

A 北航70周年校庆

题目分析

打印字符串，考察 `printf` 的基本使用（以及换行符 `'\n'`）。

示例代码

```
1  #include<stdio.h>
2  int main()
3  {
4      printf("Hello, BUAA!\nThis year is your 70th birthday!\nHappy birthday!");
5      return 0;
6  }
```

B 星幽增强

题目分析

考察输入、加法、输出整数。

示例代码

```
1  #include <stdio.h>
2
3  int main(){
4      int x,y;
5      scanf("%d%d", &x, &y);
6      printf("%d", x+y);
7      return 0;
8  }
```

C 大运村寻家记

题目分析

- 1、不要直接使用 $(a * b) \% m$ 进行运算，因为 $a * b$ 可能会超过 `int` 范围，请根据以下数学公式进行优化。
- 2、利用公式： $(a * b) \% m = ((a \% m) * (b \% m)) \% m$ ，即在程序中输出表达式 $((a \% m) * (b \% m)) \% m$ 的值。

示例代码

```
1  #include <stdio.h>
2
3  int a, m, b;
4  int main(void) {
5      scanf("%d%d%d", &a, &m, &b);
6      printf("%d", ((a % m) * (b % m)) % m);
7      return 0;
8  }
```

D 投球游戏

题目分析

本题考察学生对 *if* 的掌握情况。只使用 *if* 或是使用 *if - else* 都是可以的。

如果是单个的 *if* 语句，则会判断 *if* 后括号内的表达式的值，如果为真则执行接下来的代码块。

如果是 *if - else* 结构：

if - else 语句执行的步骤如下：

1. 首先判断第一个 `if` 条件是否成立，如果成立则执行第一个代码块，执行完毕后跳过整个 *if - else* 代码块。
2. 如果不成立，则判断第二个条件是否成立，如果成立则执行第二个代码块，执行完毕后跳过整个 *if - else* 代码块。
3. 依次对剩余条件进行判断。如果有任意一个条件成立，那么执行完所属的代码后将不会执行 *if - else* 结构中剩余的代码。
4. 如果所有条件都不成立，则执行 `else` 代码块内的内容。
5. 如果所有条件都不成立且没有 `else` 部分，则不会执行任何代码。

如下所示：

```
1  if( /*判断条件1*/ ) {
2      ... //如果判断条件1成立则执行这里
3  }
4  else if( /*判断条件2*/ ) {
5      ... //如果判断条件2成立则执行这里
6  }
7  else {
8      ... //如果判断条件1和判断条件2都不成立则执行这里
9  }
10 //如果判断条件1和判断条件2都成立则只会执行代码块1内部的内容
```

示例代码

```
1  #include <stdio.h>
2
3  int main(){
4      int a;
5      scanf("%d", &a); //读入数据
6      if(a < 53)        //判断第一个条件
7          printf("Too close!\n");
8      else if(a == 53)  //判断第二个条件
9          printf("Bingo!\n");
10     else if(a > 53)    //判断第三个条件
11         printf("Too far!\n");
12     return 0;
13 }
```

E 搬书

题目分析

本题涉及主要考点是输入输出语句，和 `while` 循环语句，由PPT例题"1-8"改编而来。

由于这里我们无法提前预知一共有多少行输入，因此尽量不要使用 `for` 循环语句，用 `while` 循环更佳。

首先，设置三个变量：`sum`，`num`，`a` 分别用来存储书籍总数，书堆个数，某一个书堆中的书籍本数。

接着，先读入一个 `a`，使其具有非 0 的数值，才能成功进入开始 `while` 循环；在循环中不断累加 `sum` 和 `num`；循环最后再次读入 `a` 更新书堆中的书本个数。

最后，按顺序输出所求变量。

示例代码

```
1  #include<stdio.h>
2  int a,sum,num;
3  int main()
4  {
5      scanf("%d",&a);
6      while(a!=0)
7      {
8          num++;
9          sum+=a;
10         scanf("%d",&a);
11     }
12     printf("%d %d %d",sum,num,sum/num);
13     return 0;
14 }
```

F 不要迟到了！

题目分析

由题目中数据范围可以知道出发时间一定在当天00:00之后，因此考虑从00:00开始所经过的分钟数。

从00:00->18:55需要 $18 \times 60 + 55 = 1135$ 分钟，其中有三分钟需要倒垃圾，有 s/v 并向上取整分钟用来赶路，而向上取整的计算方式为 $(s + v - 1)/v$ ，因此出发时间距离00:00的分钟数为

$mins = 1135 - 3 - (s + v - 1)/v$ ，对应的 $h = mins/60, m = mins \% 60$ 。

示例代码

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int s,v;
6      scanf("%d%d",&s,&v);
7      int tot=1135-3-(s+v-1)/v;
8      printf("%02d:%02d",tot/60,tot%60);
9  }
```

G 打印字符

题目分析

本题考查如何使用循环语句。注意输出的第一行和第三行左侧有一个空格。

示例代码

使用 for 循环

```
1  #include<stdio.h>
2  int main()
3  {
4      int n, i;
5      scanf("%d", &n);
6      printf(" ");
7      for (i = 1; i <= n-2; i++)
8          printf("*");
9      printf("\n");//第一行输出结束
10     for (i = 1; i <= n; i++)
11         printf("*");
12     printf("\n");//第二行输出结束
```

```

13     printf(" ");
14     for (i = 1; i <= n-2; i++)
15         printf("*");
16     printf("\n");//第三行输出结束
17     return 0;
18 }

```

使用 while 循环

```

1  #include<stdio.h>
2  int main()
3  {
4      int n, i;
5      scanf("%d", &n);
6      i = 1;
7      printf(" ");
8      while (i <= n - 2) {
9          printf("*");
10         i++;
11     }
12     printf("\n");//第一行输出结束
13     i = 1;
14     while (i <= n) {
15         printf("*");
16         i++;
17     }
18     printf("\n");//第二行输出结束
19     i = 1;
20     printf(" ");
21     while (i <= n-2) {
22         printf("*");
23         i++;
24     }
25     printf("\n");//第三行输出结束
26     return 0;
27 }

```

H 最大公约数x3

题目分析

求三个数字的最大公约数。

不妨设 $mx = \max(a, b, c)$ ，最暴力的想法就是从 mx 开始向下枚举，一直枚举到 1，如果出现某个枚举项 i ，使得 $a \% i == 0 \&\& b \% i == 0 \&\& c \% i == 0$ ，那么 i 就是我们所求的最大公约数

更快速的方法可以考虑辗转相减法/辗转相除法来计算最大公约数，计算三个数的最大公约数，可以先取两个数字，计算出他们的 gcd ，再将这个 gcd 与第三个数字求 gcd 既可。

枚举法可以如下所示：

```

1 // gcd最初为a,b,c中最大的数字
2 while(gcd>=1){
3     if(a%gcd==0&&b%gcd==0&&c%gcd==0)break;
4     gcd--;
5 }

```

辗转相除法：

引理： $\gcd(a, b) = \gcd(b, a \% b)$

则我们可以写出如下代码来计算两个数字的最大公约数：

```

1 int a, b, gcd;
2 while(a&&b){
3     int tmp_a=b;
4     int tmp_b=a%b;
5     a=tmp_a;
6     b=tmp_b;
7 }
8 if(a==0)gcd=b;
9 else gcd=a;

```

示例代码

```

1 #include<stdio.h>
2 #include<string.h>
3 #include<stdlib.h>
4
5 int main(){
6     int a, b, c;
7     scanf("%d%d%d", &a, &b, &c);
8     int gcd=a;
9     if(gcd<b)gcd=b;
10    if(gcd<c)gcd=c;
11    while(gcd>=1){
12        if(a%gcd==0&&b%gcd==0&&c%gcd==0)break;
13        gcd--;
14    }
15    printf("%d", gcd);
16    return 0;
17 }

```

I 错误发生（Error Hassei）

题目分析

完成本题前，需要熟悉以下知识点：

- 如何定义 `int` 变量，并且如何输入？
- 如何使用判断语句？
- 如何使用循环语句？

- 如何输入字符串？
- 如何输出一个整数？

如何求上取整？

我们知道，两个 `int` 类型的整数 `x` 和 `y` 通过 `x/y` 即为下取整；

于是，不难证明，`(x+y-1)/y` 即是 `x` 除以 `y` 的上取整。

如以下代码可以计算 2 除以 3 的上取整：

```
1  #include <stdio.h>
2  int main(){
3      int x = 2, y = 3;
4      printf("%d", (x+y-1)/y);
5  }
```

如何输出多行？

通过循环控制和换行符 `'\n'`，如以下代码可以输出 100 行 `Darling`：

```
1  #include <stdio.h>
2  int main(){
3      for(int i = 0; i < 100; i++) printf("Darling\n");
4  }
```

示例代码

```
1  #include<stdio.h>
2
3  int main() {
4      // 第一步：定义变量 x、y，然后输入
5      int x, y;
6      scanf("%d%d", &x, &y);
7
8      // 第二步：判断语句，判断 y 是否为 0，以此执行不同命令
9      if (y != 0) {
10         // 输出整数（x 除以 y 的上取整）
11         printf("%d", (x + y - 1) / y);
12     }
13     else {
14         // 循环语句：循环 x 次，每次输出一个字符串
15         for (int i = 0; i < x; i++) {
16             printf("Error Hassei\n");
17         }
18         printf("System Breakdown");
19     }
20 }
```

J 繁琐的识别mini

题目分析

识别并计算由字符 `x` 组成的表达式。

步骤：

1. 按照空列把每个字符分开。
2. 逐一比对每一段是哪个字符，并写出表达式。(慢慢写就行)
3. 按照加号和减号分开，然后加起来。

示例代码

```
1  #include <math.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <string.h>
5  #define LL long long
6  #define min(a, b) (a > b ? b : a)
7  #define max(a, b) (a < b ? b : a)
8
9  char s[10][200]; //map
10 int count[200]; //number of x on each rows
11
12 int c2[20] = {0, 5, 0, 0, 9, 0, 0, 7, 13};
13
14 int main() {
15     for (int i = 1; i <= 5; i++) //input
16         if (gets(s[i]) > 0);
17
18     for (int i = 1; i <= 5; i++)
19         for (int j = 0; j <= 160; j++)
20             if (s[i][j] == 'x')
21                 count[j]++;
22
23     int l = 0, u; //left;up
24
25     char formula[50] = {};
26     int top_of_formula = 0;
27     while (l <= 160) {
28         if (count[l] == 0) { //no match
29             l += 1;
30             continue;
31         }
32         if (count[l + 1]) {
33             int total = 0;
34             for (int i = 0; i <= 2; i++)
35                 total += count[l + i];
36             int line[10] = {}, line_with_sth = 0;
37             for (int i = 1; i <= 5; i++)
38                 for (int j = 0; j <= 2; j++)
39                     line[i] += (s[i][j + l] == 'x');
40             for (int i = 1; i <= 5; i++)
41                 line_with_sth += (line[i] != 0);
42             for (int i = 1; i <= 5; i++) {
43                 if (line[i] > 0) {
```



```

44         u = i;
45         break;
46     }
47 }
48
49 int this_char = 0;
50 if (line_with_sth == 1) {
51     this_char = '-';
52 }
53 if (line_with_sth == 3) {
54     this_char = '+';
55 }
56 if (line_with_sth == 5)
57     for (int i = 1; i <= 9; i++)
58         if (total == c2[i])
59             this_char = i + 48;
60 if (this_char == 0) {
61     char ur = s[u + 1][l + 2];
62     char lb = s[u + 3][l];
63     if (total == 12) {
64         int tmp = 2 * (ur == 'x') + (lb == 'x');
65         if (tmp == 1) this_char = '6';
66         if (tmp == 2) this_char = '9';
67         if (tmp == 3) this_char = '0';
68     }
69     if (total == 11) {
70         int tmp = (ur == 'x') + (lb == 'x');
71         if (tmp == 0) this_char = '5';
72         if (tmp == 1) this_char = '3';
73         if (tmp == 2) this_char = '2';
74     }
75 }
76 formula[++top_of_formula] = this_char;
77 l += 3;
78 continue;
79 }
80 if (count[l + 1] == 0) {
81     formula[++top_of_formula] = 49;
82     l += 1;
83     continue;
84 }
85 }
86
87 LL ans = 0, cur = 0, sgn = 1; //caculate
88
89 for (int i = 1; i <= 49; i++) {
90     if (formula[i] == 0) break;
91     if (formula[i] == '+') ans += sgn * cur, cur = 0, sgn = 1;
92     else if (formula[i] == '-') ans += sgn * cur, cur = 0, sgn = -1;
93     else cur = cur * 10 + formula[i] - 48;
94 }
95 ans += sgn * cur, cur = 0, sgn = 1;
96 printf("%lld", ans);
97
98 return 0;
99 }

```

