

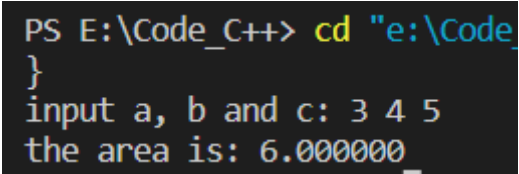
习题 1, 2, 6, 7, 10, 11

Ex1

源代码

```
1  #include <stdio.h>
2  #include <math.h>
3
4  double getArea(double a, double b, double c) {
5      double p = (a + b + c) / 2;
6      return sqrt(p * (p - a) * (p - b) * (p - c));
7  }
8
9  int main(){
10
11     double a, b, c;
12
13     printf("input a, b and c: ");
14     scanf("%lf%lf%lf", &a, &b, &c);
15     printf("the area is: %lf", getArea(a, b, c));
16
17     return 0;
18 }
```

运行结果



```
PS E:\Code_C++> cd "e:\Code_
}
input a, b and c: 3 4 5
the area is: 6.000000
```

Ex2

源代码

```
1  #include <stdio.h>
2
3  double p(int m, int n) {
4      if (n == 1)
5          return m;
6      return m * p(m - 1, n - 1);
7  }
8
9  int main() {
10
11     int m, n;
12
13     printf("input m and n: ");
14     scanf("%d%d", &m, &n);
15     printf("the factorial is: %lf", p(m, n));
16
17     return 0;
18 }
```

运行结果

```
PS E:\Code_C++\hw07> cd "e:\Code_C++\Ex2"
input m and n: 6 4
the factorial is: 360.000000
```

Ex6

源代码

```
1  #include <stdio.h>
2
3  int fib(int n) {
4      if (n <= 2)
5          return 1;
6      return fib(n - 1) + fib(n - 2);
7  }
8
9  int main() {
10
11      int n;
12
13      printf("input n: ");
14      scanf("%d", &n);
15      printf("the n_th fibonacci number is: %d", fib(n));
16
17      return 0;
18 }
```

运行结果

```
PS E:\Code_C++\hw07> cd "e:\Code_C++\Ex6"
input n: 10
the n th fibonacci number is: 55
```

Ex7

源代码

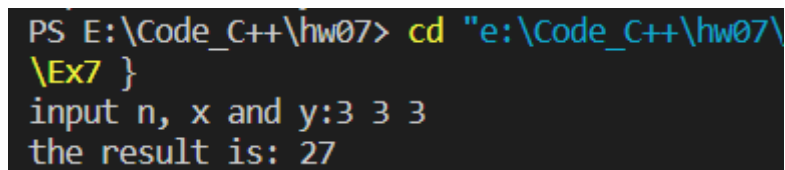
```
1  #include <stdio.h>
2
3  int Ack(int n, int x, int y) {
4      if (n == 0)
5          return x + 1;
6      else if (y == 0)
7      {
8          if (n == 1)
9              return x;
10         else if (n == 2)
11             return 0;
12         else if (n == 3)
13             return 1;
14         else /*(n >= 4)*/
```

```

15         return 2;
16     }
17     else /*(n != 0 && y != 0)*/
18         return Ack(n - 1, Ack(n, x, y - 1), x);
19 }
20
21 int main() {
22
23     int n, x, y;
24     printf("input n, x and y:");
25     scanf("%d%d%d", &n, &x, &y);
26     printf("the result is: %d", Ack(n, x, y));
27
28     return 0;
29 }

```

运行结果



```

PS E:\Code_C++\hw07> cd "e:\Code_C++\hw07\Ex7"
input n, x and y:3 3 3
the result is: 27

```

Ex10

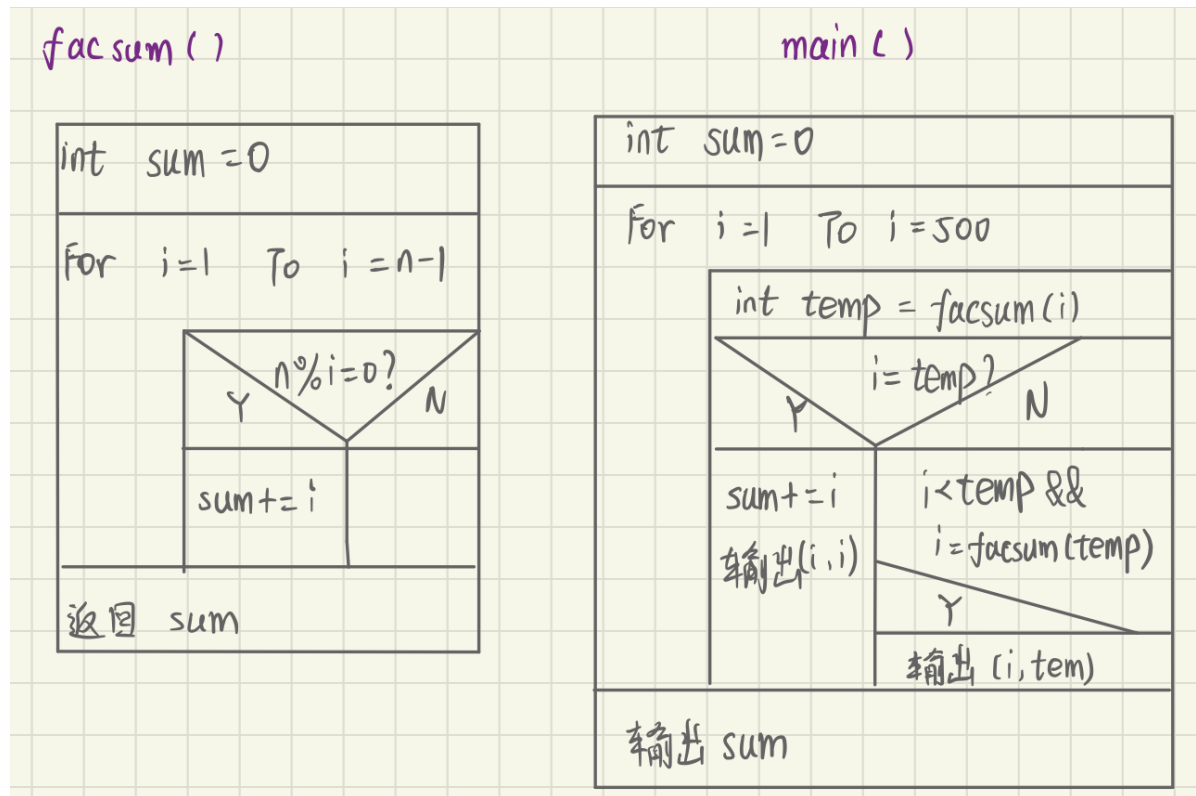
源代码

```

1  #include <stdio.h>
2
3  int facsum(int n) {
4      int sum = 0;
5      for (int i = 1; i < n; i++) {
6          if (n % i == 0)
7              sum += i;
8      }
9      return sum;
10 }
11
12 int main() {
13
14     int sum = 0;
15
16     printf("各对“亲密数”： ");
17     for (int i = 1; i <= 500; i++) {
18         int temp = facsum(i);
19         if (i == temp) {
20             sum += i;
21             printf("(%d, %d)", i, i);           // 完数也是亲密数，也要输出
22         }
23         else if (i < temp && i == facsum(temp)) { // 通过小于判断重复输出，还能
排序
24             printf("(%d, %d)", i, temp);
25         }
26     }
27     printf("\n“完数之和”： %d\n", sum);
28
29     return 0;

```

结构化流程图



运行结果

```
PS E:\Code_C++\hw07> cd "e:\Code_C++\hw07\" ; if ($?) { gcc
.\Ex10 }
各对“亲密数”: (6, 6)(28, 28)(220, 284)(496, 496)
“完数之和”: 530
PS E:\Code_C++\hw07> 
```

Ex11

源代码

```
1  #include <stdio.h>
2
3  int frac(int k) {
4      if (k < 0) {
5          printf("invalid value\n");
6          return 0;
7      }
8      if (k == 1 || k == 0)           // 注意0的阶乘为1
9          return 1;
10     return k * frac(k - 1);
11 }
12
13 int main() {
14
15     int m, n;
16     long long sum;
17
18     for (int i = 0; i < 5; i++) {
```

```

19     printf("input m and n: ");
20     scanf("%d%d", &m, &n);
21
22     if (n < m || m < 0) {
23         printf("invalid value\n");
24         continue;
25     }
26
27     sum = 0;
28     for (int j = m; j <= n; j++) {
29         sum += frac(j);
30     }
31
32     printf("the result is: %lld\n", sum);
33 }
34
35 return 0;
36 }

```

运行结果

```

PS E:\Code_C++\hw07> cd "e:\Code_C++\hw07\"
.\Ex11 }
input m and n: -3 7
invalid value
input m and n: 0 0
the result is: 1
input m and n: 1 7
the result is: 5913
input m and n: 9 13
the result is: 2454963584
input m and n: 9 4
invalid value
PS E:\Code_C++\hw07>

```