

BLG102E Extra Session

Pointers and Strings

1. An array of characters

- Write a program to print a word which is stored in an array of characters.
 - First use %c which is for formatting chars
 - Then use %s which is for formatting strings

```
#include <stdio.h>

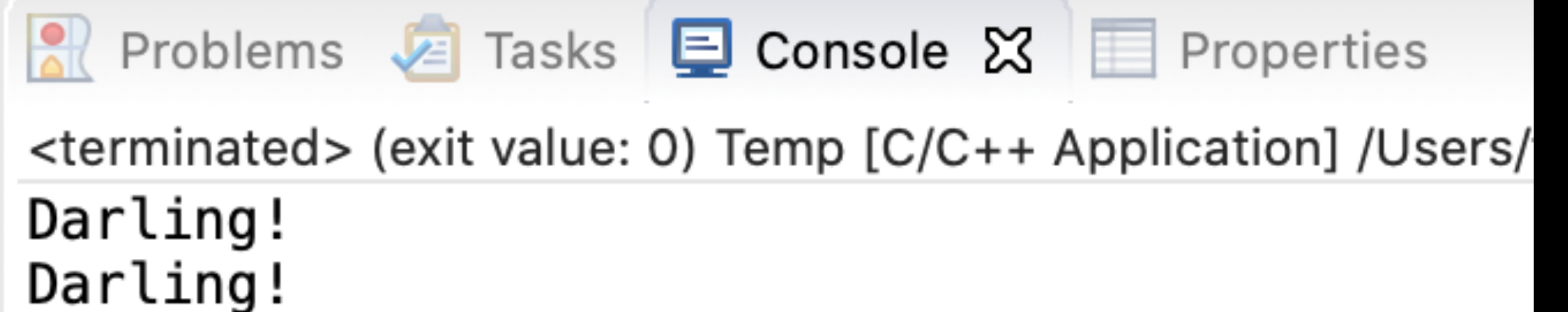
int main() {

    char word[] = {'D', 'a', 'r', 'l', 'i', 'n', 'g', '!', '\0'};
    int i;

    // First print each character using %c.

    // Then print the word as a string using %s.

    return 0;
}
```



The screenshot shows the bottom panel of a code editor with four tabs: Problems, Tasks, Console, and Properties. The Console tab is active, displaying the output of the program. The text in the console is: <terminated> (exit value: 0) Temp [C/C++ Application] /Users/ Darling! Darling! The first line is the status bar, and the following two lines are the program's output.

2. Address of an array

- Write a program to print a word stored in an array using a pointer which stores the address of the array.
 - First print the word using the array itself.
 - Second print the word using the address of the array stored in a pointer.
 - Then print the reverse of the word using the address of the array stored in a pointer.

```
#include <stdio.h>

int main() {

    char word[15];
    char *ptr = &word[0];
    // ptr = word;
    // ptr = &word[SIZE -1];
    int i;

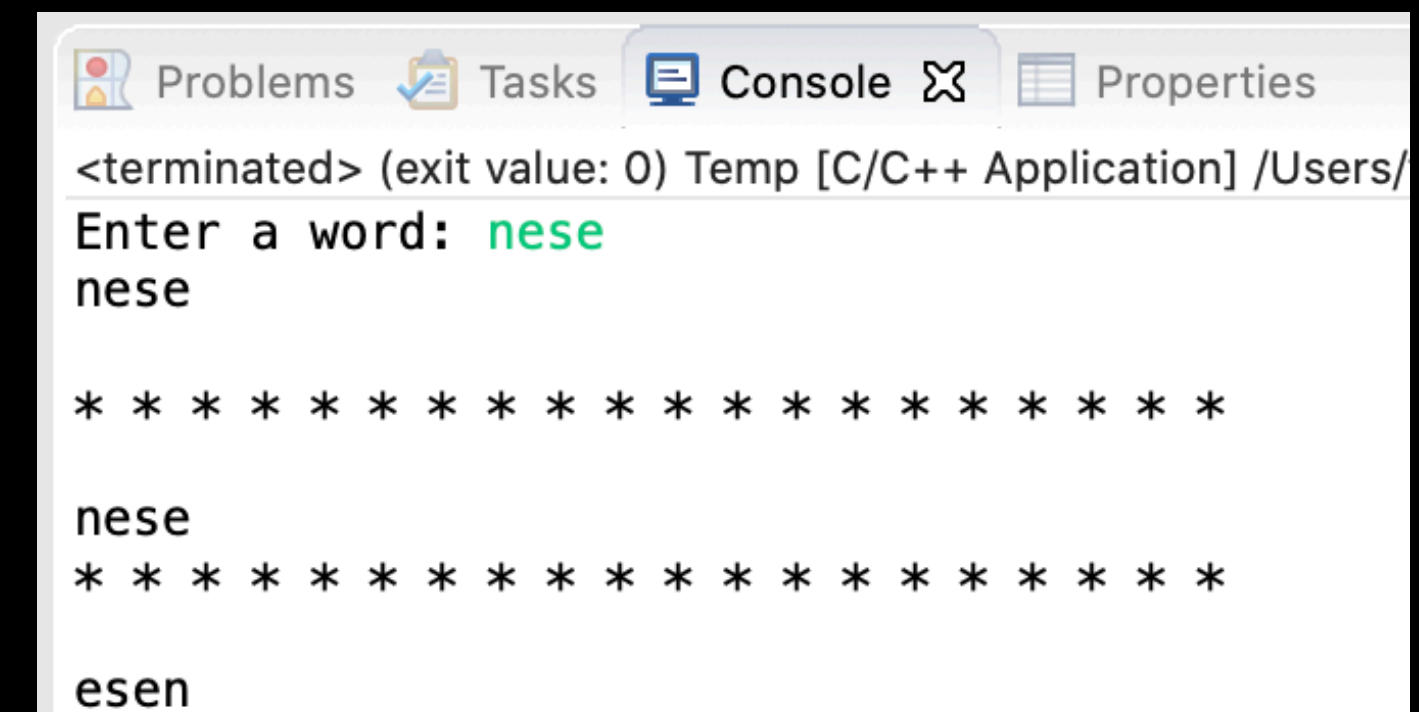
    printf("Enter a word: ");
    scanf("%s", word);

    // First print the word using the array itself.

    // Second print the word
    // using the address of the array stored in a pointer.

    // Then print the reverse of the word
    // using the address of the array stored in a pointer.

    return 0;
}
```



The screenshot shows a C++ IDE with tabs for Problems, Tasks, Console, and Properties. The Console tab is active, displaying the program's output. It starts with a prompt "Enter a word:" followed by the user input "nese". Below this, a separator line of asterisks is shown, followed by the word "nese" printed using the pointer. Another separator line of asterisks follows, and then the word "esen" is printed, which is the reverse of "nese".

```
<terminated> (exit value: 0) Temp [C/C++ Application] /Users/
Enter a word: nese
nese

* * * * *
nese
* * * * *
esen
```

3. Separate the sentence into phrases

- Write a program to print each word of a sentence separately using whitespace character ' ' as a separator.
 - First print the first word of the sentence as we don't have a whitespace character before the first word.
 - Then print the rest of the words using blank ' ' as a separator.

```
#include <stdio.h>

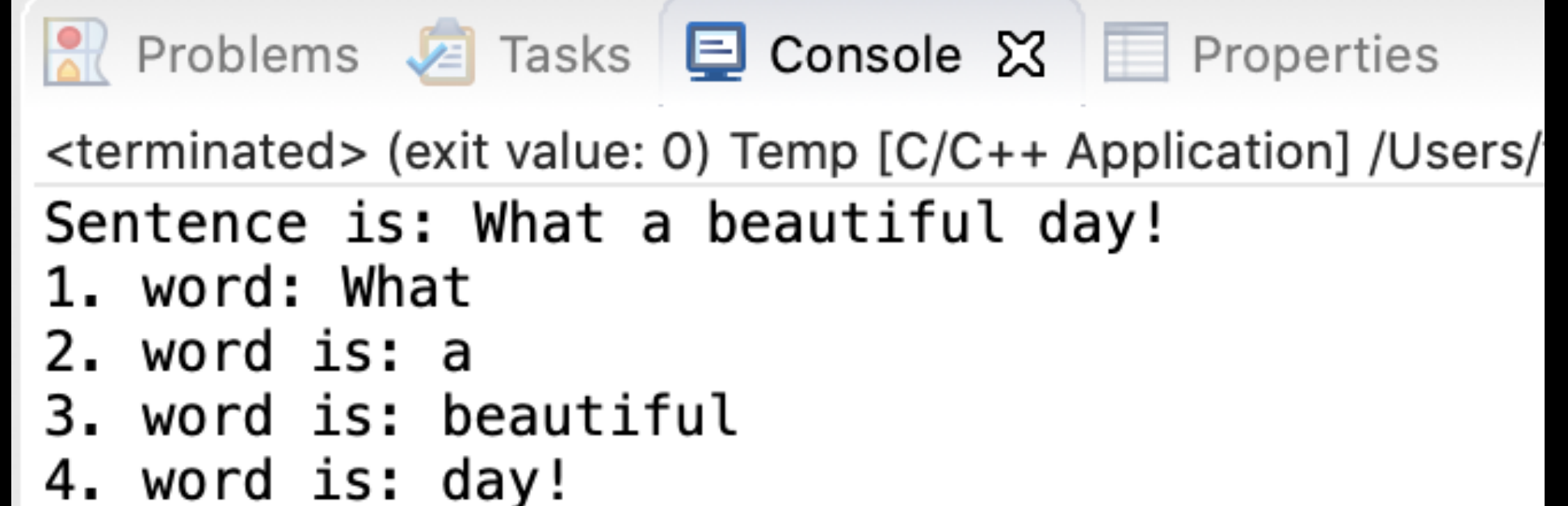
int main() {

    char *sentence = "What a beautiful day!";
    char *word = sentence;
    int i = 1;
    char *ptr;

    printf("Sentence is: %s \n", sentence);

    // Print each word of the sentence separately.

    return 0;
}
```



The screenshot shows a code editor interface with tabs for Problems, Tasks, Console, and Properties. The Console tab is active, displaying the output of a C program. The output shows the full sentence first, followed by a list of words separated by spaces.

```
<terminated> (exit value: 0) Temp [C/C++ Application] /Users/
Sentence is: What a beautiful day!
1. word: What
2. word is: a
3. word is: beautiful
4. word is: day!
```

4. Upper case format of a string

- Write a program to get upper case version of a string.
 - First get the sentence from the user using `fgets()` function.
 - Then change lower case characters to upper case characters.
- Remember upper case 'A' is 65, lower case 'a' is 97 in ASCII Table.

```
#include <stdio.h>

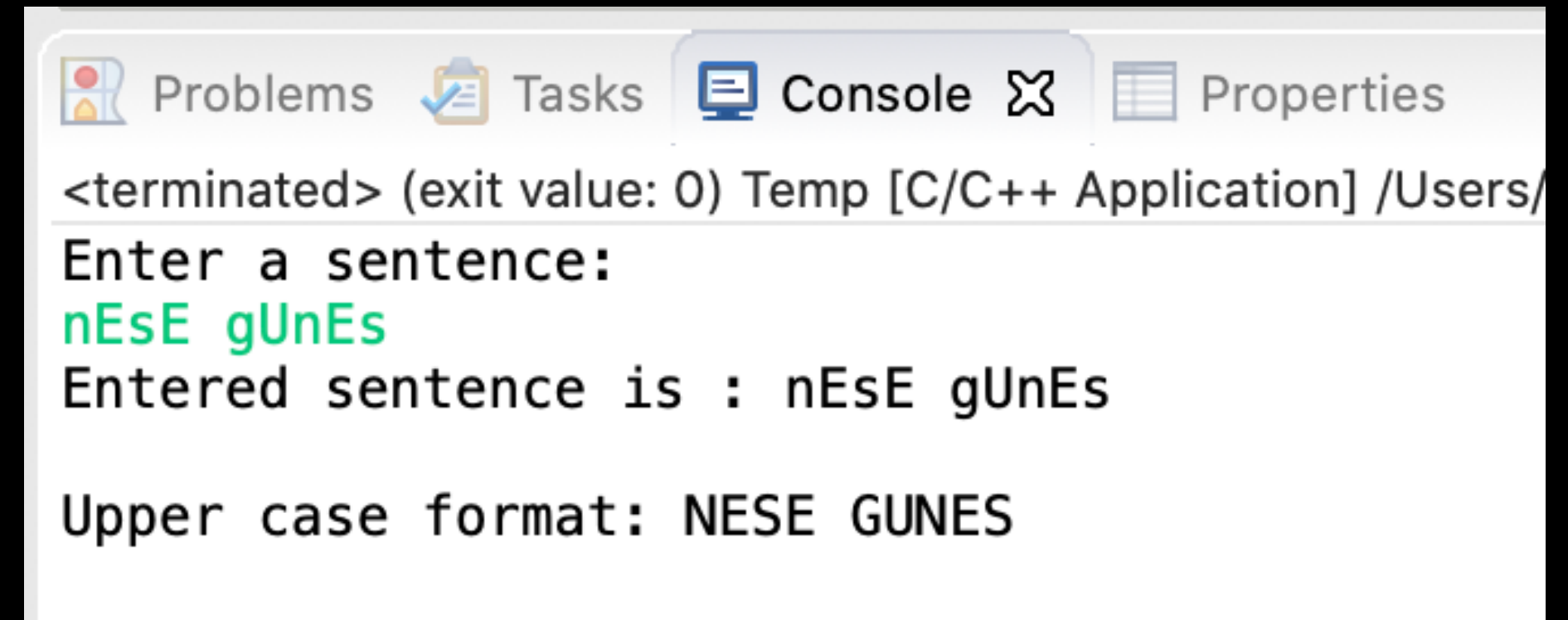
int main() {

    char str[20], *ptr;
    int diff = 'a' - 'A';

    // First get the sentence from the user using fgets() function.

    // Then change lower case characters to upper case characters.

    return 0;
}
```



The screenshot shows a console window with the following text:

```
<terminated> (exit value: 0) Temp [C/C++ Application] /Users/
Enter a sentence:
nEsE gUnEs
Entered sentence is : nEsE gUnEs

Upper case format: NESE GUNES
```


5. Length of a string

Standard C library functions

- Write a program to calculate the length of string using standard C library function `strlen()`.
 - First get the sentence from the user using `fgets()` function.
 - Next print the length of the string using `strlen()`.
 - Then use `strcpy()` function and print the copied sentence.

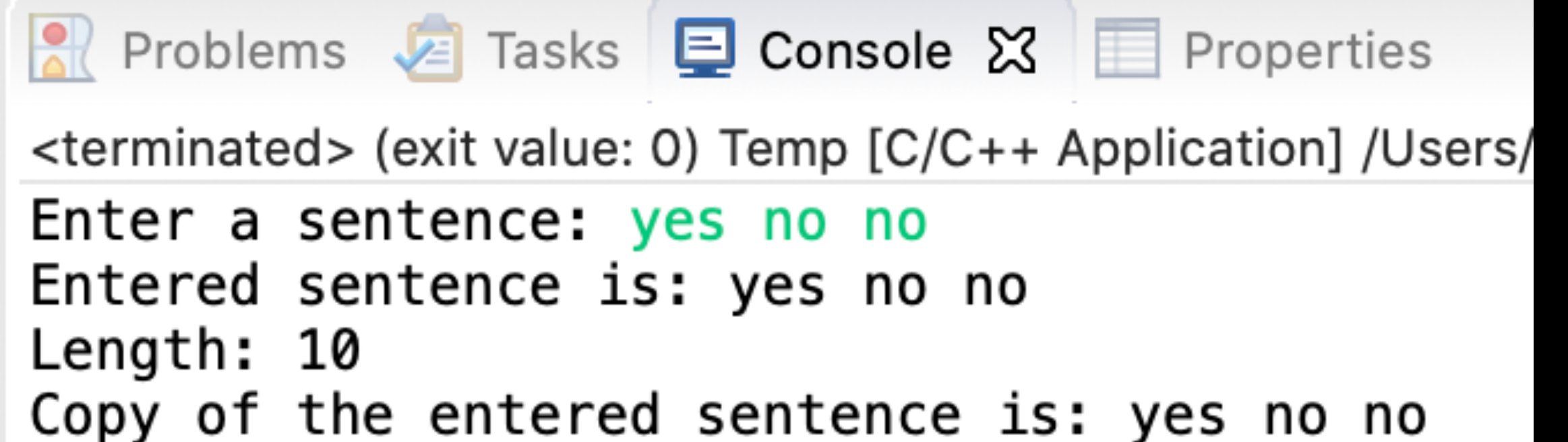
```
#include <stdio.h>
#include <string.h>

int main() {

    char sentence[20], copy[20];

    // Get the sentence using fgets() function.
    // Print the length of the function using strlen().
    // Copy the sentence and print the copied one.

    return 0;
}
```



The screenshot shows a console window with tabs for Problems, Tasks, Console, and Properties. The Console tab is active, displaying the output of a C program. The text in the console is as follows:

```
<terminated> (exit value: 0) Temp [C/C++ Application] /Users/
Enter a sentence: yes no no
Entered sentence is: yes no no
Length: 10
Copy of the entered sentence is: yes no no
```

6. Convert string to integer using atoi() from <stdlib.h>

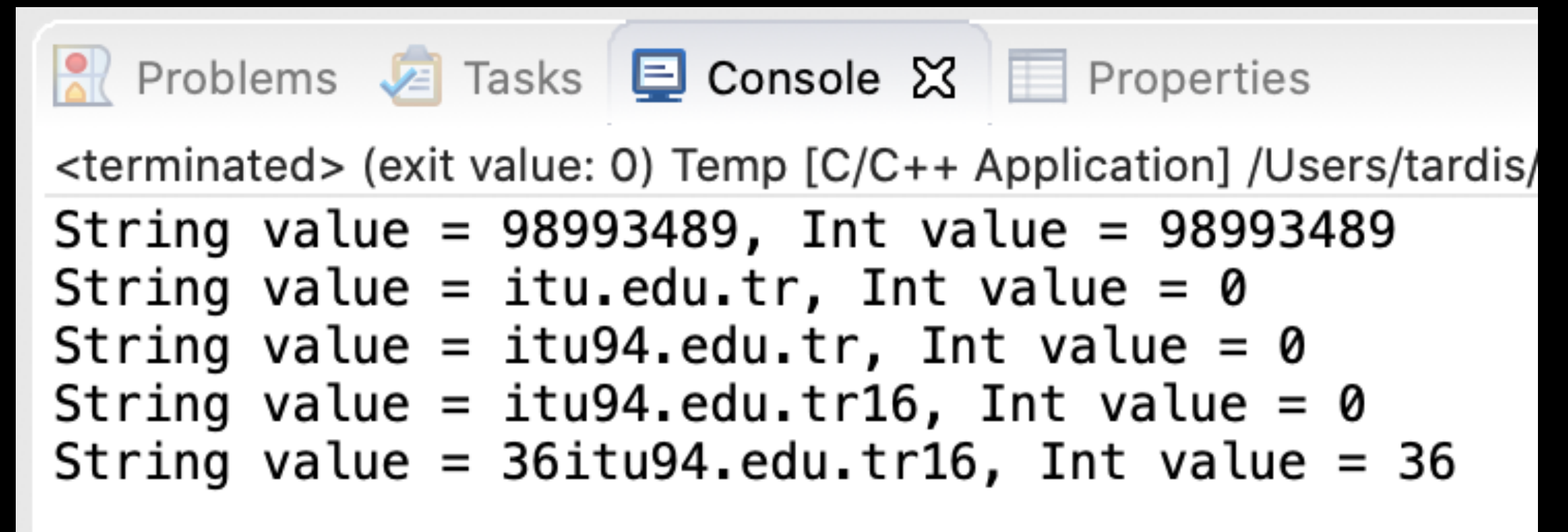
- Write a program to get the integer value of a string using C standard library function atoi(), note that we must include <stdlib.h>.
 - First get the integer value of a string which stores: "98993489".
 - Second get the integer value of a string which stores: "itu.edu.tr".
 - Next get the integer value of a string which stores: "itu94.edu.tr".
 - Then get the integer value of a string which stores: "itu94.edu.tr16".
 - Lastly get the integer value of a string which stores: "36itu94.edu.tr16".

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main () {
    int val;
    char str[20];

    // First get the integer value of a string which stores: "98993489".
    // Second get the integer value of a string which stores: "itu.edu.tr".
    // Next get the integer value of a string which stores: "itu94.edu.tr".
    // Then get the integer value of a string which stores: "itu94.edu.tr16".
    // Lastly get the integer value of a string which stores: "36itu94.edu.tr16".

    return(0);
}
```



The screenshot shows a code editor interface with tabs for Problems, Tasks, Console, and Properties. The Console tab is active, displaying the output of the program. The output shows the string value and the integer value returned by atoi() for each of the five test cases.

```
<terminated> (exit value: 0) Temp [C/C++ Application] /Users/tardis/
String value = 98993489, Int value = 98993489
String value = itu.edu.tr, Int value = 0
String value = itu94.edu.tr, Int value = 0
String value = itu94.edu.tr16, Int value = 0
String value = 36itu94.edu.tr16, Int value = 36
```

Good luck.