**实验三 问题四分析**

代码如下：

#include <stdio.h>

#define min 0

#define max 1000

int main()

{

1 int i=0,input=0,valid=0,value[100],sum=0,average;

2 printf("Input:(-999 for exit)");

3 scanf("%d",&value[i]);

4 while ((value[i]!=-999)&&(input<100))

{

5 ++input;

6 if ((value[i]>=min)&&(value[i]<=max))

{

7 ++valid;

8 sum=sum+value[i];

}

9 ++i;

10 scanf("%d",&value[i]);

}

1. if (valid>0)

12 average=sum/valid;

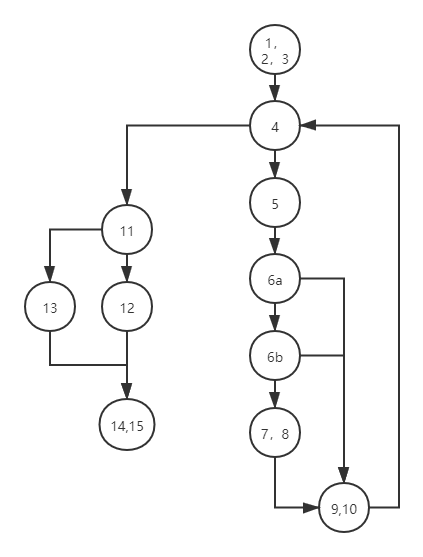
13 else average=-999;

14 printf("average=%d,sum=%d,number=%d\n",average,sum,valid);

15 return 0;

}

可以画出程序流图如下：



根据控制流图可以算出图的环路复杂度：

V(G)=封闭区域数=判定结点数+1=5

根据分析，如果程序没有进入while循环，则一定不会经过11-12这条路径，因此路径1-2-3-4-11-12-14-15不需要考虑；如果进入了while循环但一次都没有执行if判断，则一定也不会经过11-12这条路径，因此路径1-2-3-4-5-6-9-10-4-11-12-14-15不需要考虑；如果经历过while循环里面的if判断，则一定不会经过11-13这条路径，因此路径1-2-3-4-5-6-7-8-9-10-4-11-13-14-15不需要考虑。根据上面分析可以在基本路径集中导出以下有效的独立路径：

1-2-3-4-11-13-14-15

1-2-3-4-5-6-9-10-4-11-13-14-15

1-2-3-4-5-6-7-8-9-10-4-11-12-14-15

因此可以设计以下测试用例

|  |  |  |
| --- | --- | --- |
| 测试用例 | 执行路径 | 预期结果 |
| -999 | 1-2-3-4-11-13-14-15 | Average=-999,sum=0,num=0 |
| -1,-2,-3,-999 | 1-2-3-4-5-6-9-10-4-11-13-14-15 | Average=-999,sum=0,num=0 |
| 1,2,3,-4,-999 | 1-2-3-4-5-6-7-8-9-10-4-11-12-14-15 | Average=2,sum=6,num=3 |
| 1,2,3.5,-999 | 1-2-3-4-5-6-7-8-9-10-4-11-12-14-15 | Average=2.167,sum=6.5,num=3 |