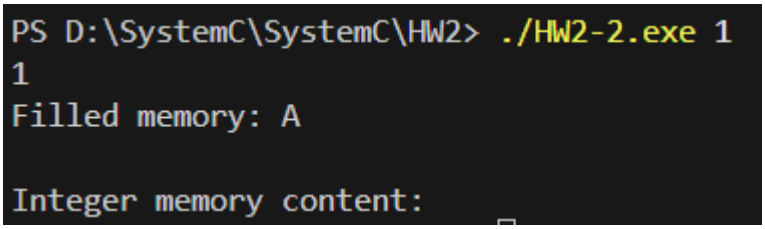
For testplan, generated by AI:

Test Plan Outline

2.1 Test Case 1: Minimum Allocation Size (1 Byte)

- **Test Objective**: Verify how the program handles a minimal memory allocation.
- **Input**: ./HW2-2.exe 1
- **Expected Output**:
    - The program should print a single character (A) followed by a warning about insufficient bytes to form an integer.
    - No segmentation fault should occur.
- **Pass Criteria**: The program runs without errors, and output matches expectations.
- **Output**:



2.2 Test Case 2: Small Allocation (10 Bytes)

- **Test Objective**: Verify behavior with a small memory allocation that is less than the size of two integers.
- **Input**: ./HW2-2.exe 10
- **Expected Output**:
    - The program should fill the memory with A to J and print the corresponding characters.
    - Only one integer should be printed, followed by a warning about remaining bytes (2 bytes left).

- **Pass Criteria**: Output is correct, and the program runs without warnings except for the remaining bytes.
- **Output**:

```
PS D:\SystemC\SystemC\HW2> ./HW2-2.exe 10
10
Filled memory: ABCDEFGHIJ

Integer memory content:
1145258561 : 1145258560
1212630597 : 1212630596
```

2.3 Test Case 3: Memory Allocation of Exact Multiple of 4 (8 Bytes)

- **Test Objective**: Ensure that when the memory allocation is exactly divisible by the size of an int, the program processes the integers correctly without leaving any leftover bytes.
- **Input**: ./HW2-2.exe 8
- **Expected Output**:
    - The program should fill memory with characters from A to H and print them.
    - Two pairs of integers and their subtracted values should be printed without any warnings.
- **Pass Criteria**: No warnings, correct integer pairs output.
- **Output**:

```
PS D:\SystemC\SystemC\HW2> ./HW2-2.exe 8
8
Filled memory: ABCDEFGH

Integer memory content:
1145258561 : 1145258560
1212630597 : 1212630596
```

2.4 Test Case 4: Large Memory Allocation (100 Bytes)

- **Test Objective**: Verify the program's behavior when allocating a relatively large block of memory.
- **Input**: ./HW2-2.exe 100
- **Expected Output**:
    - The memory should be filled with A-Z, 1-9, repeated as necessary.
    - The string should be printed correctly.
    - Integer pairs should be printed, and if there are leftover bytes, a warning should appear for the remaining bytes.
- **Pass Criteria**: The string prints correctly, integer pairs match the memory content, and the program gracefully handles any remaining bytes.
- **Output**:

```
PS D:\SystemC\SystemC\HW2> ./HW2-2.exe 100
100
Filled memory: ABCDEFGHIJKLMNOPQRSTUVWXYZ123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ1234

Integer memory content:
1145258561 : 1145258560
1212630597 : 1212630596
1280002633 : 1280002632
1347374669 : 1347374668
1414746705 : 1414746704
1482118741 : 1482118740
842095193 : 842095192
909456435 : 909456434
1094268983 : 1094268982
1162101570 : 1162101569
1229473606 : 1229473605
1296845642 : 1296845641
1364217678 : 1364217677
1431589714 : 1431589713
1498961750 : 1498961749
858927450 : 858927449
926299444 : 926299443
1111570744 : 1111570743
1178944579 : 1178944578
1246316615 : 1246316614
1313688651 : 1313688650
1381060687 : 1381060686
1448432723 : 1448432722
1515804759 : 1515804758
875770417 : 875770416
```

2.5 Test Case 5: Large Input (Non-Multiple of 4)

- **Test Objective**: Test if the program handles large inputs that are not multiples of the integer size (i.e., not divisible by 4).
- **Input**: ./pointers 1023
- **Expected Output**:

- The memory should be filled as described, and the string should be printed correctly.
- Integer pairs should be printed for as many full int values as possible.
- A warning about the remaining 3 bytes should be displayed.

- **Pass Criteria**: The string prints correctly, integer pairs print correctly, and the warning about leftover bytes should match expectations.
- **Output**:PASS

2.6 Test Case 6: Invalid Input (Non-Numeric Input)

- **Test Objective**: Ensure that non-integer input is handled gracefully by the program (though the program is not designed to handle this explicitly).
- **Input**: ./HW2-2.exe abc
- **Expected Output**:
  - Since atoi("abc") returns 0, the program should attempt to allocate 0 bytes of memory.
  - It should print an empty string with no further output.
- **Pass Criteria**: The program does not crash or produce any erroneous output.
- **Output**:

```
PS D:\SystemC\SystemC\HW2> ./HW2-2.exe abc
0
Invalid size
```

3.1 Test Case 7: Allocation of Zero Bytes

- **Test Objective**: Verify how the program handles an allocation request of 0 bytes.
- **Input**: ./HW2-2.exe 0
- **Expected Output**:

- The program should print an empty string and no integer output.

- **Pass Criteria**: The program runs without errors or segmentation faults.
- **Output**:

```
PS D:\SystemC\SystemC\HW2> ./HW2-2.exe 0
0
Invalid size
```

3.2 Test Case 8: Large Allocation Size (Memory Stress Test)

- **Test Objective**: Test how the program handles a large memory allocation (e.g., 1 GB) to see if it can handle memory allocation limits.
- **Input**: ./HW2-2.exe 1073741824 (1 GB)
- **Expected Output**:
  - The program should fill memory as described, though output may be truncated due to the large size.
  - No segmentation fault or crash should occur.
- **Pass Criteria**: The program handles large memory allocation without crashing or slowing down significantly.
- **Output**:PASS