

while 循环结构

2022 年 8 月 16 日

1. 随机输入若干个 1-10(包含端点) 之间的整数，当随机数为 8 时，输入结束。输出这个过程中产生地随机数。

```
1  #include <stdlib.h> // 包含随机数函数的库
2  #include <time.h> // 包含时间操作的库
3  #include <stdio.h>
4  int main()
5  {
6      srand(time(NULL)); // 布置随机数种子
7      int n = 0; //
8      printf("Below is a random progression:\n");
9      do{
10         n = rand() % 10 + 1; // rand()返回一个整数，可能是很大的整数
11         // %进行取余运算之后，得到的值的范围在0-9，加1后，得到的范围在1-10
12         printf("%d\t", n);
13     }while(n != 8); // 只要n!=8，就会一直循环；n=8时退出
14
15     return 0;
16 }
```

2. 求一个输入正整数各个数位上的数字之和。

```
1  #include <stdio.h>
2  int main()
3  {
4      int input_num = 0;
5      printf("Input a natural number:\n");
6      scanf("%d", &input_num);
7  }
```

```

8     int sum = 0;
9     while(input_num) // C语言用非0值表示真，用0表示假
10    {
11        int digit = input_num % 10; // 取出最低位
12        sum += digit; //
13        input_num /= 10; // 去掉末位
14    }
15    printf("The sum of the input number is: %d.\n", sum);
16 }

```

3. 已知 2021 年 1 月 1 日是星期五，请问，从该天往后，下一个“1 月 1 日为星期五”的年份是哪个？
-

```

1  #include <stdio.h>
2  int main()
3  {
4      // 每经过一个闰年，就会在经过整数周之后，多出2天
5      int remainder_in_leap_years = 2;
6      // 每经过普通的一年，就会在经过整数周之后，多出1天
7      int remainder_in_other_years = 1;
8      int year = 2022; // 第一个测试的年份
9      int delay = 1; // 第一个测试的年份里推迟了1月1日的星期日子推迟1天
10     // 以下delay%7就是delay%7!=0的意思，非0值都可以表示真
11     while(delay % 7) // 推迟了7天，也就跟原来星期数相同
12     {
13         if(year % 4) // 不被4整除的不是闰年
14         {
15             delay += remainder_in_other_years; // 推迟非闰年多出的天数
16         }
17         else // 被4整除是闰年
18         {
19             delay += remainder_in_leap_years; // 推迟闰年多出的天数
20         }
21         year++;
22     }
23     if(year < 2100) // 考虑下一年
24     {
25         printf("year: %d\n", year);
26     }

```

```

27     else
28     {
29         printf("program incorrect\n");
30     }
31     return 0;
32 }

```

4. 有一口井，它的深度为正整数，且由用户输入。一只蜗牛在某一天刚刚天亮时，从井底出发往上爬。已知它白天能往上爬 5 米，晚上滑落 3 米，那么，经过多少天，它能爬到井外？（答案须为整数。）

```

1  #include <stdio.h>
2  int main()
3  {
4      int well_depth;
5      printf("Input the depth of a well:\n");
6      scanf("%d", &well_depth);
7      int height = 0;
8      int day_num = 0;
9      while(1)
10     {
11         day_num++;
12         height += 5;
13         printf("5 meters upward\n");
14         if(height >= well_depth)
15         {
16             break;
17         }
18         height -= 3;
19         printf("3 meters downward\n");
20         printf("height achieved = %d\n", height);
21     }
22     printf("days required: %d", day_num);
23     return 0;
24 }

```

5. 开发一个猜数字的小游戏，计算机在 0~2 之间随机产生一个数，并提示用户按照这个范围去猜（因此当数字给定之后，用户有 1/3 的概率猜中）。要求：

- (a) 如果用户的猜测不在 0~2 范围内, 就再次请求用户进行猜测, 直到用户做出的猜测在这个范围内为止;
- (b) 如果用户猜中了, 则告知用户猜中, 否则, 须告知用户待猜的数字是什么;
- (c) 无论用户是否猜中, 须询问用户是否继续进行游戏, 此时用户只允许输入 y 或者 n, 如果用户输入其他字符, 则再次请求用户进行输入, 直到用户的输入符合这个要求。

```
1  #include <stdio.h>
2  #include <time.h>
3  #include <stdlib.h>
4  int main()
5  {
6      srand(time(NULL));
7      while(1)
8      {
9          int magic_num = rand() % 3; // 产生待猜测的数字 (0~2范围内)
10         int guess_num;
11
12         do
13         {
14             printf("make a guess (0-2):\n"); // 提示用户做出猜测
15             scanf("%d", &guess_num); // 获取用户的猜测
16         } while (guess_num < 0 || guess_num > 2); // 用户的猜测不在合理范围内
17
18         if(guess_num == magic_num) // 猜中
19         {
20             printf("correct\n");
21         }
22         else // 猜不中
23         {
24             printf("sorry, the magic number is %d\n", magic_num);
25         }
26
27         char ch; // 用户获取用户关于是否继续游戏的输入
28         do
29         {
30             getchar(); // 前面输入的时候, 多出一个回车未处理, 这里getchar()吸收了该回车输入
31             printf("like to continue? (y/n)\n"); // 提示用户输入要不要继续玩
32             scanf("%c", &ch); // 获取用户的输入
```

```

33     } while (ch != 'y' && ch != 'n'); // 只接受'y'和'n'两种输入
34     if(ch == 'n') // 用户不想继续玩
35     {
36         break; // 退出循环
37     }
38 }
39 }

```

6. 开发一个加法答题小游戏，要求程序产生若干加法式子，用户作答，并在最后输出总分。具体要求如下：

- (a) 加法式子是两个数相加的形式，并且两个加数都是 1~9 之间的整数；
 - (b) 游戏总分为 100 分，一共 10 个加法题目，各个题目的分值相同；
 - (c) 如果用户答对，要提示用户答对，否则要给出正确答案；
 - (d) 如果用户答对，就得到该题的分值，否则就不得分；
 - (e) 游戏结束之后，输出用户的总分。
-

```

1  #include <stdio.h>
2  #include <time.h>
3  #include <stdlib.h>
4  const int QUESTION_COUNT = 10; // 题量
5  const int FULL_MARK = 100; // 满分
6  int main()
7  {
8      srand(time(NULL));
9      int a, b;
10     int answer;
11     int remaining_question_count = QUESTION_COUNT;
12     int total_score = 0;
13     int score_per_question = FULL_MARK / QUESTION_COUNT; // 每道题目得分
14     do{
15         a = rand() % 9 + 1;
16         b = rand() % 9 + 1;
17         printf("%d + %d = ?\n", a, b);
18         scanf("%d", &answer);
19         if(answer == a + b) // 答对
20         {

```

```
21     printf("correct\n");
22     total_score += score_per_question; // 得分
23 }
24 else // 答错
25 {
26     printf("sorry, the correct answer should be %d\n", a + b);
27 }
28 remaining_question_count--; // 需要回答的问题数减少
29 }while(remaining_question_count); // 仍有未答问题
30 printf("finally, your total score is %d\n", total_score);
31 return 0;
32 }
```
