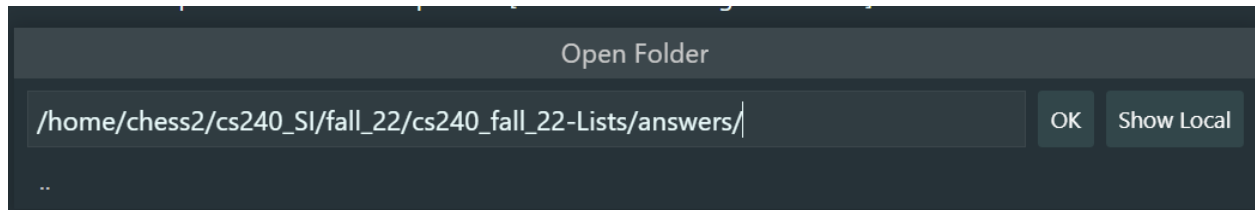
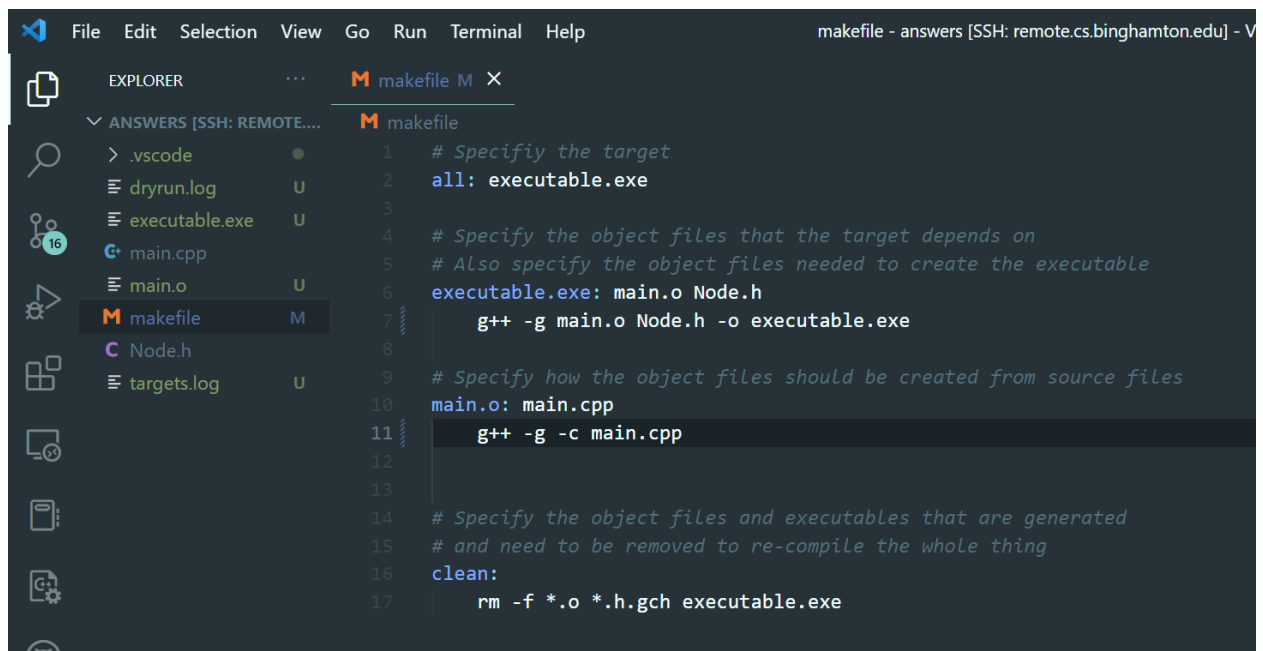


A quick VScode debugger guide

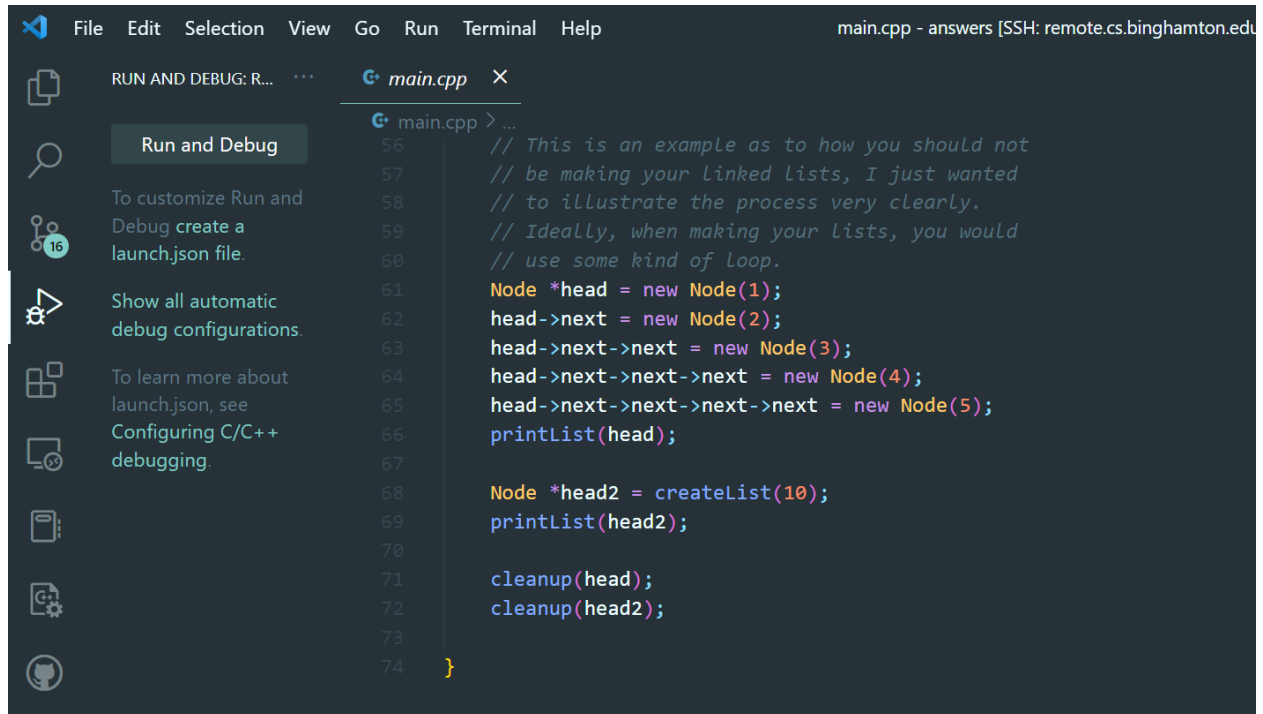
1. Open the target folder that houses your project and executable using the “**Open Folder**” option in the File menu.



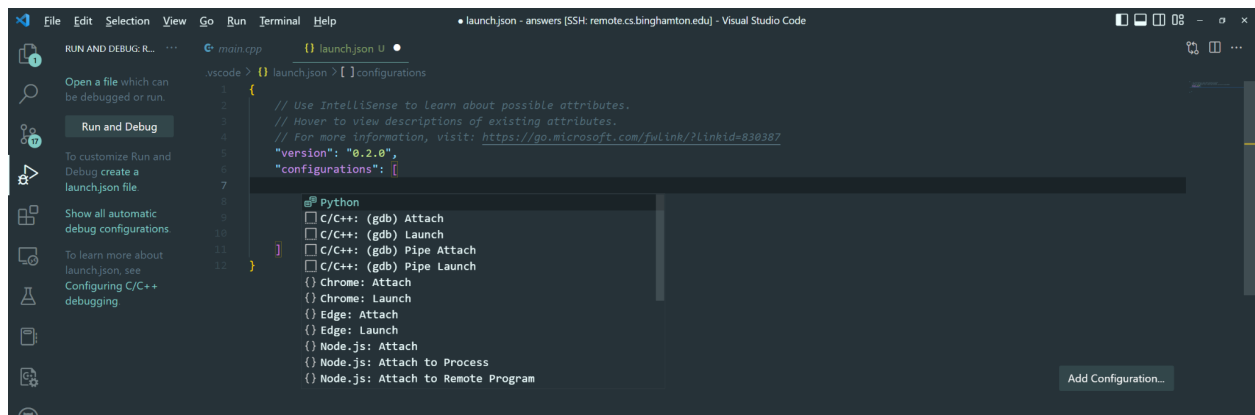
2. Make sure you're compiling with debug symbols (-g), and note your executable's name



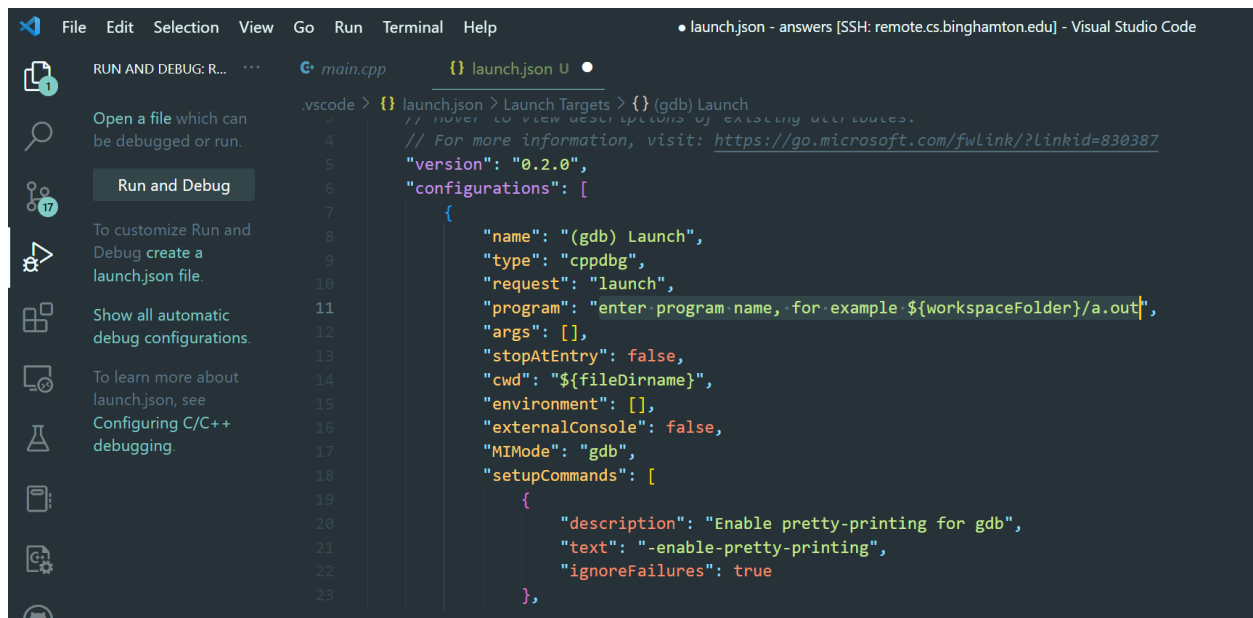
3. Click the **Run and Debug** button, it's the arrow looking thing on the left side of the page. Then click "**create a launch.json file**"



4. Click the "**Add Configuration**" button on the bottom right and select "**C/C++: (gdb) Launch**".

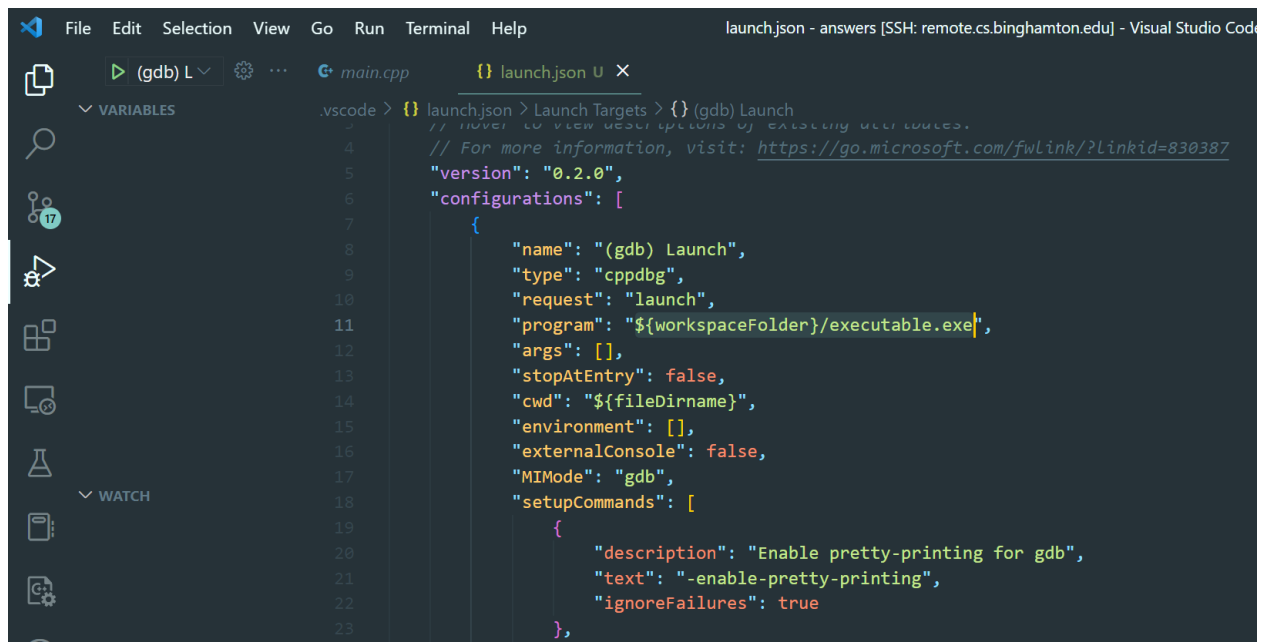


5. Locate the **"program"** string and change it from this...



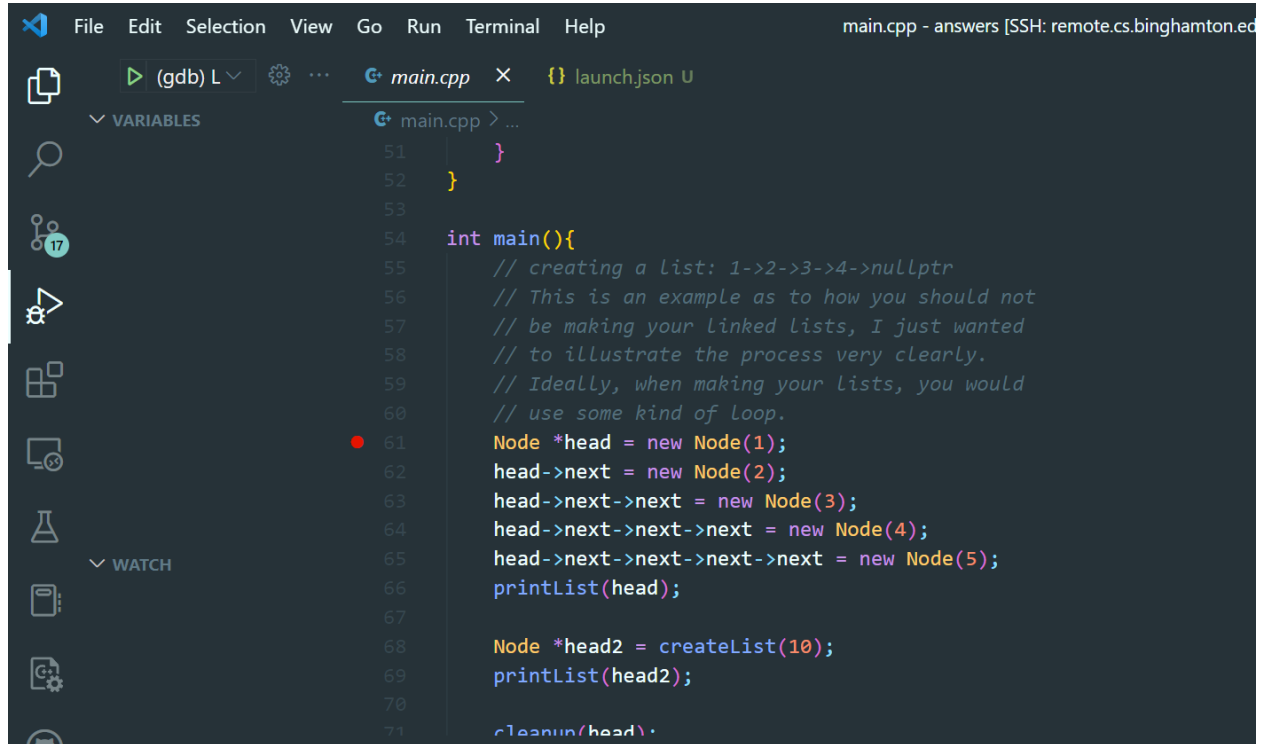
```
.vscode > {} launch.json > Launch Targets > {} (gdb) Launch
// Hover to view descriptions of existing attributes.
// For more information, visit: https://go.microsoft.com/fwlink/?Linkid=830387
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "(gdb) Launch",
      "type": "cppdbg",
      "request": "launch",
      "program": "enter program name, for example ${workspaceFolder}/a.out",
      "args": [],
      "stopAtEntry": false,
      "cwd": "${fileDirname}",
      "environment": [],
      "externalConsole": false,
      "MIMode": "gdb",
      "setupCommands": [
        {
          "description": "Enable pretty-printing for gdb",
          "text": "-enable-pretty-printing",
          "ignoreFailures": true
        }
      ]
    }
  ]
}
```

To reflect your executable's name. If your executable were named **"my_exe"**, yours would be **"\${workspaceFolder}/my_exe"**. Make sure to **save the launch.json** file!



```
.vscode > {} launch.json > Launch Targets > {} (gdb) Launch
// Hover to view descriptions of existing attributes.
// For more information, visit: https://go.microsoft.com/fwlink/?Linkid=830387
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "(gdb) Launch",
      "type": "cppdbg",
      "request": "launch",
      "program": "${workspaceFolder}/executable.exe",
      "args": [],
      "stopAtEntry": false,
      "cwd": "${fileDirname}",
      "environment": [],
      "externalConsole": false,
      "MIMode": "gdb",
      "setupCommands": [
        {
          "description": "Enable pretty-printing for gdb",
          "text": "-enable-pretty-printing",
          "ignoreFailures": true
        }
      ]
    }
  ]
}
```

6. Open a file that you want to debug, and leave it as the active window. Note that my `main.cpp` is italicized as *main.cpp* because it is active (I clicked on it last). Add a breakpoint on whatever line you like.

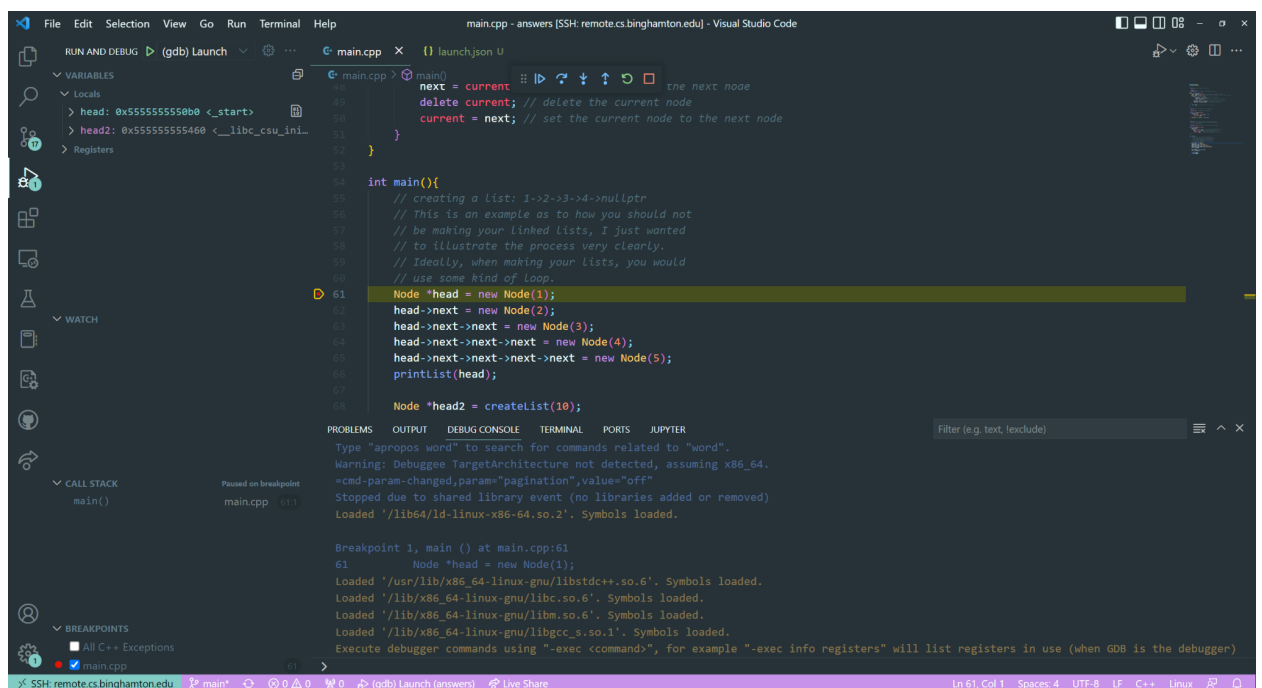


```
File Edit Selection View Go Run Terminal Help main.cpp - answers [SSH: remote.cs.binghamton.edu]

main.cpp x {} launch.json U

VARIABLES
main.cpp > ...
51 }
52 }
53
54 int main(){
55     // creating a list: 1->2->3->4->nullptr
56     // This is an example as to how you should not
57     // be making your linked lists, I just wanted
58     // to illustrate the process very clearly.
59     // Ideally, when making your lists, you would
60     // use some kind of loop.
61     Node *head = new Node(1);
62     head->next = new Node(2);
63     head->next->next = new Node(3);
64     head->next->next->next = new Node(4);
65     head->next->next->next->next = new Node(5);
66     printList(head);
67
68     Node *head2 = createList(10);
69     printList(head2);
70
71     cleanup(head);
}
```

7. Press the green run arrow, **(gdb) Launch**, and enjoy your debugging.



```
File Edit Selection View Go Run Terminal Help main.cpp - answers [SSH: remote.cs.binghamton.edu] - Visual Studio Code

RUN AND DEBUG (gdb) Launch main.cpp x {} launch.json U

VARIABLES
main.cpp > (main)
> head: 0x5555555558b0 <_start>
> head2: 0x555555555460 <__libc_csu_in_...>
> Registers

WATCH
main.cpp > ...

int main(){
    // creating a list: 1->2->3->4->nullptr
    // This is an example as to how you should not
    // be making your linked lists, I just wanted
    // to illustrate the process very clearly.
    // Ideally, when making your lists, you would
    // use some kind of loop.
61 Node *head = new Node(1);
62 head->next = new Node(2);
63 head->next->next = new Node(3);
64 head->next->next->next = new Node(4);
65 head->next->next->next->next = new Node(5);
66 printList(head);
67
68 Node *head2 = createList(10);
69 printList(head2);
70
71 cleanup(head);
}
```