計算機程式語言

作　　業　　三

姓名	潘廣霖
學號	B03203004
原始程式檔名	HW03_002.cpp (_ = [A-D])

評　分　項　目

分數比重	項　　目	得　分
40%	程式是否能正確執行?	
40%	程式之使用者介面與輸出結果?	
	是否有繳交原始程式檔與執行檔?	
	程式中的註解是否恰當?	
	程式之結構與邏輯是否正確?	
	程式碼的格式是否合乎要求?	
20%	程式之綜合評分	
總　分		

評語:

A.

```
//============================================================
//  PROGRAMMER  : 潘廣霖
//  DATE        : 2015-10-22
//  FILENAME    : HW03A002.CPP
//  DESCRIPTION : Calculate PI using Leibniz series
//============================================================

#include "stdafx.h"
#include<algorithm>
#include<cstdlib>
#include<iomanip>
using namespace std;

int main(int argc, _TCHAR* argv[]) {
    // inaccurate after saved as       11...
    const double PI = 3.141592653589793238463;

    // specify precision
    cout << setprecision(15);

    // print the accurate PI (in double) for comparing
    // note the spaces inserted here is for aligning with the result
    cout << setw(20) << right << "ACCURATE PI = "
         << setw(16) << PI
         << endl;

    double s = 0;
    for (int k = 0; k <= 1e8; k++) {
        // 1-(k%2)*2 -- to apply negative sign if k % 2 == 1
        s += 4 * (1.0 - (k % 2) * 2) / (2 * k + 1);
        if (k == 1e3 || k == 1e4 || k == 1e5 || k == 1e6 || k == 1e7 || k ==
1e8) {
            cout << "k = "     << setw(9) << k
                 << ", pi = "  << setw(16) << s
                 << ", err = " << setw(16) << fixed << (s - PI)
                 << endl;
            // clear fixed flag
            // ref: http://stackoverflow.com/a/12094984/2281355
                cout.unsetf(ios_base::fixed);

            if (k == 1e5) {
                // print a separator
                    cout << "------" << endl;
                }
        }
    }
    system("pause");
}
```

B.

```
//=================================================================
//  PROGRAMMER  : 潘廣霖
//  DATE        : 2015-10-22
//  FILENAME    : HW03B002.CPP
//  DESCRIPTION : Find all prime numbers within 1000
//=================================================================

// the boundry of integers we are instrested in
#define MAX 1000

#include "stdafx.h"
#include<iostream>
#include<algorithm>
#include<cstdlib>
#include<iomanip>
using namespace std;

int main(int argc, _TCHAR* argv[]) {
    // np[i] == false iff number i is prime
    bool np[MAX + 1];
    for (int i = 0; i < MAX; i++) { np[i] = false; }

    // starting from 2, remove 2*2, 2*3, ... from list
    for (int i = 2; i < MAX; i++) {
        // skip numbers that are already removed
        if (np[i]) continue;
        for (int j = 2 * i; j < MAX; j += i) {
            np[j] = true;
        }
    }

    cout << "PRIME TABLE: " << endl;
    for (int i = 2, cnt = 0; i < MAX; i++) {
        if (!np[i]) {
            cout << setw(5) << i << " ";
            cnt++;
            if (cnt % 12 == 0) {
                cout << endl;
            }
        }
    }

    cout << endl;

    cout << "TWIN PRIMES:" << endl;
    for (int i = 4, cnt = 0; i < MAX; i++) {
        if (!np[i] && !np[i-2]) {
            cout << "("
```

```
                << setw(3) << (i - 2)
                << ", "
                << setw(3) << i
                << ")  ";
            cnt++;
            if (cnt % 6 == 0) {
                cout << endl;
            }
        }
    }

    cout << endl;

    system("pause");
}
```

C.

```
//============================================================
//  PROGRAMMER  : 潘廣霖
//  DATE        : 2015-10-22
//  FILENAME    : HW03C002.CPP
//  DESCRIPTION : Simulate a traditional 12-key dialer
//============================================================

#include "stdafx.h"
#include<iostream>
using namespace std;

int main(int argc, _TCHAR* argv[]) {
    string keys[10] = {"",
        " ",    "ABC", "DEF",
        "GHI",  "JKL", "MNO",
        "PQRS", "TUV", "WXYZ"
    };

    string output;
    int prev = 0, cnt = 0, c;
    char tok;
    while(cin >> tok) {
        c = tok - '0';
        if (!prev && !c) {
            break;
        } else if (prev == c) {
            // the same key as previous one
            cnt++;
        } else if (prev != 0) {
            // flush it
            output += keys[prev][cnt % keys[prev].length()];
            cnt = 0;
            prev = 0;
        }
        prev = c;
    }
    cout << output << endl;

    cin.get();
}
```

D.

```
//=================================================================
//  PROGRAMMER  : 潘廣霖
//  DATE        : 2015-10-22
//  FILENAME    : HW03D002.CPP
//  DESCRIPTION : GCD table of number 1 to 20
//=================================================================

#include "stdafx.h"
#include<iostream>
#include<iomanip>
#include<cstdlib>

using namespace std;

// my favorite GCD implementation
int gcd(int a, int b) {
    return a ? gcd(b%a, a) : b;
}


int main(int argc, _TCHAR* argv[]) {

    // table header
    cout << setw(6) << "";
    for (int i = 1; i <= 20; i++) {
        cout << setw(3) << right << i;
    }

    // horizonal line
    cout << endl
         << setw(68) << setfill('=') << ""
         << endl;
    cout << setfill(' ');

    // table body
    for (int i = 1; i <= 20; i++) {
        cout << setw(3) << right << i
             << " | ";
        for (int j = 1; j <= 20; j++) {
            cout << setw(3) << right << gcd(i, j);
        }
        cout << endl;
    }


    system("pause");
}
```