

Devin Hardy

```
// Devin Hardy
// Asg 1
// Attempt at the parcing problem

#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <cstring>
#include <string>
#include <algorithm>

char* makeCStrg(std::string str)
{
    char* char_array = new char[str.length() + 1];
    char_array[str.length()] = '\0';
    for (int i = 0; i < str.length(); i++)
    {
        char_array[i] = str[i];
    }
    return char_array;
}

// [X] string to c-string
// [?] splits the c-string into tokens
// [?] have pointers to tokens
// [?] pointer at pointers
// special characters '<', '>', '<<', '>>', '|' = white space
// return a dynamically allocated array of c-strings
// double pointer is an array of something
char** parseCommandLine(std::string aCommandLine)
{
    int M = 10;
    int i = 0;
    const char delim[] = { '<', '>', '<<', '>>', '|', ' ' };
    char* cStrg = makeCStrg(aCommandLine);
    char* token = strtok(cStrg, delim);
    char** pptr = new char* [M]; // pointer array of 5
    while (token!=NULL) // get the rest of the tokens
    {
        pptr[i] = token; //
        i++;
        token = strtok(NULL, delim);
    }
    pptr[i] = NULL;
    return pptr; // return pointer
}

// So far string into a c string

int main()
{
    std::string example = "example<line|of code >>no promise it works";
    char** tokens = parseCommandLine(example);
    int itr = 0;
    while (tokens[itr] != NULL)
    {
        std::cout << tokens[itr] << std::endl;
    }
}
```

```

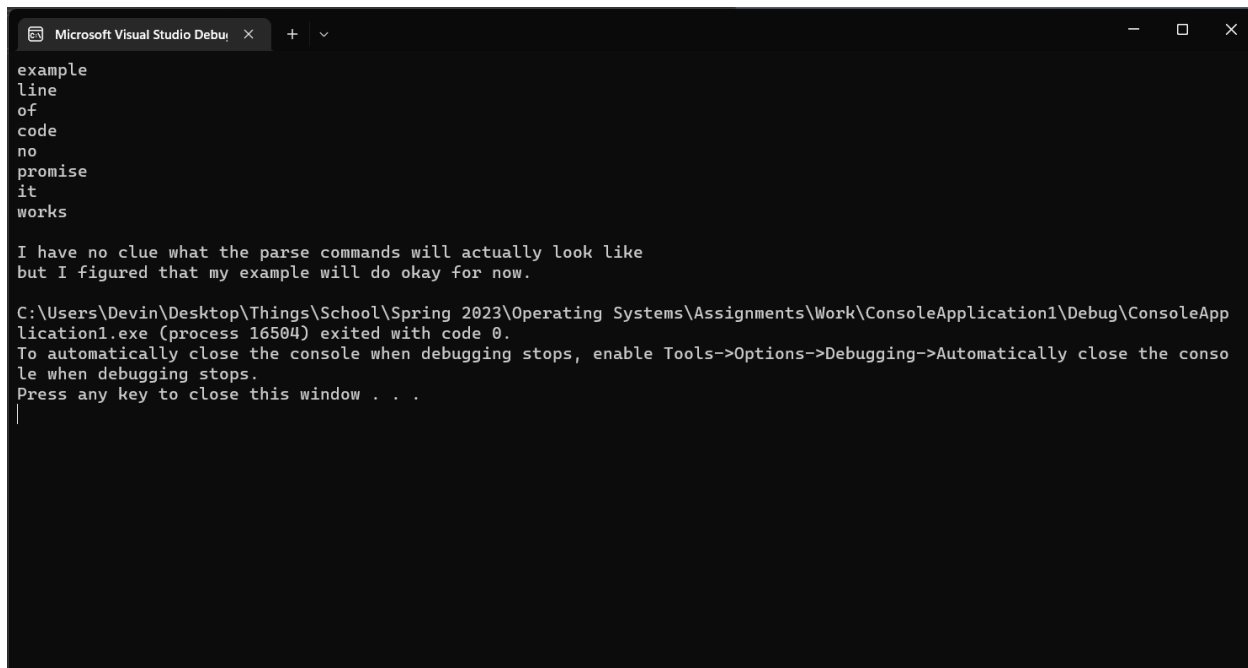
        itr++;
    }
    std::cout << std::endl;

    std::cout << "I have no clue what the parse commands will actually look like" <<
std::endl;
    std::cout << "but I figured that my example will do okay for now." << std::endl;
    return 0;
}

// Run program: Ctrl + F5 or Debug > Start Without Debugging menu
// Debug program: F5 or Debug > Start Debugging menu

// Tips for Getting Started:
// 1. Use the Solution Explorer window to add/manage files
// 2. Use the Team Explorer window to connect to source control
// 3. Use the Output window to see build output and other messages
// 4. Use the Error List window to view errors
// 5. Go to Project > Add New Item to create new code files, or Project > Add Existing
Item to add existing code files to the project
// 6. In the future, to open this project again, go to File > Open > Project and select
the .sln file

```



```

Microsoft Visual Studio Debug Console
example
line
of
code
no
promise
it
works

I have no clue what the parse commands will actually look like
but I figured that my example will do okay for now.

C:\Users\Devin\Desktop\Things\School\Spring 2023\Operating Systems\Assignments\Work\ConsoleApplication1\Debug\ConsoleApp
lication1.exe (process 16504) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .

```