(num);
151         dayofyear.print(num);
152     }
153     return 0;
154 }

```

Output

```

Please enter how many times u want to transit. 3
Enter the day u want to know: 15
Day 15 would be January 15
Enter the day u want to know: 37
Day 37 would be February 6
Enter the day u want to know: 62
Day 62 would be March 3

```

3.

```
1  #include <iostream>
2  using namespace std;
3
4  class Pstring : public string{
5      private:
6
7      public:
8          Pstring() : string(){}
9          Pstring(const char* s) : string(s){}
10         Pstring(const string& str) : string(str){}
11         bool isPalindrome(){
12             int size = this->size()/2;
13             for(int i = 0; i < size; i++) {
14                 if(this->operator[](i) != this->operator[](this->size() - 1 - i)) return false;
15             }
16             return true;
17         }
18     };
19
20 int main(int argc, char* argv[]){
21     Pstring pstring;
22     string s;
23     while(true) {
24         cout << "Try to input something." << endl;
25         getline(cin, pstring);
26         if(pstring.isPalindrome()){
27             cout << "It is Palindrome!!" << endl << endl;
28         } else {
29             cout << "It is not Palindrome!!" << endl << endl;
30         }
31     }
32     return 0;
33 }
```

Output

```
aloola
It is Palindrome!!

Try to input something.
rteij
It is not Palindrome!!

Try to input something.
oial[
It is not Palindrome!!
```

4.

```

4.cpp > main(int, char * [])
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4  class DivSales{
5      private:
6          int season[4];
7          int sum;
8      public:
9          DivSales(){
10             int i=4;
11             while(i-->0) this->season[i] = 0;
12             this->sum = 0;
13         }
14         void sales(int a, int b, int c, int d){
15             season[0] = a;
16             season[1] = b;
17             season[2] = c;
18             season[3] = d;
19         }
20         int getsales(int num){
21             return season[num];
22         }
23         int getsum(){
24             for(int i = 0; i < 4 ; i++){
25                 this->sum += this->season[i];
26             }
27             return this->sum;
28         }
29     };
30     int main(int argc, char* argv[]){
31         DivSales divsales[6];
32         int input[4];
33         for(int i = 0; i < 6; i++){
34             cout<<"====division sale====\n";
35             cout<<"-          # "<<i+1<<"          -\n";
36             for (int j = 0; j < 4; j++){
37                 cout<<"Sales for season "<<j+1<<": ";
38                 cin>>input[j];
39             }
40             cout<<"====\n";
41             divsales[i].sales(input[0], input[1], input[2], input[3]);
42         }
43         cout<<"\n====Final report====\n";
44         for(int i = 0; i < 6; i++){
45             cout<<"Division "<<i+1<<" ";
46             for (int j = 0; j < 4; j++){
47                 cout<<setw(5)<<divsales[i].getsales(j)<<" ";
48             }
49             cout<<"\n-----\n";
50         }
51         int sum = 0;
52         for(int i = 0; i < 6 ; i++){
53             sum += divsales[i].getsum();
54         }
55         cout<<"The whole sales is "<<setw(7)<<sum<<endl;
56         return 0;
57     }

```

Output

```

====division sale====
-          # 1          -
Sales for season 1: 1
Sales for season 2: 1
Sales for season 3: 1
Sales for season 4: 1
====division sale====
-          # 2          -
Sales for season 1: 2
Sales for season 2: 2
Sales for season 3: 2
Sales for season 4: 2
====division sale====
-          # 3          -
Sales for season 1: 3
Sales for season 2: 3
Sales for season 3: 3
Sales for season 4: 3
====division sale====
-          # 4          -
Sales for season 1: 4
Sales for season 2: 4
Sales for season 3: 4
Sales for season 4: 4
====division sale====
-          # 5          -
Sales for season 1: 5
Sales for season 2: 5
Sales for season 3: 5
Sales for season 4: 5
====division sale====
-          # 6          -
Sales for season 1: 6
Sales for season 2: 6
Sales for season 3: 6
Sales for season 4: 6
====
====Final report====
Division 1      1      1      1      1
-----
Division 2      2      2      2      2
-----
Division 3      3      3      3      3
-----
Division 4      4      4      4      4
-----
Division 5      5      5      5      5
-----
Division 6      6      6      6      6
-----
The whole sales is 84

```

5.

```
5.cpp > Rational
1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4  class Rational{
5      private:
6          int num;
7          int den;
8          void reduce(){
9              int gcd = this->gcd(abs(num), abs(den));
10             num = num/gcd;
11             den = den/gcd;
12         }
13         int gcd(int x, int y) {
14             int r = 0;
15             while(y != 0) {
16                 r = x % y;
17                 x = y;
18                 y = r;
19             }
20             return x;
21         }
22     public:
23     Rational(int num, int den){
24         this->num = num;
25         this->den = den;
26     }
27
28     friend ostream& operator<<(ostream &os, const Rational &r) {
29         return os << r.rational_to_string();
30     }
31     Rational operator+(const Rational &that) {
32         Rational rational(
33             this->num * that.den + that.num * this->den,
34             this->den * that.den
35         );
36         rational.reduce();
37         return rational;
38     }
39     Rational operator-(const Rational &that) {
40         Rational rational(
41             this->num * that.den - that.num * this->den,
42             this->den * that.den
43         );
44         rational.reduce();
45         return rational;
46     }
47     Rational operator*(const Rational &that) {
48         Rational rational(
49             this->num * that.num,
50             this->den * that.den
51         );
52         rational.reduce();
53         return rational;
54     }
55     Rational operator/(const Rational &that) {
```

```

56     Rational rational(
57         this->num * that.den,
58         this->den * that.num
59     );
60     rational.reduce();
61     return rational;
62 }
63 string rational_to_string() const {
64     return to_string(this->num) + "/" + to_string(this->den);
65 }
66
67 };
68
69 int main(int argc, char* argv[]){
70     int num;
71     int den;
72     char option, tmp;
73     while(true){
74         cout<<"input something:";
75         cin>>num>>tmp>>den;
76         Rational rational1(num, den);
77
78         cin>>option;
79         cin>>num>>tmp>>den;
80         Rational rational2(num, den);
81         if(option == '+')      cout<<"="<<(rational1 + rational2)<<endl;
82         else if(option == '-') cout<<"="<<(rational1 - rational2)<<endl;
83         else if(option == '*') cout<<"="<<(rational1 * rational2)<<endl;
84         else if(option == '/') cout<<"="<<(rational1 / rational2)<<endl;
85     }
86     return 0;
87 }

```

Output

```

input something:2/3+6/9
=4/3
input something:1/7-14/1
=-97/7
input something:3/5*1/2
=3/10
input something:3/5/1/2
=6/5

```