Line-by-Line Explanation of C Program (Palindrome)

Code Snippet 1

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

• stdio.h is used for input-output functions like printf() and gets().
• string.h is used for string functions such as strlen() and strcmp().

Code Snippet 2

```
void reverseString(char str[], char rev[]) {
    int i, len;
    len = strlen(str);

    for (i = 0; i < len; i++) {
        rev[i] = str[len - i - 1];
    }
    rev[len] = '\0';
}
```

• reverseString() is a user-defined function to reverse the string.
• strlen() calculates the length of the string.
• The loop copies characters from end to beginning.
• '\0' is added to properly end the string.

Code Snippet 3

```
int main() {
    char str[100], rev[100];

    printf("Enter a string: ");
    gets(str);
```

• main() is the execution starting point.
• Two character arrays store original and reversed strings.
• gets() takes string input including spaces.

Code Snippet 4

```
    reverseString(str, rev);

    printf("\nOriginal String : %s", str);
    printf("\nReversed String : %s", rev);

    if (strcmp(str, rev) == 0) {
        printf("\nResult : The string is a PALINDROME");
    } else {
        printf("\nResult : The string is NOT a PALINDROME");
    }
```

```
    return 0;
}
```

• reverseString() is called to reverse the string.
• strcmp() compares both strings.
• Equal strings mean palindrome.

Complete C Program

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

void reverseString(char str[], char rev[]) {
    int i, len;
    len = strlen(str);

    for (i = 0; i < len; i++) {
        rev[i] = str[len - i - 1];
    }
    rev[len] = '\0';
}

int main() {
    char str[100], rev[100];

    printf("Enter a string: ");
    gets(str);

    reverseString(str, rev);

    printf("\nOriginal String : %s", str);
    printf("\nReversed String : %s", rev);

    if (strcmp(str, rev) == 0) {
        printf("\nResult : The string is a PALINDROME");
    } else {
        printf("\nResult : The string is NOT a PALINDROME");
    }

    return 0;
}
```