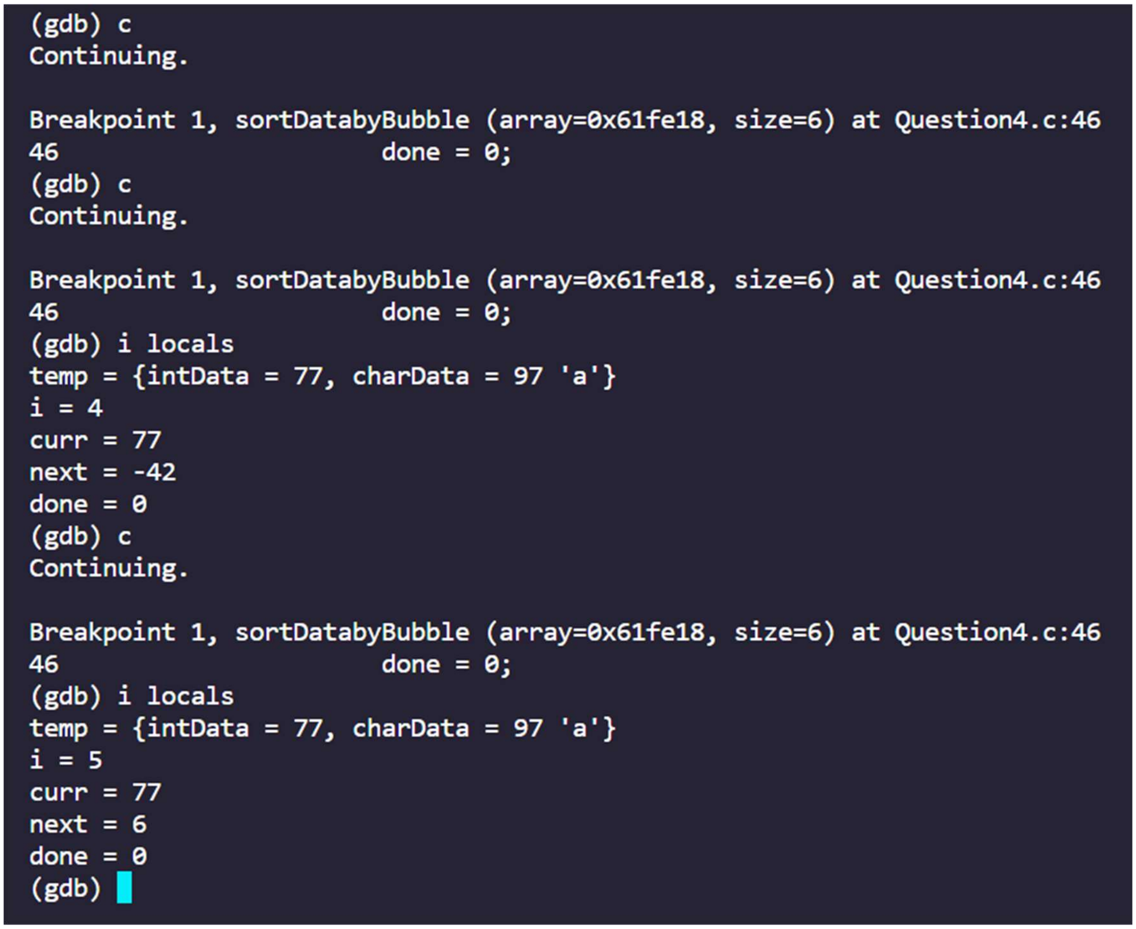
Bug Report – Q4 Bubble Sort

Bug observation 1: Boundary error

Program fails to stop at the second last element in the input array.

- Input: **int n = 6, input[] = {{10, 'c'}, {2, 'B'}, {-5, 'k'}, {12, 'z'}, {77, 'a'}, {-42, '?'}};**
- Expected: for loop exits when i = 4
- Actual: for loop exits i = 5



```
(gdb) c
Continuing.

Breakpoint 1, sortDatabyBubble (array=0x61fe18, size=6) at Question4.c:46
46                      done = 0;
(gdb) c
Continuing.

Breakpoint 1, sortDatabyBubble (array=0x61fe18, size=6) at Question4.c:46
46                      done = 0;
(gdb) i locals
temp = {intData = 77, charData = 97 'a'}
i = 4
curr = 77
next = -42
done = 0
(gdb) c
Continuing.

Breakpoint 1, sortDatabyBubble (array=0x61fe18, size=6) at Question4.c:46
46                      done = 0;
(gdb) i locals
temp = {intData = 77, charData = 97 'a'}
i = 5
curr = 77
next = 6
done = 0
(gdb)
```

GDB Analysis:

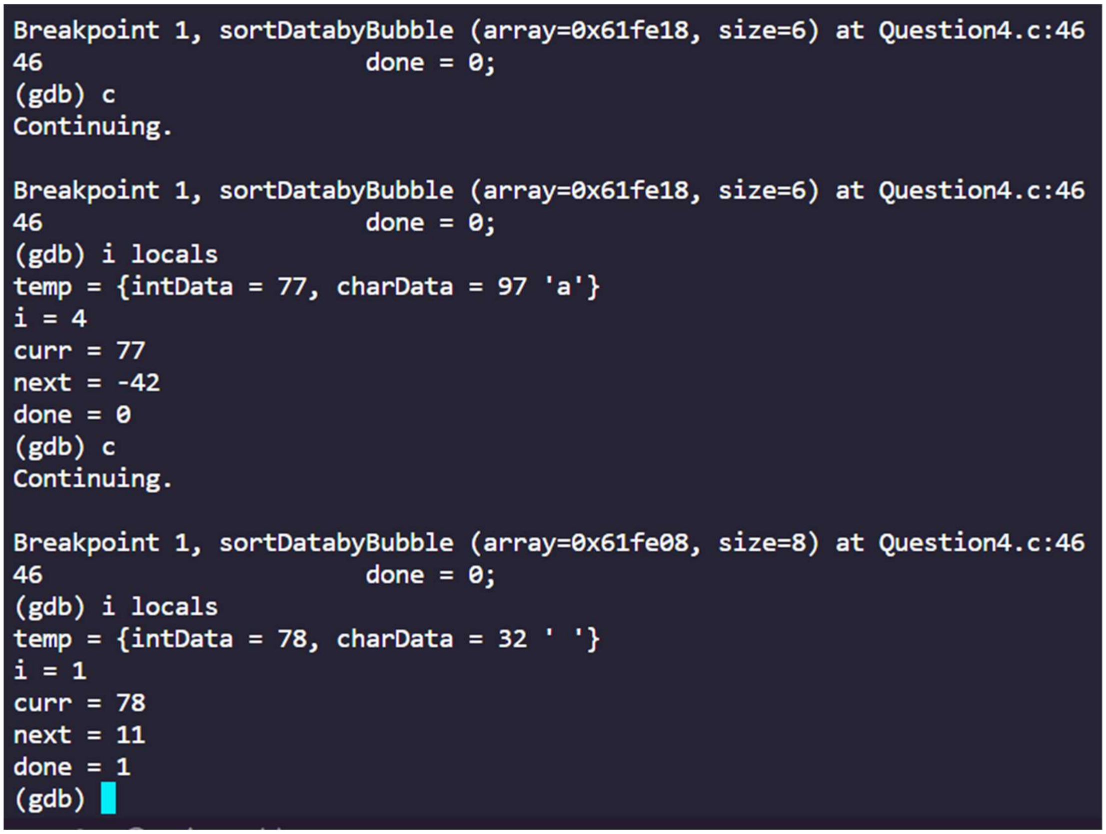
- Breakpoint placed at line 46 in Question4.c, right after the for loop completes 1 iteration
- Because the maximum index of the array is 4, GDB recognized that the variable "next" (array[i+1]) holds a value that's out of range.

Possible root cause:

Line 29 incorrectly used the <= logical operator instead of <

Bug Fix Validation 1:

Line 29 corrected to **for(i = 0; i < size - 1; i++).** Actual outcome matched.



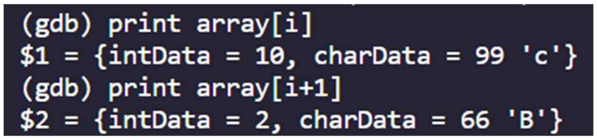
```
Breakpoint 1, sortDatabyBubble (array=0x61fe18, size=6) at Question4.c:46
46                      done = 0;
(gdb) c
Continuing.

Breakpoint 1, sortDatabyBubble (array=0x61fe18, size=6) at Question4.c:46
46                      done = 0;
(gdb) i locals
temp = {intData = 77, charData = 97 'a'}
i = 4
curr = 77
next = -42
done = 0
(gdb) c
Continuing.

Breakpoint 1, sortDatabyBubble (array=0x61fe08, size=8) at Question4.c:46
46                      done = 0;
(gdb) i locals
temp = {intData = 78, charData = 32 ' '}
i = 1
curr = 78
next = 11
done = 1
(gdb)
```

Bug Observation 2: Behavioural error

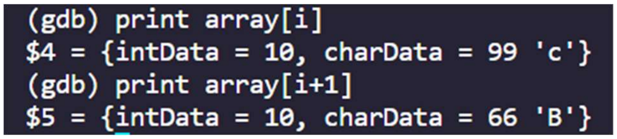
Program fails to swap elements when the subsequent one is greater than the current.

- Input: **int n = 6, input[] = {{10, 'c'}, {2, 'B'}, {-5, 'k'}, {12, 'z'}, {77, 'a'}, {-42, '?'}};**
- Expected: Swap {10, 'c'} with {2, 'B'} because 10 > 2
- Actual: Modified {2, 'B'} but did nothing to {10, 'c'}

Before swap (shown left) and after swap (shown right)



```
(gdb) print array[i]
$1 = {intData = 10, charData = 99 'c'}
(gdb) print array[i+1]
$2 = {intData = 2, charData = 66 'B'}
```



```
(gdb) print array[i]
$4 = {intData = 10, charData = 99 'c'}
(gdb) print array[i+1]
$5 = {intData = 10, charData = 66 'B'}
```

GDB Analysis:

- Breakpoint placed at line 35 in Question4.c, right as the program enters the conditional
- GDB indicates that the swapping algorithm didn't correctly switch element i and i+1's indices.

Possible root cause:

The variables in the swapping algorithm are placed wrong.

Bug Fix Validation 2:

Corrected lines 40-44 to

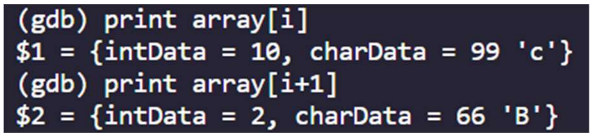
array[i].intData = array[i + 1].intData;

array[i].charData = array[i + 1].charData;

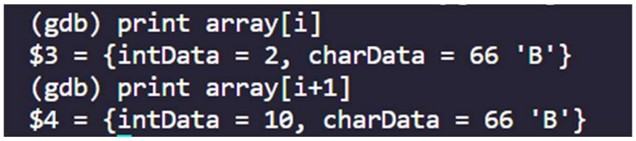
array[i + 1].intData = temp.intData;

array[i + 1].charData = charData;

Actual output matched. Before swap (shown on left) and after swap (shown on right) below.



```
(gdb) print array[i]
$1 = {intData = 10, charData = 99 'c'}
(gdb) print array[i+1]
$2 = {intData = 2, charData = 66 'B'}
```



```
(gdb) print array[i]
$3 = {intData = 2, charData = 66 'B'}
(gdb) print array[i+1]
$4 = {intData = 10, charData = 66 'B'}
```