

CS111, C Programming Lab / Condition & Loop

黄嘉炜

huangjw3@mail.sustech.edu.cn



Outline



- Review
- Condition: Showcase, switch
- Loop: Showcase, while
- Assignment



Review: Problem #1



```
1 long long aa,bb,cc;
2 double A;
3 scanf("%lld %lld %lld",&aa,&bb,&cc);
4 printf("%lld",(bb*bb-4*aa*cc));
5 double a=aa,b=bb,c=cc;
6 long long aa,bb,cc;
7 scanf("%lld %lld %lld",&aa,&bb,&cc);
8 printf("%lld",(bb*bb-4*aa*cc));
9 double a=aa,b=bb,c=cc;
10 if(a==0)
11 {
12 printf("%.5lf",(-c/b));
13 }
14 else
Bug?
```



Review: Problem #2

```
int cc0;
int cc1;
int cc2;
int cc2;
int cc3;
scanf("%d %d",&msg,&key);
```

```
cc2= a_2+39;
         if(a_0<=25){
38
                  cc3= a_3 +97;//转为ASCII码
         else{
                  cc3= a_3+39;
         printf("%c%c%c%c",cc0,cc1,cc2,cc3
```



思考: 这两个 cc2

- ▶ 是同个变量吗?
- ▶ 重复定义?
- ▶ 最后输出是哪个变量?

Not Sure? Let's debug it...

How to debug in VS Code? 关键点



> 添加/删除断点

注意:需要在非运行状态操作

```
C lab2_showcase_debugger.c X
C lab2_showcase_debugger.c > 分 main()
       #include <stdio.h>
      int main()
          int c0 = 0;
 单击以添加断点。 Fset = 0;
          scant("%d", &offset);
          if (offset >= 26) {
               char c0 = 'A' + offset;
 10
 11
          else {
               char c0 = 'a' + offset;
 12
 13
 14
          printf("input offset: %d, c0 %c\n", offset, c0);
 15
          return 0;
 16
```

```
lab2 showcase debugger.c X
C lab2_showcase_debugger.c > 分 main()
       #include <stdio.h>
       int main()
          int c0 = 0;
          int offset = 0;
          coanf/"od" oaffset);
   删除断点
                 Delete
                          A' + offset;
   编辑 断点...
   禁用断点
 12
               char c0 = 'a' + offset;
 13
 14
          printf("input offset: %d, c0 %c\n", offset, c0);
 15
          return 0;
 16
```

How to debug in VS Code? 关键点







- ➤ 添加/删除断点
- ▶ 启动调试:继续、逐过程、单步、跳出、

```
C lab2_showcase_debugger.c X
 C lab2_showcase_debugger.c > 分 main()
       #include <stdio.h>
       void doSomething(){
            puts("doSomething");
       int main()
          int c0 = 0;
          int offset = 0;
          doSomething();
11
 12
          scanf("%d", &offset);
 13
          if (offset >= 26) {
 14
               char c0 = 'A' + offset;
          else {
 17
               char c0 = 'a' + offset;
          printf("input offset: %d, c0 %c\n", offset, c0);
          return 0;
 20
 21
```

How to debug in VS Code? 关键点



运行和测试 D gcc.exe - 生 V 〇 ··· lab2 showcase debugger.c × ∨ 安田 C lab2_showcase_debugger.c > @ main() #include <stdio.h> tocals c0: 0 void doSomething(){ offset: 26 puts("doSomething"); > Registers int main() - 監視 int $c\theta = 0$: int offset = 0; doSomething(); scanf("%d", &offset); D 13 if (offset >= 26) { char c0 = 'A' + (offset - 26); 调用堆栈 printf("inside if, c0 %c\n", c0); Thread #1 因 STEP 已暂停 else (main() lab2_showcase_debugge. char c0 = 'a' + offset; > Thread #2 printf("inside else, c0 %c\n", c0);

- >添加/删除断点
- ▶ 启动调试:继续、逐过程、单步、跳出、...
- ▶ 查看信息: 变量、调用堆栈

```
运行和调试 D gcc.exe - 生 > ۞ …
                                     lab2_showcase_debugger.c ×
                                      C lab2 showcase debugger.c > 😭 main()
∨ 変量
                                            #include <stdio.h>

∨ Locals

    c0: 65 'A'
                                            void doSomething(){
    c0 #2: 65 'A'
                                                  puts("doSomething");
    offset: 26
 > Registers
                                            int main()
 监视
                                                int c\theta = \theta:
                                                int offset = 0;
                                               doSomething();
                                                scanf("%d", &offset);
                                                if (offset >= 26) {
                                                     char c0 = 'A' + (offset - 26);
~ 调用堆栈
                                   D 15
                                                     printf("inside if, c0 %c\n", c0);
∨ Thread #1
                       因 STEP 已暂停
                                                else {
    main() lab2 showcase debugge...
                                                     char c0 = 'a' + offset;
> Thread #2
                                                     printf("inside else, c0 %c\n", c0);
```

How to debug in VS Code?

运行和调试 **D** gcc.exe - 生 ∨ ⑫ … lab2_showcase_debugger.c X C lab2_showcase_debugger.c > 😭 main() #include (stdio.h> ∨ Locals void doSomething(){ offset: 26 puts("doSomething"); Registers int main() 监视 int $c\theta = \theta$; int offset = 0; doSomething(); scanf("%d", &offset); if (offset >= 26) { char c0 = 'A' + (offset - 26); 调用维栈 printf("inside if, c0 %c\n", c0); Thread #1 因 STEP 已暂停 else { main() lab2 showcase debugge... char c0 = 'a' + offset; > Thread #2 printf("inside else, c0 %c\n", c0); printf("input offset: %d, c0 %c\n", offset, c0); D 22

关键点

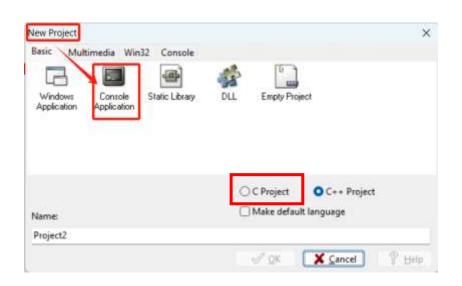


- >添加/删除断点
- ▶ 启动调试:继续、逐过程、单步、跳出、...
- ▶ 查看信息: 变量、调用堆栈

```
doSomething
26
inside if, c0 A
input offset: 26, c0
```

How to debug in Dev-C++ 关键点





注意:

Dev-C++需针对 debug 代码创建一个 project, 然后把待调试代码放到project中

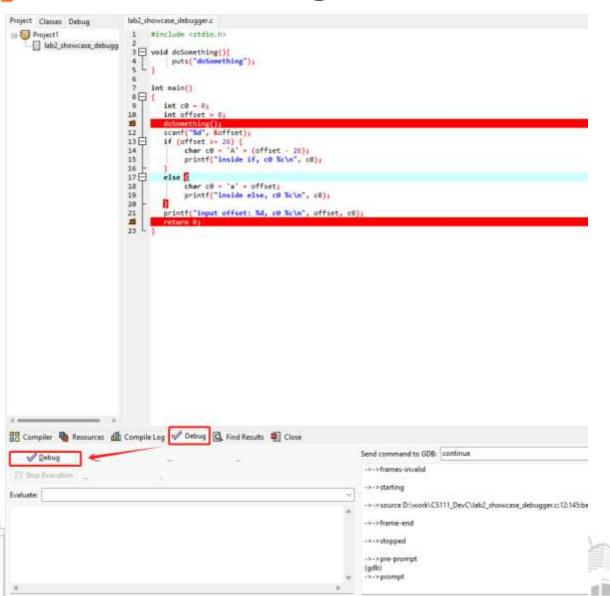
▶ 添加/删除断点

▶ 启动调试:继续、逐过程、单步、跳出、...

▶ 查看信息: 变量、调用堆栈

```
lab2_showcase_debugger.c
Project Classes Debug
Project1
                                  #include <stdio.h>
     lab2_showcase_debugg
                              3 - void doSomething(){
                                       puts("doSomething");
                                  int main()
                                     int c0 = 0;
                                     int offset = 0;
                                     doSomething();
                                     scanf("%d", &offset);
                                     if (offset >= 26) {
                                          char c0 = 'A' + (offset - 26);
                                          printf("inside if, c0 %c\n", c0);
                            17 -
                                     else [
                                          char c0 = 'a' + offset;
                                          printf("inside else, c0 %c\n", c0);
                             19
                             20
                                     printf("input offset: %d, c0 %c\n", offset, c0);
```

How to debug in Dev-C++



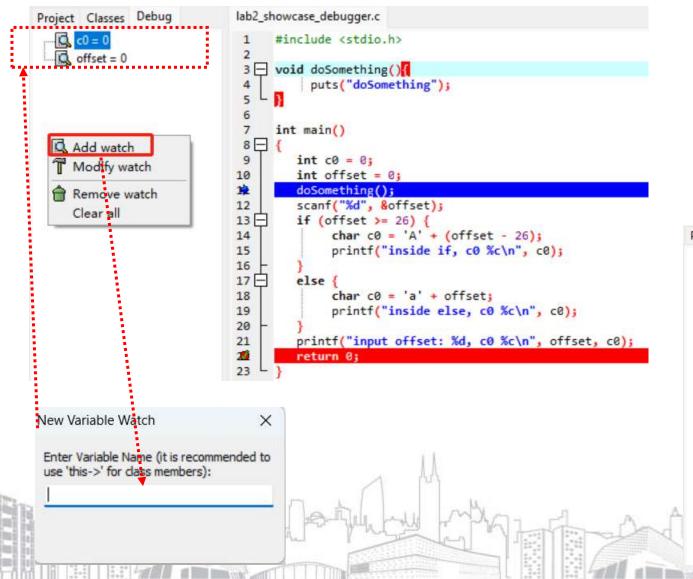
关键点



- ➤ 添加/删除断点
- ▶ 启动调试:继续、逐过程、单步、跳出、.

```
lab2_chowcase_debugger.c
                                     #include cutting.htm
                                 3 [] void doScenthing()(
                                           puts("doSomething");
                                       1f (offset >= 20) {
    char c0 = 'A' = (offset - 20);
                                              printf("inside if, c0 %c'u", c0);
                                               char co - 'a' - offset;
                                          printf("input offset; %d, c8 %c'u", offset, c8)
Compiler & Resources of Compile Log V Detrog & Find Hessits A Close
                                                                                               Send command to GDB: break "Di/work/CS111_DevC/lab2_showcase_debugger.c";22
 26 Stop Execution Year CPU window
                                                           Skip function
                                                                             Into instruction
                                                                                                -> -> source Drimorki,CS111_DevCrish2_showcme_debugges.ci11:126/freg:0x401566
                                                                                                ->->-frame-end
                                                                                                -5-3 gré-prompt
                                                                                                49-3 prompt
```

How to debug in Dev-C++



关键点



- ➤ 添加/删除断点
- ▶ 启动调试:继续、逐过程、单步、跳出、...
- ▶ 查看信息: 变量、调用堆栈

```
Project Classes Debug
                             lab2_showcase_debugger.c
  c0 = 0
                                  #include <stdio.h>
  offset = 26
                                  void doSomething(){
                                       puts("doSomething");
                                  int main()
                                     int c0 = 0;
                                     int offset = 0
                                     doSomething():
                                     scanf("%d", &offset);
                                     if (offset >= 26)
                                          char c0 = 'A' + (offset - 26);
                             15
                                          printf("inside if, c0 %c\n", c0);
                             16
                            17 -
                                     else
                                          char c0 = 'a' + offset;
                             18
                                          printf("inside else, c0 %c\n", c0);
                             19
                             20
                             21
                                     printf("input offset: %d, c0 %c\n", offset, c0);
                                     return 0;
```

Outline



- Review
- Condition: Showcase, switch
- Loop: Showcase, while
- Assignment



Condition: Review, if ... else ...

How to re-write as "switch ... case ..."?

```
int a = 0;
         scanf("%d", &a);
 6
         if (a == 13)
 8
              puts("Card Number: K");
10
         else if (a == 12)
11
12
13
              puts("Card Number: Q");
14
         else if (a == 11)
15
16
              puts("Card Number: J");
17
18
         else if (a == 1)
19
20
              puts("Card Number: A");
21
22
23
         else if (a > 1 && a <= 10)
24
              printf("Card Number: %d\n", a);
25
26
         else
27
28
29
              puts("Error Card Number!");
          puts("run after if...else... block");
31
         return 0;
```

Condition: Showcase, switch

```
SUSTech Southern University of Science and Technology
```

```
14
Error Card Number!
```

Bugs?

```
Card Number: 3
run after switch..case... block
```

```
int a = 0;
          scanf("%d", &a);
 6
         if (a <= 0 | a > 13)
 8
              puts("Error Card Number!");
 9
10
              return 0;
11
          switch (a)
12
13
14
              case 13:
                  puts("Card Number: K");
15
16
              case 12:
17
                  puts("Card Number: Q");
18
              case 11:
                  puts("Card Number: J");
19
20
              case 1:
                  puts("Card Number: A");
21
              default:
22
                  printf("Card Number: %d\n", a);
23
24
          puts("run after switch..case... block");
25
26
          return 0;
```

Outline



- Review
- Condition: Showcase, switch
- Loop: Showcase, while
- Assignment



Loop: Showcase, while



••••

Results 2 and 3: Solve the quadratic equation $ax^2 + bx + c = 0$ and output the two solutions as double-precision floating-point numbers (保留5位小数).

>

➤ Note: For square root operations, include the <math.h> header file and utilize the sqrt() function.

扩展思考:尝试不使用 math.h 中 sqrt 函数;采用二分法近似求解出平方根,中间语法需要

用到循环(e.g. while) 语句。

Let's coding together?

double my_sqrt(double val)

Loop: Show

Need a test code framework?

```
double my_sqrt(double val) {
         return 0.0; // TODO
     int main()
 9
10
         int pass = 0, fail = 0;
         double x = 0;
11
12
         while(1) {
13
             // step1, check input, when error input, break
14
             scanf("%lf", &x);
             if (x < 0) {
15
16
                  printf("error input: %lf, exit\n", x);;
17
                 break;
18
19
              // step2, check result, and check difference
             double my result = my sqrt(x);
20
21
             double diff = abs(my_result * my_result - x);
22
             if (diff < 0.000001) {
23
                  pass++;
                  printf("Passed: x = %.61f, result %.61f, Continue to test\n", x, my_result);
24
25
                  continue;
26
             fail++;
27
             printf("Failed: x = %.61f, result %.61f, Error %.61f\n", x, my_result, diff);
28
29
         printf("Test finished, pass %d, fail %d\n", pass, fail);
31
         return 0;
32
```

Outline



- Review
- Condition: Showcase, switch
- Loop: Showcase, for
- Assignment



Assignment 1)

输入2个整数,求最大公约数



Given two integers a and b ($1 \le a, b \le 2 \times 10^9$), your task is to find the **greatest common divisor (GCD)** of the two integers, which is the greatest integer that divides both of them.

Hint

Here are the details of the algorithm. Suppose $a \geq b$ (note that the data does NOT guarantee this!)

- 0. Let $r = a \mod b$
- 1. Loop
 - \circ If r=0, break the loop.
 - \circ Otherwise, we have $r \neq 0$. Move b's value into a, then move r's value into b, finally calculate $r = a \mod b$.
- 2. One of a, b, r holds the answer. Who? \otimes

<u>24</u> 18 6 66 44

22

189 46

1

Assignment 2)

输入2个日期, 计算这两个日期的相差多少天?



Given two dates d_1 and d_2 of format YYYY-MM-DD, calculate the absolute differential days between them.

It is guaranteed that both d_1 and d_2 are valid dates between 1001-01-01 and 2024-12-31.

Format

Input

There are 2 lines in the input.

The first line contains a string of format YYYY-MM-DD representing d_1 .

The second line contains a string of format $\begin{subarray}{c} \begin{subarray}{c} \begin{subarray}{c}$

Output

Print the absolute differential days between d_1 and d_2 .

2024-02-19 2024-03-21 31 2020-03-01 2020-02-28

Hint

Remember to consider the leap year. You can check your implementation in one of the previous labs.

You can use the following code to read the date string and obtain three integers representing the date. (This is where you need to add – in the format string!)

```
int year, month, day;
scanf("%d-%d-%d", &year, &month, &day);
```

Subtasks NOTE: 本道题在OJ拿到60+分,即可算满分

This problem is kind of challenging. If you cannot finish the whole problem, we have prepared some subtasks with simpler situations:

- Subtask 1: 10 pts for successfully handling the above 4 samples.
- Subtask 2: Another 30 pts for handling the situation where d₁, d₂ are dates in the same month of year 2020.
- Subtask 3: Another 30 pts for handling the situation where d_1, d_2 are dates within the same year.

The final 30 pts is for the full solution to the problem.



THANK YOU