

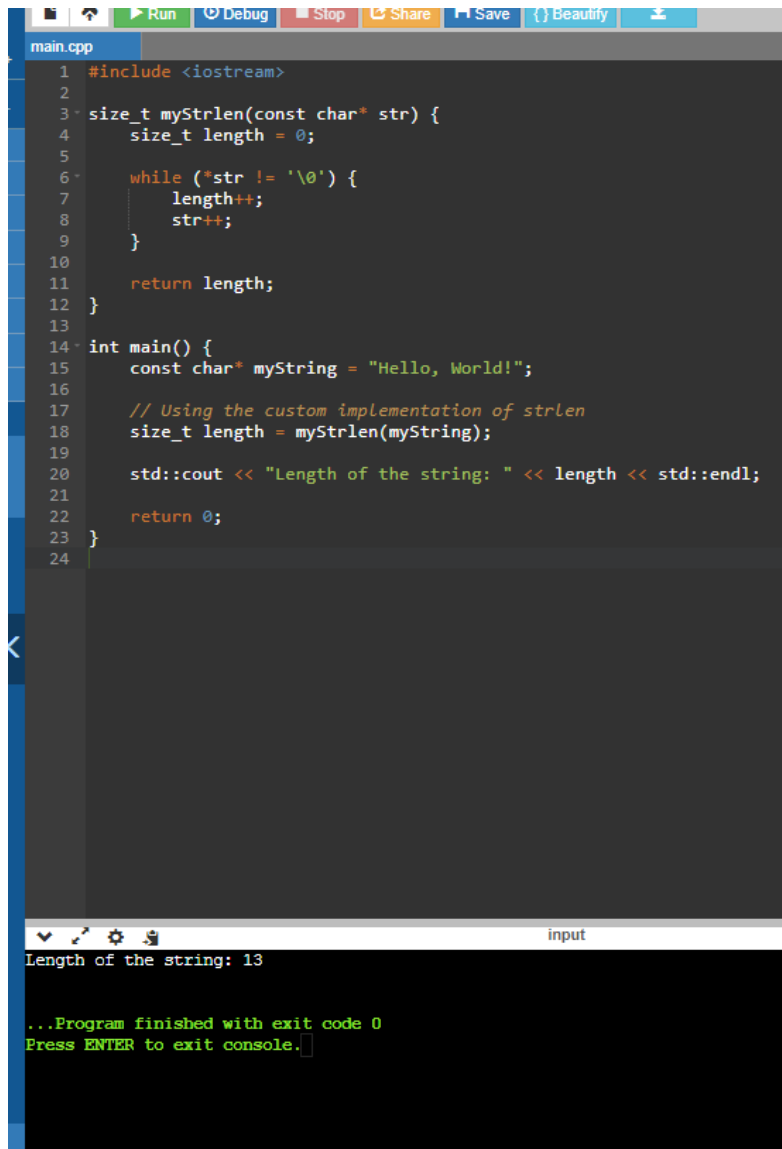
KANISHK BEHL

102105058

3EIC-2

DSA ASSIGN-3

WAP a program to impement strlen() function.



The image shows a screenshot of a C++ IDE with a dark theme. The editor window displays a file named `main.cpp` with the following code:

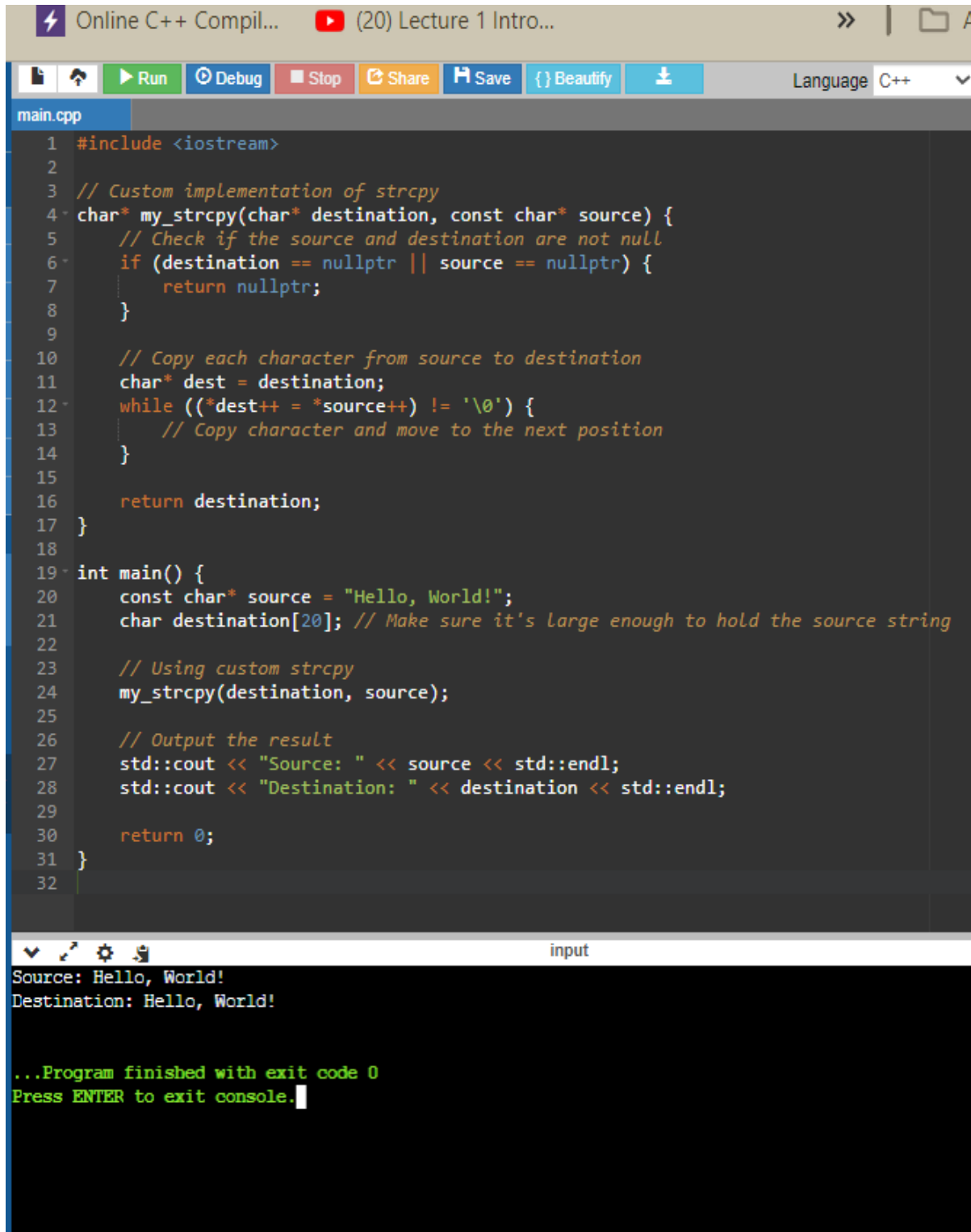
```
1 #include <iostream>
2
3 size_t myStrlen(const char* str) {
4     size_t length = 0;
5
6     while (*str != '\0') {
7         length++;
8         str++;
9     }
10
11     return length;
12 }
13
14 int main() {
15     const char* myString = "Hello, World!";
16
17     // Using the custom implementation of strlen
18     size_t length = myStrlen(myString);
19
20     std::cout << "Length of the string: " << length << std::endl;
21
22     return 0;
23 }
24
```

Below the editor, there is a console window titled "input". It shows the output of the program:

```
Length of the string: 13

...Program finished with exit code 0
Press ENTER to exit console.
```

WAP to impement strcpy() function.



The screenshot displays an online C++ compiler interface. The top bar shows the compiler name 'Online C++ Compil...' and a video icon with the text '(20) Lecture 1 Intro...'. Below this is a toolbar with buttons for 'Run', 'Debug', 'Stop', 'Share', 'Save', 'Beautify', and a download icon. The 'Language' dropdown is set to 'C++'. The main editor area, titled 'main.cpp', contains the following C++ code:

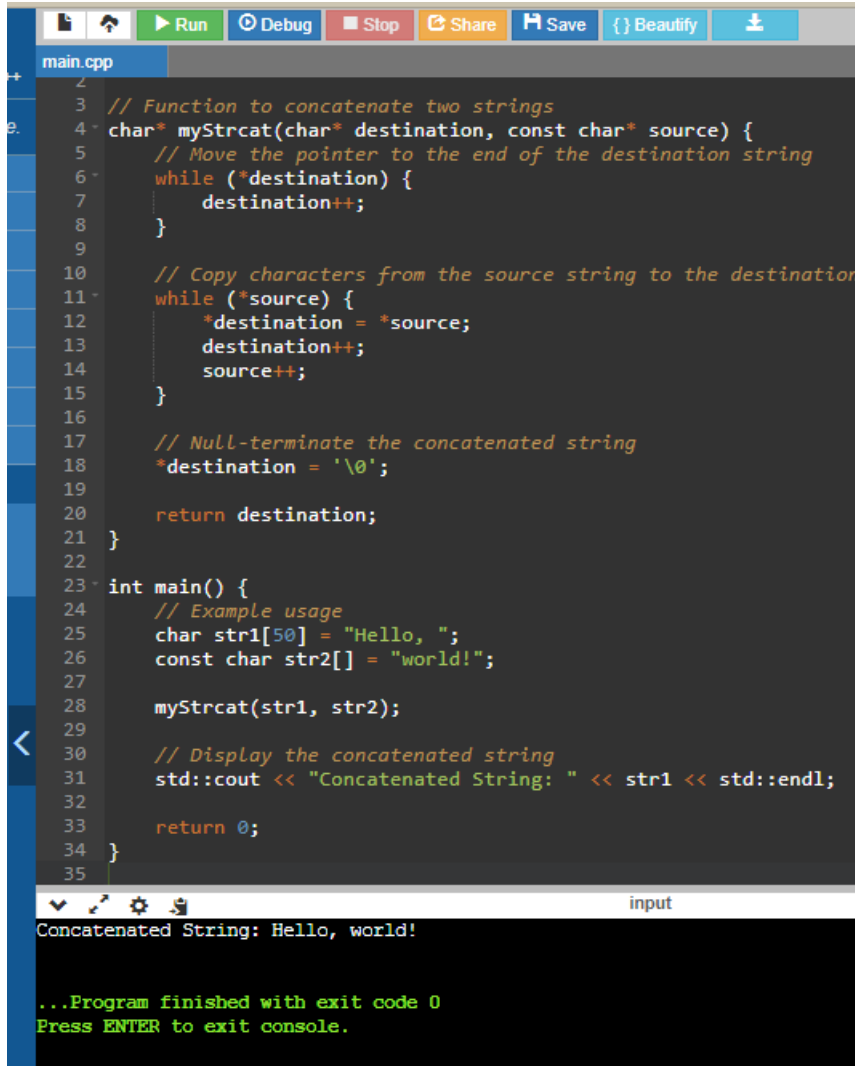
```
1 #include <iostream>
2
3 // Custom implementation of strcpy
4 char* my_strcpy(char* destination, const char* source) {
5     // Check if the source and destination are not null
6     if (destination == nullptr || source == nullptr) {
7         return nullptr;
8     }
9
10    // Copy each character from source to destination
11    char* dest = destination;
12    while ((*dest++ = *source++) != '\0') {
13        // Copy character and move to the next position
14    }
15
16    return destination;
17 }
18
19 int main() {
20     const char* source = "Hello, World!";
21     char destination[20]; // Make sure it's large enough to hold the source string
22
23     // Using custom strcpy
24     my_strcpy(destination, source);
25
26     // Output the result
27     std::cout << "Source: " << source << std::endl;
28     std::cout << "Destination: " << destination << std::endl;
29
30     return 0;
31 }
32
```

Below the code editor is a console window with a title bar containing icons for window management and the label 'input'. The console output is as follows:

```
Source: Hello, World!
Destination: Hello, World!

...Program finished with exit code 0
Press ENTER to exit console.
```

WAP A PROGRAM TO IMPLEMENT STRCAT() function.



The image shows a C++ IDE with a dark theme. The top toolbar includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The file name 'main.cpp' is visible in the top left. The code is as follows:

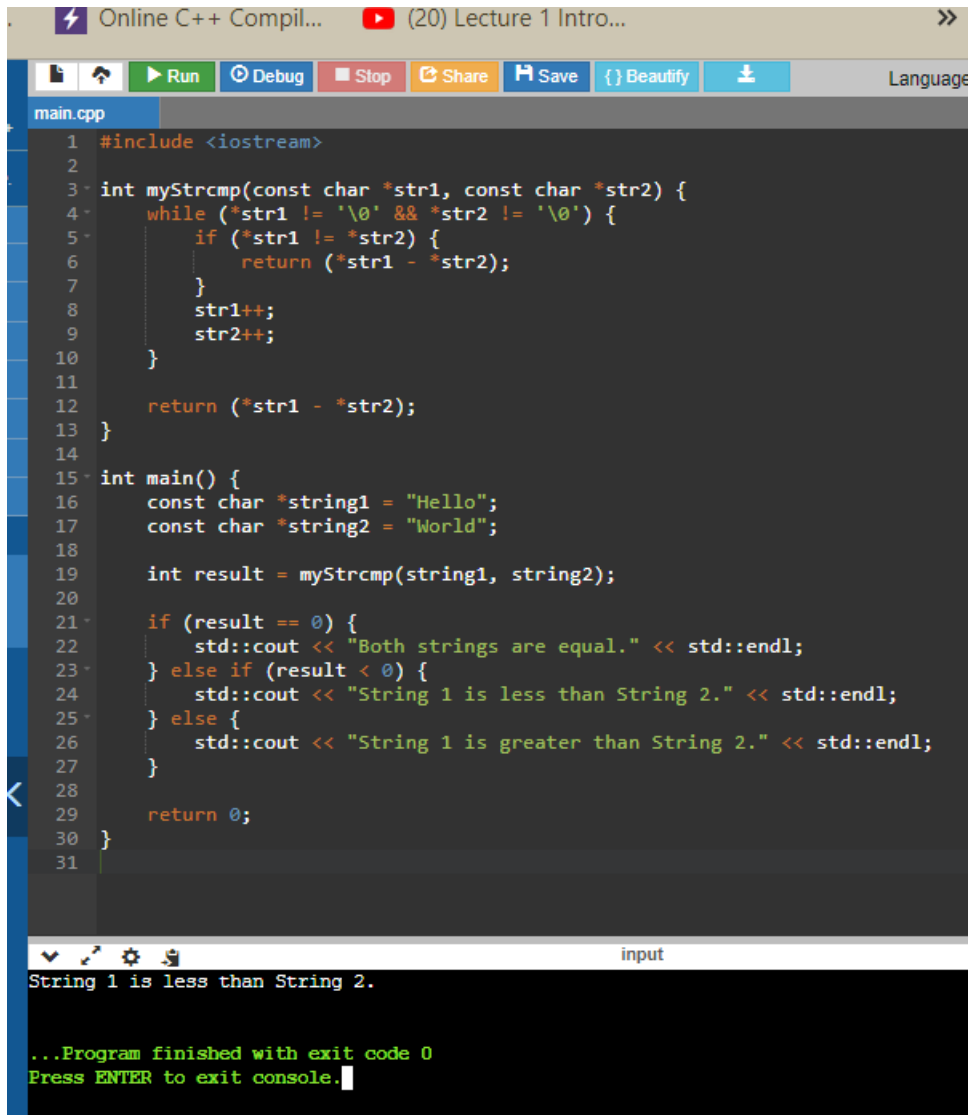
```
2
3 // Function to concatenate two strings
4 char* myStrcat(char* destination, const char* source) {
5     // Move the pointer to the end of the destination string
6     while (*destination) {
7         destination++;
8     }
9
10    // Copy characters from the source string to the destination
11    while (*source) {
12        *destination = *source;
13        destination++;
14        source++;
15    }
16
17    // Null-terminate the concatenated string
18    *destination = '\0';
19
20    return destination;
21 }
22
23 int main() {
24     // Example usage
25     char str1[50] = "Hello, ";
26     const char str2[] = "world!";
27
28     myStrcat(str1, str2);
29
30     // Display the concatenated string
31     std::cout << "Concatenated String: " << str1 << std::endl;
32
33     return 0;
34 }
35
```

At the bottom, there is a console window with the following output:

```
input
Concatenated String: Hello, world!

...Program finished with exit code 0
Press ENTER to exit console.
```

WAP to implement strcmp() function.



The screenshot shows an online C++ compiler interface. The top bar includes a lightning bolt icon, the text "Online C++ Compil...", a YouTube icon, "(20) Lecture 1 Intro...", and a double right arrow icon. Below this is a toolbar with buttons for "Run", "Debug", "Stop", "Share", "Save", "Beautify", and a download icon. The "Language" dropdown is set to "C++". The main editor area, titled "main.cpp", contains the following C++ code:

```
1 #include <iostream>
2
3 int myStrcmp(const char *str1, const char *str2) {
4     while (*str1 != '\0' && *str2 != '\0') {
5         if (*str1 != *str2) {
6             return (*str1 - *str2);
7         }
8         str1++;
9         str2++;
10    }
11
12    return (*str1 - *str2);
13 }
14
15 int main() {
16     const char *string1 = "Hello";
17     const char *string2 = "World";
18
19     int result = myStrcmp(string1, string2);
20
21     if (result == 0) {
22         std::cout << "Both strings are equal." << std::endl;
23     } else if (result < 0) {
24         std::cout << "String 1 is less than String 2." << std::endl;
25     } else {
26         std::cout << "String 1 is greater than String 2." << std::endl;
27     }
28
29     return 0;
30 }
31
```

Below the code editor is a console window with the label "input". It displays the output of the program:

```
String 1 is less than String 2.

...Program finished with exit code 0
Press ENTER to exit console.
```

WAP to demonstrate limitations of 2D array of characters.

```
main.cpp
2
3 int main() {
4     const int rows = 2;
5     const int cols = 3;
6
7     // Creating a 2D array of characters
8     char charArray[rows][cols] = {
9         {'A', 'B', 'C'},
10        {'D', 'E', 'F'}
11    };
12
13    // Displaying the original array
14    std::cout << "Original 2D Array:" << std::endl;
15    for (int i = 0; i < rows; ++i) {
16        for (int j = 0; j < cols; ++j) {
17            std::cout << charArray[i][j] << ' ';
18        }
19        std::cout << std::endl;
20    }
21
22    // Attempting to add a new row
23    charArray[2][0] = 'G'; // This will result in undefined behavior
24
25    // Displaying the modified array
26    std::cout << "Modified 2D Array:" << std::endl;
27    for (int i = 0; i < rows + 1; ++i) { // Note: Accessing beyond the allocated size
28        for (int j = 0; j < cols; ++j) {
29            std::cout << charArray[i][j] << ' ';
30        }
31        std::cout << std::endl;
32    }
33
34    return 0;
35 }
36
```

input

```
Original 2D Array:
A B C
D E F
Modified 2D Array:
A B C
D E F
G
*** stack smashing detected ***: terminated

...Program finished with exit code 0
Press ENTER to exit console.
```

WAP to demonstrate an array of pointers to strings.

Online C++ Compiler (20) Lecture 1 Intro...

Run Debug Stop Share Save Beautify Language C++

main.cpp

```
1 #include <iostream>
2
3 int main() {
4     // Array of pointers to strings
5     const char *strings[] = {
6         "Hello,",
7         "This",
8         "is",
9         "an",
10        "array",
11        "of",
12        "pointers",
13        "to",
14        "strings."
15    };
16
17    // Displaying the strings using array of pointers
18    std::cout << "Strings using Array of Pointers:" << std::endl;
19    for (int i = 0; i < sizeof(strings) / sizeof(strings[0]); ++i) {
20        std::cout << strings[i] << " ";
21    }
22
23    // Accessing individual characters in each string
24    std::cout << "\n\nIndividual characters in each string:" << std::endl;
25    for (int i = 0; i < sizeof(strings) / sizeof(strings[0]); ++i) {
26        const char *currentString = strings[i];
27        while (*currentString != '\0') {
28            std::cout << *currentString << " ";
29            ++currentString;
30        }
31        std::cout << std::endl;
32    }
33
34    return 0;
35 }
36
```

input

Strings using Array of Pointers:
Hello, This is an array of pointers to strings.

Individual characters in each string:
H e l l o ,
T h i s
i s
a n
a r r a y
o f
p o i n t e r s
t o
s t r i n g s .