形式化方法 实验小作业4（作业3） Frama-c

1. 代码

使用了一段冒泡排序的代码

```c
#include <stdio.h>

void bubbleSort(int arr[], int n) {
    int i, j, temp;
    for (i = 0; i < n-1; i++) {
        for (j = 0; j < n-i-1; j++) {
            if (arr[j] > arr[j+1]) {
                // 交换元素
                temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}

// 打印数组的函数
void printArray(int arr[], int n) {
    int i;
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main() {
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr)/sizeof(arr[0]);

    printf("排序前的数组: ");
    printArray(arr, n);

    bubbleSort(arr, n);

    printf("排序后的数组: ");
    printArray(arr, n);

    return 0;
}

```

2.运行结果

通过分析可以清楚的知道每个值的范围和情况，且得出的报告显示该程序是有效且无漏洞的

```
vixic@ViXiC:/mnt/d/homework/Formal_Methods/lab4$ frama-c -eva loop.c
[kernel] Parsing loop.c (with preprocessing)
[eva] Analyzing a complete application starting at main
[eva:initial-state] Values of globals at initialization

[eva] using specification for function printf_va_3
[eva] using specification for function printf_va_1
[eva] loop.c:20: starting to merge loop iterations
[eva] using specification for function printf_va_2
[eva] loop.c:6: starting to merge loop iterations
[eva] loop.c:5: starting to merge loop iterations
[eva] using specification for function printf_va_4
[eva] ====== VALUES COMPUTED ======
[eva:final-states] Values at end of function bubbleSort:
  i ∈ [6..2147483647]
  j ∈ [1..2147483647] or UNINITIALIZED
  temp ∈ [1..64] or UNINITIALIZED
  arr[0..5] ∈ {11; 12; 22; 25; 34; 64}
     [6] ∈ {90}
[eva:final-states] Values at end of function printArray:
  i ∈ [7..2147483647]
  S___fc_stdout[0..1] ∈ [--..--]
[eva:final-states] Values at end of function main:
  arr[0..5] ∈ {11; 12; 22; 25; 34; 64}
     [6] ∈ {90}
  n ∈ {7}
  __retres ∈ {0}
  S___fc_stdout[0..1] ∈ [--..--]
[eva:summary] ====== ANALYSIS SUMMARY ======
  ------------------------------------------------------------------------------
  3 functions analyzed (out of 3): 100% coverage.
  In these functions, 35 statements reached (out of 35): 100% coverage.
  ------------------------------------------------------------------------------
  No errors or warnings raised during the analysis.
  ------------------------------------------------------------------------------
  0 alarms generated by the analysis.
  ------------------------------------------------------------------------------
  Evaluation of the logical properties reached by the analysis:
    Assertions        0 valid     0 unknown     0 invalid      0 total
    Preconditions     4 valid     0 unknown     0 invalid      4 total
  100% of the logical properties reached have been proven.
  ------------------------------------------------------------------------------
```


```
[eva:summary] ====== ANALYSIS SUMMARY ======
  ------------------------------------------------------------------------------
  3 functions analyzed (out of 3): 100% coverage.
  In these functions, 35 statements reached (out of 35): 100% coverage.
  ------------------------------------------------------------------------------
  No errors or warnings raised during the analysis.
  ------------------------------------------------------------------------------
  0 alarms generated by the analysis.
  ------------------------------------------------------------------------------
  Evaluation of the logical properties reached by the analysis:
    Assertions        0 valid     0 unknown     0 invalid      0 total
    Preconditions     4 valid     0 unknown     0 invalid      4 total
  100% of the logical properties reached have been proven.
  ------------------------------------------------------------------------------
```


3.其他

安装 `frama-c` 参照了Get Frama-C的教程，环境为WSL-Ubuntu22.04