

臺北市立大安高級工業職業學校 108 學年度第 1 學期 電子科

實習課程學習成果紀錄表

班級名稱	電子一甲	學號	0804117	姓名	張峻魁
科目名稱	程式設計實習		章節名稱	Ch6-3 工作項目五	
實作名稱	質數列表器				
功能說明	藉由列出指定範圍內的質數，體驗電腦計算能力的強大。				
評量項目	等級(A 最高，E 最低)			總評結果	
程式說明	<input type="checkbox"/> A <input type="checkbox"/> B <input type="checkbox"/> C <input type="checkbox"/> D <input type="checkbox"/> E				
執行結果	<input type="checkbox"/> A <input type="checkbox"/> B <input type="checkbox"/> C <input type="checkbox"/> D <input type="checkbox"/> E				
心得感想	<input type="checkbox"/> A <input type="checkbox"/> B <input type="checkbox"/> C <input type="checkbox"/> D <input type="checkbox"/> E				

心得感想

一個簡單的問題，藉由許多不同的數學方法得到答案。但直接簡單的做法，可能較費時；複雜的作法則可能更省時，藉由比較不同計算方式，我學習到尋找好策略的重要性。大程式不一定比較功能強大；小程式不一定效率高。在研究的過程中，也慢慢發現更多可變化的細節，且可能加速質數的判定，經過這番努力，這隻程式才漸漸成形，漸漸成為畫面上所見的樣貌。

程式說明

```
#include <iostream> // 匯入標頭檔 iostream
#include <iomanip> // 匯入標頭檔 iomanip
#include <cmath> // 匯入標頭檔 cmath
#include <ctime> // 匯入標頭檔 ctime
using namespace std; // 宣告命名空間 std
// 空虛的空行
int i,j,total=0,sqrt_i; // 定義整數變數
const short store_prime_range=168; // 設定緩存空間長度
short prime[store_prime_range],last=2; // 設定緩存空間
bool isPrime; // 定義布林變數
// 空虛的空行
int all_check(int input){ // 副程序(試除法)
    for(i=2;i<=input;i++){ // 對所有數做檢查
        sqrt_i=sqrt(i); // 暫存平方根值
        if(sqrt_i*sqrt_i==i) // 檢查平方數
            isPrime=false; // 若為平方數，則不是質數
        else // 否則
            isPrime=true; // 可能是質數
        for(j=2;j<i && isPrime;j++) // 檢查是否為質數
            if(i%j==0) // 若找到因數
                isPrime=false; // 則不是質數
        if(isPrime==true){ // 若為質數
            cout<<" "<<i; // 輸出
            total++; // 累加質數
        } // 檢測完畢
    } // 所有數皆檢測完畢
    return total; // 回傳累加
} // 程序結束
// 空虛的空行
int all_check2(int input){ // 副程序(試除法，跳過偶數)
    if(input>=2) cout<<" "<<2; // 對 2 做例外處理
    for(i=3;i<=input;i+=2){ // 對所有數做檢查
        sqrt_i=sqrt(i); // 暫存平方根值
        if(sqrt_i*sqrt_i==i) // 檢查平方數
            isPrime=false; // 若為平方數，則不是質數
        else // 否則
            isPrime=true; // 可能是質數
        for(j=2;j<i && isPrime;j++) // 檢查是否為質數
            if(i%j==0) // 若找到因數
                isPrime=false; // 則不是質數
        if(isPrime==true){ // 若為質數
            cout<<" "<<i; // 輸出
            total++; // 累加質數
        } // 檢測完畢
    } // 所有數皆檢測完畢
    if(input>=2) total++; // 對 2 做例外處理
    return total; // 回傳累加
} // 程序結束
// 空虛的空行
int erato(int input){ // 副程序(埃拉托斯尼特法)
    for(i=2;i<=input;i++){ // 對所有數做檢查
        sqrt_i=sqrt(i); // 暫存平方根值
        if(sqrt_i*sqrt_i==i) // 檢查平方數
```

```

        isPrime=false; // 若為平方數，則不是質數
    else // 否則
        isPrime=true; // 可能是質數
    for(j=2;j<=sqrt_i && isPrime;j++) // 檢查是否為質數
        if(i%j==0) // 若找到因數
            isPrime=false; // 則不是質數
    if(isPrime==true){ // 若為質數
        cout<<" "<<i; // 輸出
        total++; // 累加質數
    } // 檢測完畢
} // 所有數皆檢測完畢
return total; //回傳累加
} // 程序結束

```

// 空虛的空行

```

int erato2(int input){ // 副程序(埃拉托斯尼特法，跳過偶數)
    if(input>=2) cout<<" "<<2; // 對 2 做例外處理
    for(i=3;i<=input;i+=2){ // 對所有奇數做檢查
        sqrt_i=sqrt(i); // 暫存平方根值
        if(sqrt_i*sqrt_i==i) // 檢查平方數
            isPrime=false; // 若為平方數，則不是質數
        else // 否則
            isPrime=true; // 可能是質數
        for(j=3;j<=sqrt_i && isPrime;j+=2) // 檢查是否為質數
            if(i%j==0) // 若找到因數
                isPrime=false; // 則不是質數
        if(isPrime==true){ // 若為質數
            cout<<" "<<i; // 輸出
            total++; // 累加質數
        } // 檢測完畢
    } // 所有奇數皆檢測完畢
    if(input>=2) total++; // 對 2 做例外處理
    return total; //回傳累加
} // 程序結束

```

// 空虛的空行

```

int erato_temp(int input){ // 副程序(埃拉托斯尼特法，使用緩存)
    for(i=2;i<=input;i++){ // 對所有數做檢查
        sqrt_i=sqrt(i); // 暫存平方根值
        if(sqrt_i*sqrt_i==i) // 檢查平方數
            isPrime=false; // 若為平方數，則不是質數
        else // 否則
            isPrime=true; // 可能是質數
        for(j=0;j<store_prime_range && j<total;j++) // 優先檢查緩存內質數
            if(i%prime[j]==0) // 若找到因數
                isPrime=false; // 則不是質數
        for(j=last+1;j<=sqrt_i && isPrime;j++) // 檢查其他可能
            if(i%j==0) // 若找到因數
                isPrime=false; // 則不是質數
        if(isPrime==true){ // 若為質數
            cout<<" "<<i; // 輸出
            if(total<store_prime_range){ // 若緩存未滿
                prime[total]=i; // 列入緩存
                last=i; // 記為末項
            } // 寫入完成
            total++; // 累加質數
        } // 檢測完畢
    }
}

```

```

    } // 所有數皆檢測完畢
    return total; //回傳累加
} // 程序結束
// 空虛的空行
int erato2_temp(int input){ // 副程序(埃拉托斯尼特法，跳過偶數，使用緩存)
    if(input>=2) cout<<" "<<2; // 對 2 做例外處理
    for(i=3;i<=input;i+=2){ // 對所有奇數做檢查
        sqrt_i=sqrt(i); // 暫存平方根值
        if(sqrt_i*sqrt_i==i) // 檢查平方數
            isPrime=false; // 若為平方數，則不是質數
        else // 否則
            isPrime=true; // 可能是質數
        for(j=0;j<store_prime_range && j<total;j++) // 優先檢查緩存內質數
            if(i%prime[j]==0) // 若找到因數
                isPrime=false; // 則不是質數
        for(j=last+2;j<=sqrt_i && isPrime;j+=2) // 檢查其他可能
            if(i%j==0) // 若找到因數
                isPrime=false; // 則不是質數
        if(isPrime==true){ // 若為質數
            cout<<" "<<i; // 輸出
            if(total<store_prime_range){ // 若緩存未滿
                prime[total]=i; // 列入緩存
                last=i; // 記為末項
            } // 寫入完成
            total++; // 累加質數
        } // 檢測完畢
    } // 所有奇數皆檢測完畢
    if(input>=2) total++; // 對 2 做例外處理
    return total; //回傳累加
} // 程序結束
// 空虛的空行
main(){ // 主程序
    int in,in2,total,time; // 定義輸入變數
    cout<<"(1) 試除法"<<endl; // 輸出功能資訊
    cout<<"(2) 試除法 (跳過偶數)"<<endl; // 輸出功能資訊
    cout<<"(3) 埃拉托斯尼特法"<<endl; // 輸出功能資訊
    cout<<"(4) 埃拉托斯尼特法 (跳過偶數)"<<endl; // 輸出功能資訊
    cout<<"(5) 埃拉托斯尼特法 (使用緩存)"<<endl; // 輸出功能資訊
    cout<<"(6) 埃拉托斯尼特法 (跳過偶數 & 使用緩存)"<<endl; // 輸出功能資訊
    cout<<"請輸入要執行的演算法(代號)："<<endl; // 輸出功能資訊
    cin>>in; // 接收輸入
    cout<<"請輸入一個數字："<<endl; // 輸出功能資訊
    cin>>in2; // 接收輸入
    time=clock(); // 紀錄時間
    if(in==1) total=all_check(in2); // 試除法
    if(in==2) total=all_check2(in2); // 試除法(跳過偶數)
    if(in==3) total=erato(in2); // 埃拉托斯尼特法
    if(in==4) total=erato2(in2); // 埃拉托斯尼特法(跳過偶數)
    if(in==5) total=erato_temp(in2); // 埃拉托斯尼特法(使用緩存)
    if(in==6) total=erato2_temp(in2); // 埃拉托斯尼特法(跳過偶數)(使用緩存)
    cout<<endl<<"共 "<<total<<" 個質數"<<endl; // 輸出累加
    cout<<"共費時 "<<(float)(clock()-time)/CLOCKS_PER_SEC<<" 秒"<<endl; // 輸出時間差
    system("pause"); // 暫停畫面，方便使用者閱讀
} // 程式結束

```

執行結果

```
命令提示字元 - C:\Users\user\Desktop\6-3-work5_extend.exe
C:\Users\user\Desktop\6-3-work5_extend.exe
(1) 試除法
(2) 試除法 (跳過偶數)
(3) 埃拉托斯尼特法
(4) 埃拉托斯尼特法 (跳過偶數)
(5) 埃拉托斯尼特法 (使用緩存)
(6) 埃拉托斯尼特法 (跳過偶數 & 使用緩存)
請輸入要執行的演算法(代號): 6
請輸入一個數字: 1000000
```

```
命令提示字元 - C:\Users\user\Desktop\6-3-work5_extend.exe
994199 994229 994237 994241 994247 994249 994271 994297 994303 994307 9943
09 994319 994321 994337 994339 994363 994369 994391 994393 994417 994447 99
4453 994457 994471 994489 994501 994549 994559 994561 994571 994579 994583
994603 994621 994657 994663 994667 994691 994699 994709 994711 994717 99472
3 994751 994769 994793 994811 994813 994817 994831 994837 994853 994867 994
871 994879 994901 994907 994913 994927 994933 994949 994963 994991 994997 9
95009 995023 995051 995053 995081 995117 995119 995147 995167 995173 995219
995227 995237 995243 995273 995303 995327 995329 995339 995341 995347 9953
63 995369 995377 995381 995387 995399 995431 995443 995447 995461 995471 99
5513 995531 995539 995549 995551 995567 995573 995587 995591 995593 995611
995623 995641 995651 995663 995669 995677 995699 995713 995719 995737 99574
7 995783 995791 995801 995833 995881 995887 995903 995909 995927 995941 995
957 995959 995983 995987 995989 996001 996011 996019 996049 996067 996103 9
96109 996119 996143 996157 996161 996167 996169 996173 996187 996197 996209
996211 996253 996257 996263 996271 996293 996301 996311 996323 996329 9963
61 996367 996403 996407 996409 996431 996461 996487 996511 996529 996539 99
6551 996563 996571 996599 996601 996617 996629 996631 996637 996647 996649
996689 996703 996739 996763 996781 996803 996811 996841 996847 996857 99685
9 996871 996881 996883 996887 996899 996953 996967 996973 996979 997001 997
013 997019 997021 997037 997043 997057 997069 997081 997091 997097 997099 9
97103 997109 997111 997121 997123 997141 997147 997151 997153 997163 997201
997207 997219 997247 997259 997267 997273 997279 997307 997309 997319 9973
27 997333 997343 997357 997369 997379 997391 997427 997433 997439 997453 99
7463 997511 997541 997547 997553 997573 997583 997589 997597 997609 997627
997637 997649 997651 997663 997681 997693 997699 997727 997739 997741 99775
1 997769 997783 997793 997807 997811 997813 997877 997879 997889 997891 997
897 997933 997949 997961 997963 997973 997991 998009 998017 998027 998029 9
98069 998071 998077 998083 998111 998117 998147 998161 998167 998197 998201
998213 998219 998237 998243 998273 998281 998287 998311 998329 998353 9983
77 998381 998399 998411 998419 998423 998429 998443 998471 998497 998513 99
8527 998537 998539 998551 998561 998617 998623 998629 998633 998651 998653
998681 998687 998689 998717 998737 998743 998749 998759 998779 998813 99881
9 998831 998839 998843 998857 998861 998897 998909 998917 998927 998941 998
947 998951 998957 998969 998983 998989 999007 999023 999029 999043 999049 9
99067 999083 999091 999101 999133 999149 999169 999181 999199 999217 999221
999233 999239 999269 999287 999307 999329 999331 999359 999371 999377 9993
89 999431 999433 999437 999451 999491 999499 999521 999529 999541 999553 99
9563 999599 999611 999613 999623 999631 999653 999667 999671 999683 999721
999727 999749 999763 999769 999773 999809 999853 999863 999883 999907 99991
7 999931 999953 999959 999961 999979 999983
共 78498 個質數
共費時 13.346 秒
請按任意鍵繼續 . . .
```