

VG101 — Introduction to Computer and Programming

Assignment 5 (15/11/2016)

Manuel — UM-JI (Fall 2016)

- Write each exercise in a different file
- Include simple comments in the code
- If applicable, split the implementation over several functions
- Write a single README file per assignment
- Archive all the files in a zip file and upload it onto Sakai

Ex. 1 — Array

Write a program taking an integer n as input and displaying all the primes less than n . All the primes should be stored in an array.

Specifications.

- Start by reading the integer n
- Use comments to clearly indicate which array contains the primes

Ex. 2 — Arrays and functions

Write a function which takes as input a month, and the name of the first day of the month. It should display the calendar for the requested month.

Input specifications.

- First input is the month as an integer
- Second input is the day of the month as a string

Ex. 3 — Loop, array, and sorting

The goal of this exercise is to write a C program that simulates a deck of 52 cards. Start by printing the cards in the following order, $2 < 3 < \dots < 10 < Jack < Queen < King < Ace$ assuming *Spades* < *Hearts* < *Diamonds* < *Clubs*. Then shuffle them, print them out in their shuffled order, sort them following the above order, and print the resulting deck.

Specifications.

- A total of three decks should be displayed
- The user should press enter each time a deck is printed

Ex. 4 — Strings

Write a program to find the number of times a given string occurs in a sentence. The user will input both the sentence and the word.

Sample output (ex. 4)

```
$ ./h5 -ex4
Input a sentence: good morning, have you seen the cat and the dog?
Input a string: the
The string 'the' occurs 2 times
```

Group Exercise

The goal of the following exercise is to practice programming as a *pair*. For a better group work experience the following scenario is recommended.

- Sit in a comfortable environment and work together as a team;
- A student plays the “Driver” and the other one the “Navigator”;
- The *driver*’s work is to type on the keyboard while the *navigator* provides suggestions;
- Both the *driver* and the *navigator* should pay attention to common typos and errors;
- Roles can be exchanged after a while;
- Both students are expected to think of the whole problem;

Please respect your partner and get in touch *as soon as possible*. Cooperate and give the best for everybody to fully benefit from this arrangement.

Ex. 5 — Low level C programming

The following program performs a multiplication using an algorithm similar to the one from Karatsuba.

1. Detail Karatsuba algorithm in the README file (search it on internet).
2. Add comments to the code to describe what is done, line by line.
3. Explain in the README file what specific adjustments were made to the algorithm in order to improve the efficiency.
4. Search online what is a divide and conquer strategy.
5. Using a divide and conquer approach, together with the operators `&`, `|`, `<<` and `>>`, write an efficient function to replace the for loops marked as “not optimal”.

Low level multiplication (ex. 5)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  #define SWAP(a,b) { a ^= b; b ^= a; a ^= b; }
5  unsigned long int mult(unsigned long int a, unsigned long int b);
6  int main () {
7      unsigned long int a, b;
8      srand(time(NULL));
9  #ifndef TEST
10     a=rand(); b=rand();
11     printf("%ld*%ld=%ld %ld\n",a,b,mult(a,b), RAND_MAX);
12 #endif
13 #ifdef TEST
14     int i;
15     for(i=0; i< 1000000; i++) {
16         a=rand(); b=rand();
17         if(mult(a,b)!=a*b) {
18             fprintf(stderr,"Error (%d): a=%ld, b=%ld, a*b=%ld, k(a,b)=%ld\n",\
19                 i,a,b,a*b,mult(a,b));
20             exit(-1);

```

```
21     }
22 }
23 #endif
24 }
25 unsigned long int mult(unsigned long int a, unsigned long int b) {
26     int i, n, N;
27     unsigned long int x0,y0,z0,z1=1;
28     if(a<b) SWAP(a,b);
29     if(b==0) return 0;
30     for(n=-1, i = 1; i <= b; i<=1, n++); /* not optimal */
31     for(N=n; i <= a; i<=1, N++);
32
33     y0=b&((1<<n)-1);
34     x0=a&((1<<N)-1);
35     z0=mult(x0,y0);
36     i=N+n;
37     return ((z1<<i)+(x0<<n)+(y0<<N)+z0);
38 }
```

Groups

苏浩雄 (Haoxiong Su) 516021910707
熊家正 (Xiong Jiazheng) 516021910564

朱皓轩 (Zhu Haoxuan) 516370910157
周瑞星 (Zhou Ruixing) 516370910130

何航吉 (He Hangji) 516370910247
秦宇隽 (Qin Yujun) 516370910199

黄子瀛 (Huang Ziyang) 516370910169
沈阳 (Shen) 516370910040

王菁滢 (Wang) 516370910029
江南佳 (Nanjia Jiang) 516370910212

刘知正 (Liu Zhizheng) 516370910063
芮意进 (Rui Yijin) 516370910254

吕步 (Lv) 516370910173
王嫣然 (Yanran Wang) 516370910082

金浩 (Jin) 516370910116
廖昕皓 (Liao) 516370910037

陶元杰 (Tao Yuanjie) 516370910068
杨瑞敏 (Yang Ruimin) 516370910127

王师溟 (Wang Shi) 516021910693
刘嘉晨 (Liu Jiachen) 516370910079

叶柔霜 (Ye Roushuang) 516370910241
瞿昱易 (Qu Yuyi) 516370910238

胡哲榜 (Hu Zhebang) 516370910115
陆思轶 (Lu Siyi) 5133709149

朱文耀 (Zhu Wenyao) 5143709246
石一茗 (Shi Yiming) 516370910108

彭以绘 (Peng Yihui) 516370910213
郑智心 (Zheng Zhixin) 516370910083

SARIT KITTIRATTANAPAIBOON 516370990012
苏镜瑜 (Su Jing) 516370910123

陈嘉懿 (Chen Jiayi) 516370910211
孙旭华 (Sun Xuhua) 516370910255

秦亦蕉 (Qin Yijiao) 516021910124
沈叶沁 (Shen Yeqin) 516370910162

王依勤 (Wang Yiqin) 516370910109
OSAMA MALIK AWAN 516370990003

陈佳茜 (Chen Jiaxi) 516370910159
葛天逸 (Ge Tianyi) 516370910168

孙琦岳 (Sun Qiyue) 516021910259
关书文 (Guan Shuwen) 516021910603

李金儒 (Li Jinru) 516370910036
朱柯闻 (Zhu Keyue) 516370910158

王天子 (Wang Tianyu) 516370910256
朱波颖 (Zhu Boying) 516370910165

TAE HOON KWEON 516370990005
翁可歆 (Weng Kexin) 516370910163

陶奕彤 (Tao Yitong) 516370910081
聂鞠荷 (Nie Juhe) 516021910662

陈东箭 (Chen Dongjian) 516370910166
范哲良 (Fan Zheliang) 516021910518

潘智辉 (Pan Zhihui) 516370910174
林牧星 (Lin Muxing) 516370910078

屠立芸 (Tu Liyun) 516370910240
王涵之 (Wang Hanzhi) 516021910768

王奕纬 (Wang Yiwei) 516370910043
周宇航 (Zhou Yuhang) 516021910296

杜轲 (Du Ke) 516370910114
周耘平 (Chou YunPing) 516370910156

何屹滔 (He Yitao) 516370910248
姜乐钧 (Jiang Lejun) 516370910250

钱星 (Qian Xingyue) 516370910028
丁恺雯 (Ding) 516370910024

刘惟中 (Liu Weizhong) 515370910160
汪庭康 (Wang Tingkang) 516370910124

马至皓 (Ma Zhihao) 516370910039
刘诗雨 (Liu Shiyu) 516370910027

胡炳城 (Hu Bingcheng) 516021910219
戚伏波 (Qi Fubo) 516370910175

任欣阳 (Ren Xinyang) 516370910080
成隽奕 (Cheng Junyi) 516370910032

周家成 (Zhou Jiacheng) 516370910075
朱文琪 (Wenqi Zhu) 516370910217

谢舒翔 (Xie Shuxiang) 516370910070
项之渊 (Xiang Zhiyuan) 516370910126

潘崇聃 (Pan Chongdan) 516370910121
苏星宇 (Su Xingyu) 516021910560

唐星辉 (Tang Xing) 516370910042
孙天行 (Sun Tianxing) 516021910289

宣思源 (Xuan Siyuan) 516370910215
闫李豪 (Yan Lihao) 516021910715

郭成彰 (Guo Chengzhang) 516021910639
李雨姗 (Li Yushan) 516370910026

Delmwin BAEKA 516370990007
孙艺 (Sun Yi) 516370910214

朴柱薰 (Pu Zhu) 516370990013
孟天宇 (Meng Tianyu) 516370910065

JESCO ALEXANDER BECK 714370990038
鲍家君 (Bao Jiajun) 516370910112

洪颖慧 (Hong Yinghui) 516370910160
鲁瑞明 (Lu Ruiming) 516021910167

张沈宇 (Zhang Shenyu) 516370910129
刘倪逸秋 (Liu Niyiqu) 516370910118

陈曦雯 (Chen Xiwen) 516021910271
朱昊杰 (Zhu Haojie) 515010910035

姜渊清 (Lou Yuanqing) 516370910120
郭凌云 (Guo Lingyun) 516021910303

沈琮凯 (Shen Congkai) 516370910066
耿若馨 (Geng Ruoxin) 516370910077

刘紫薇 (Ziwei Liu) 516370910161
仰佳欣 (Yang Jiaxin) 516370910205

陈培煜 (Chen Peiyu) 516021910369
吴明远 (Wu Mingyuan) 516370910125

乐一炜 (Le Yiwei) 516370910117
沈子浩 (Shen Zihao) 516370910200

陈智博 (Chen Zhibo) 515020910276
陈天博 (Chen Tianbo) 5133709262

王瑞琦 (Wang Ruiqi) 516370910203
时广泽 (Shi Guangze) 516021910199

沈缘 (Shen Yuan) 516370910122
周家西 (Zhou Jiaxi) 516370910110

张佳宁 (Zhang Jianing) 516370910073
於宛灏 (Yu Wanhao) 516370910258

盛禹国 (Sheng Yuguo) 516370910041
李睿 (Li Rui) 516370910251

蒋沁诚 (Jiang Qincheng) 516370910034
董政元 (Dong Zhengyuan) 516370910113