

PTA 选择结构 E

rogeryoung

2021 年 04 月 13 日

目录

1	关于代码风格	2
2	简单判断	4
2.1	7-3 春夏秋冬	4
2.2	7-4 判断能否被 3, 5, 7 整除	4
2.3	7-5 有一门课不及格的学生	5
2.4	7-6 骑车与走路	5
2.5	7-10 于龙加	6
2.6	7-13 洛希极限	6
2.7	7-14 最简单的 if - else 练习: 乘法还是加法?	7
2.8	7-21 西安距离	7
2.9	7-24 编程实现两个分数相加	7
2.10	7-25 分支结构	8
2.11	7-26 有多少位是 7?	8
2.12	7-27 判断体质完整版	9
2.13	7-29 多分支表达-倍数问题	9
2.14	7-32 分段计算居民水费	9
2.15	7-35 计算分段函数 (双分支)	10
3	最值问题	10
3.1	7-16 找最大数和最小数	11
3.2	7-31 判断三角形的形状	11
4	嵌套判断	12
4.1	7-8 简单计算器	12
4.2	7-20 【分支】【-时制转换 A-】	13
4.3	7-28 多分支表达-数据奇偶判断	13
5	利用数组	14
5.1	7-17 帮小明出主意	14
5.2	7-23 根据输入的数字, 输出需要上课的节数	14
5.3	7-30 输入一个数字, 输出其对应的星期几的英文单词	15
5.4	7-33 输出星期名	15

PTA 选择结构 EASY 部分, [PDF](#)。

1 关于代码风格

我的代码风格可能和很多人不同,但只要遵从统一规范的代码都是风格良好的。我比较喜欢大括号放在同一行,因为这样可以一页看更多的代码。

假如不进行格式化,我想你是没有心情看这一串的

```
1 bool miller_rabbin(int n)
2 {   int ppp[10] = {2,7,61};
3     int a=n-1,b=0,j,v,x;
4     if(n<3) return n==2;
5     while(1-a&1) a>>=1,++b;
6     for(int i=0;i<=2;i++){
7         x = ppp[i];
8         if(n==x) return true;
9         v = power(x,a,n);
10        if(v==1||v==n-1) continue;
11        for(j=0;j<b;++j) {
12            v = v*v%n; if(v==n-1) break; }
13        if(j>=b) return false;
14        } return true;
15 }
```

格式化, 高亮

```
1 bool miller_rabbin(int n) {
2     if (n < 3)
3         return (n == 2);
4     int a = n - 1, b = 0;
5     while (1 - (a & 1)) {
6         a >>= 1;
7         ++b;
8     }
9     int prime[10] = {2, 7, 61};
10    for (int i = 0; i <= 2; i++) {
11        int x = prime[i];
12        if (n == x)
13            return true;
14        int v = power(x, a, n);
15        if (v == 1 || v == n - 1)
16            continue;
```

```
17     int j;
18     for (j = 0; j < b; j++) {
19         v = v * v % n;
20         if (v == n - 1)
21             break;
22     }
23     if (j >= b)
24         return false;
25 }
26 return true;
27 }
```

良好代码格式能够让程序本身的逻辑一目了然。

除了排版等要求不能太多空白时采取紧凑的代码风格，其他时候尽可能把代码 **展开写**。

具体的一些建议：

1 最基本的是好好缩进

代码是结构性的，良好的缩进可以体现出这种结构。

左大括号的缩进方式，缩进是用 space 还是 tab，这些争论倒无关紧要。风格统一就行。

2 不要压行，一行只做一件事

比如 `if` 条件后换行，内层缩进。

```
1  if (condition)
2      statement;
3
4  if (condition) {
5      statement;
6  } else {
7      statement;
8  }
```

`switch` 也是一样，分开写。

```
1  switch(condition) {
2      case 'A':
3          statement;
4          break;
5  }
```

这样，条件是什么，语句是什么，一目了然。

功能不同的代码片段之间空行。

3 重要的变量见名知意

比如一堆 `int a,b,c,d,e,f;`，要费很大劲猜这个变量究竟是什么意思，可读性很差。

4 尽可能延后变量的定义

尽可能延后变量的定义，尽可能缩小变量的作用域。

在现代的编辑器中，我们可以很方便的查找出该变量在何处定义，定义时有什么注释。定义的上下文一般都是和这个变量有关的信息，有助于我们理解变量的作用。

同样，声明不要全在开头挤着，用到了再定义。

5 贴代码

很多时候需要临时的把代码展示出来，比如交流思路时，把代码贴在 QQ 内会显得杂乱无章，截图又不方便调试。

这里建议贴于 [Ubuntu Pastebin](#)，能够提供良好的代码高亮和一定的保存时间，而且不需要登录。

将自己的代码格式化再发送是一种礼貌。

2 简单判断

部分题过于简单，不再解释。

2.1 7-3 春夏秋冬

```
1 int main() {
2     int n;
3     scanf("%d", &n);
4     n = n % 100;
5     if (n <= 2) {
6         printf("winter\n");
7     } else if (n <= 5) {
8         printf("spring\n");
9     } else if (n <= 8) {
10        printf("summer\n");
11    } else if (n <= 11) {
12        printf("autumn\n");
13    } else {
14        printf("winter\n");
15    }
16    return 0;
17 }
```

2.2 7-4 判断能否被 3, 5, 7 整除

注意末尾不能有空格，可以通过 flag 来判断。

```
1 int main() {
2     int n;
3     scanf("%d", &n);
4     int flag = 0;
5     if (n % 3 == 0) {
6         printf("3");
```

```

7         flag = 1;
8     }
9     if (n % 5 == 0) {
10         if (flag)
11             printf(" ");
12         printf("5");
13         flag = 1;
14     }
15     if (n % 7 == 0) {
16         if (flag)
17             printf(" ");
18         printf("7");
19         flag = 1;
20     }
21     if (!flag)
22         printf("n");
23     return 0;
24 }

```

2.3 7-5 有一门课不及格的学生

当低于 60 时 flag + 1, 只需判断 flag 是否恰为 1。

```

1 int main() {
2     int a,b;
3     scanf("%d %d", &a, &b);
4     int flag = 0;
5     if (a < 60)
6         flag++;
7     if (b < 60)
8         flag++;
9     if (flag == 1)
10        printf("1");
11    else
12        printf("0");
13    return 0;
14 }

```

2.4 7-6 骑车与走路

设距离为 x , 则骑车和步行所花的时间分别为

$$t_1 = 27 + x/3 + 23$$

$$t_2 = x/1.2$$

可以直接据此计算再判断，也可以算出关键点 $x = 100$ 再判断。

```

1  int main() {
2      int n;
3      scanf("%d", &n);
4      if (n < 100)
5          printf("Walk");
6      else if (n == 100)
7          printf("All");
8      else
9          printf("Bike");
10     return 0;
11 }
```

2.5 7-10 于龙加

题目有坑，结果不能以 0 开头。

```

1  int main() {
2      int a, b;
3      scanf("%d %d", &a, &b);
4      if (a == 0)
5          printf("%d", b);
6      else
7          printf("%d%d", a, b);
8      return 0;
9  }
```

2.6 7-13 洛希极限

题目的难点主要在于看不懂，我也帮不了什么（

```

1  int main() {
2      int type;
3      double f1, f2;
4      scanf("%lf %d %lf", &f1, &type, &f2);
5      if (type == 0) {
6          f1 *= 2.455;
7      } else {
8          f1 *= 1.26;
9      }
```

```

10     printf("%.2lf ", f1);
11     if (f1 > f2)
12         printf("T_T");
13     else
14         printf("^_^");
15     return 0;
16 }

```

2.7 7-14 最简单的 if - else 练习: 乘法还是加法?

```

1  int main() {
2      double f1, f2;
3      scanf("%lf %lf", &f1, &f2);
4      int op;
5      scanf("%d", &op);
6      if (op == 0) {
7          printf("%.2lf", f1 + f2);
8      } else {
9          printf("%.2lf", f1 * f2);
10     }
11     return 0;
12 }

```

2.8 7-21 西安距离

```

1  int main() {
2      int a, b, c, d;
3      scanf("%d %d %d %d", &a, &b, &c, &d);
4      printf("%d", abs(a-c) + abs(b-d));
5      return 0;
6  }

```

2.9 7-24 编程实现两个分数相加

通分, 用 gcd 化简。

$$\frac{a}{b} + \frac{c}{d} = \frac{ad + bc}{bd}$$

```

1  int gcd(int a, int b) {
2      return a ? gcd(b%a, a) : b;
3  }
4

```

```

5  int main() {
6      int a,b,c,d;
7      scanf("%d/%d+%d/%d", &a, &b, &c, &d);
8      int e = a*d + b*c;
9      int f = b * d;
10     int g = gcd(e,f);
11     printf("%d/%d+%d/%d=%d/%d", a, b, c, d, e/g, f/g);
12     return 0;
13 }

```

2.10 7-25 分支结构

题目有坑，非字母无需输出。

```

1  int main() {
2      char d;
3      scanf("%c", &d);
4      if ('A' <= d && d <= 'Z') {
5          printf("%d\n", d);
6      } else if ('a' <= d && d <= 'z') {
7          printf("%c\n", d - 32);
8      }
9      return 0;
10 }

```

2.11 7-26 有多少位是 7?

取模可以得到末位，除 10 再取模可以得到次末位的数字。因为不超过四位数，反复四次即可。字符串数组也可以。

```

1  int main() {
2      int n;
3      scanf("%d", &n);
4      int ans = 0;
5      for (int i = 0; i < 4; i++) {
6          if (n % 10 == 7)
7              ans++;
8          n /= 10;
9      }
10     printf("%d\n", ans);
11     return 0;
12 }

```

2.12 7-27 判断体质完整版

```
1  int main() {
2      double f1, f2;
3      scanf("%lf %lf", &f1, &f2);
4      double bmi = f1 / f2 / f2;
5      if (bmi < 18.5)
6          printf("偏瘦");
7      else if (bmi < 24)
8          printf("正常");
9      else if (bmi < 28)
10         printf("偏胖");
11     else if (bmi < 40)
12         printf("肥胖");
13     else
14         printf("极重度肥胖");
15     return 0;
16 }
```

2.13 7-29 多分支表达-倍数问题

我想了一会既是 5 的倍数又是 3 的倍数会怎样，没想出来。打算先交一次看看，结果过了。。

```
1  int main() {
2      int a;
3      scanf("%d", &a);
4      int f1, f2;
5      f1 = f2 = 0;
6      if (a % 3 == 0)
7          f1 = 1;
8      if (a % 5 == 0)
9          f2 = 1;
10     if (f1 && !f2) {
11         printf("%d", a % 5);
12     } else if (!f1 && f2) {
13         printf("%d", a % 3);
14     } else if (!f1 && !f2) {
15         printf("%d", a % 15);
16     }
17     return 0;
18 }
```

2.14 7-32 分段计算居民水费

```
1 int main() {
2     double x;
3     scanf("%lf", &x);
4     if (x < 0) {
5         printf("Input Data error!");
6     } else if (x <= 15) {
7         printf("%.2lf", 4 * x / 3);
8     } else {
9         printf("%.2lf", 2.5 * x - 17.5);
10    }
11    return 0;
12 }
```

2.15 7-35 计算分段函数（双分支）

```
1 int main() {
2     double x;
3     scanf("%lf", &x);
4     if(x == 0) {
5         printf("%.2lf", 0.0);
6     } else {
7         printf("%.2lf", 1 / x);
8     }
9     return 0;
10 }
```

3 最值问题

寻找最值有多种方法，这里介绍一种通用的方法。

关键的，对于数 a, b 取最大值只需

```
1 int max(int a, int b) {
2     if (a < b)
3         return b;
4     return a;
5 }
```

或者简写成三目运算符

```
1 return a < b ? b : a;
```

若需要取一个数组的最大值，可以通过反复取 `max`。

```
1 int a[N];
2 int m = a[0];
3 for (int i = 0; i < N; i++)
4     m = max(a[i], m);
```

3.1 7-16 找最大数和最小数

```
1 int main() {
2     int a, b, c;
3     scanf("%d %d %d", &a, &b, &c);
4     int max = a > b ? a : b;
5     max = max > c ? max : c;
6     int min = a < b ? a : b;
7     min = min < c ? min : c;
8     printf("max=%d,min=%d", max, min);
9     return 0;
10 }
```

3.2 7-31 判断三角形的形状

题目引号没标全，输出应为 简单三角形。可以直接判断，也可以像我一样排序后再判断。

若需取中间值，可以把三数求和后再减去最大最小值，剩下的自然是中间值。若数字范围很大，三数求和会溢出，可以用异或。

```
1 int main() {
2     int a, b, c;
3     scanf("%d %d %d", &a, &b, &c);
4     int max = a > b ? a : b;
5     max = max > c ? max : c;
6     int min = a < b ? a : b;
7     min = min < c ? min : c;
8     int mid = a ^ b ^ c ^ min ^ max;
9     // 如果不怕溢出 int，可以使用加法
10    // int mid = a + b + c - min - max;
11    if (min + mid <= max)
12        printf("NO");
13    else if (min == max)
14        printf("等边");
15    else if (min == mid && min*min + mid*mid == max*max)
16        printf("等腰直角");
17    else if (min == mid || max == mid)
18        printf("等腰");
```

```
19     else if (min*min + mid*mid == max*max)
20         printf("直角");
21     else
22         printf("普通三角形");
23     return 0;
24 }
```

4 嵌套判断

当分类只需简单的一层时，简单的判断即可。当分类错综复杂时，我们需要谨慎的理清它们的关系。

当一族分类完全的包含于一种分类下时，建议使用 `if` 嵌套，利用层次关系组织我们的程序。倘若摊成一层，不但增大了思考的难度，还增加了出问题的概率。

清晰的代码是最好的注释。

4.1 7-8 简单计算器

简单分类，注意 `%c` 需要严格对应。

```
1  int main() {
2      int a,b;
3      char op;
4      scanf("%d %d %c", &a, &b, &op);
5      int ans;
6      int flag = 1;
7      if (op == '+') {
8          printf("%d", a + b);
9      } else if (op == '-') {
10         printf("%d", a - b);
11     } else if (op == '*') {
12         printf("%d", a * b);
13     } else if (op == '/') {
14         if (b == 0)
15             printf("Divided by zero!\n");
16         else
17             printf("%d", a / b);
18     } else {
19         printf("Invalid operator!\n");
20     }
21     return 0;
22 }
```

4.2 7-20 【分支】【—时制转换 A—】

题目描述有误，读入需要加冒号，即 `scanf("%d:%d", &h, &m)`。
注意细节，12 点的情况需要详细考虑。

```
1 int main() {
2     int h, m;
3     scanf("%d:%d", &h, &m);
4     if (h < 12) {
5         printf("%02d:%02d AM\n", h, m);
6     } else if (h == 12) {
7         if (m == 0)
8             printf("%02d:%02d AM\n", h, m);
9         else
10            printf("%02d:%02d PM\n", h, m);
11    } else {
12        printf("%02d:%02d PM\n", h - 12, m);
13    }
14    return 0;
15 }
```

4.3 7-28 多分支表达-数据奇偶判断

```
1 int main() {
2     int a, b;
3     scanf("%d,%d", &a, &b);
4     if (a == 0 || b == 0)
5         return 0;
6     if (a % 2 == 0) {
7         if (b % 2 == 0)
8             printf("%d+%d=%d", a, b, a + b);
9         else
10            printf("%d/%d=%d", a, b, a / b);
11    } else {
12        if (b % 2 == 0)
13            printf("%d*d=%d", a, b, a * b);
14        else
15            printf("%d-%d=%d", a, b, a - b);
16    }
17    return 0;
18 }
```

5 利用数组

一部分题其实 `if - else` 或 `switch` 都能做，我只是觉得数组更适合。

5.1 7-17 帮小明出主意

可以 `if - else`，也可以 `switch`，我觉得数组比较好。

```
1 char s[][20] = {
2     "answer = First",
3     "answer = Second",
4     "answer = Third",
5     "answer = Fourth"
6 };
7
8 int main() {
9     int a;
10    scanf("%d", &a);
11    a = a % 4;
12    printf("%s", s[a]);
13    return 0;
14 }
```

5.2 7-23 根据输入的数字，输出需要上课的节数

```
1 char s[][80] = {
2     "",
3     "星期一 8节课",
4     "星期二 10节课",
5     "星期三 6节课",
6     "星期四 8节课",
7     "星期五 6节课",
8     "今天没有课，可以好好休息一下啦！ ",
9     "今天没有课，可以好好休息一下啦！ "
10 };
11
12 int main() {
13     int d;
14     scanf("%d", &d);
15     printf("%s", s[d]);
16     return 0;
17 }
```

5.3 7-30 输入一个数字，输出其对应的星期几的英文单词

```
1  char s[][10] = {
2      "",
3      "Monday",
4      "Tuesday",
5      "Wednesday",
6      "Thursday",
7      "Friday",
8      "Saturday",
9      "Sunday"
10 };
11
12 int main() {
13     int a;
14     scanf("%d", &a);
15     if (1 <= a && a <= 7) {
16         printf("%s", s[a]);
17     } else {
18         printf("输入错误! ");
19     }
20     return 0;
21 }
```

5.4 7-33 输出星期名

```
1  char s[][10] = {
2      "Sunday",
3      "Monday",
4      "Tuesday",
5      "Wednesday",
6      "Thursday",
7      "Friday",
8      "Saturday"
9  };
10
11 int main() {
12     int a;
13     scanf("%d", &a);
14     if (0 <= a && a <= 6) {
15         printf("%s", s[a]);
16     } else {
17         printf("None");
18     }
19 }
```

```
18     }  
19     return 0;  
20 }
```
