

Forum:[Hack Forums](#)

Thread: [Converting a String to each element of an integer array.](#)

AssistanceDelinquent - November 8, 2018, 4:41 pm

I have no idea on how to do this task with C.

For example: When user enters 12345 this will be stored as a string. I want to gather each digit of user answer and convert/migrate it to all integer arrays element.

So a integer has the size of 5 and 1 would be intarray[0], 2 would be in intarray[1] etc. etc.

My code directly adds the string[i] to intarray[i]. This is incorrect. Can someone provide any insight for a solution of my problem.

Irritable - November 9, 2018, 8:21 am

(Nov 8, 2018 05:41 PM)AssistanceDelinquent Wrote: [\[CITE\]](#) I have no idea on how to do this task with C.

For example: When user enters 12345 this will be stored as a string. I want to gather each digit of user answer and convert/migrate it to all integer arrays element.

So a integer has the size of 5 and 1 would be intarray[0], 2 would be in intarray[1] etc. etc.

My code directly adds the string[x] to intarray[x]. This is incorrect. Can someone provide any insight for a solution of my problem. Hello, AssistanceDelinquent.

You need to use [scanf\(\)](#) to solve this problem. The reference to the function is hyperlinked.

I do not know what you have so far, nor do I know what your programming requirements are. Here is an example of how to achieve what you're requesting:

Code `#include <stdio.h> /* scanf() */`

```
int main() {
    const int MAX_ARRAY_ELEMENTS = 5;

    int arrayOfInts[MAX_ARRAY_ELEMENTS];

    printf("Please enter 5 digits: ");
    for (int n = 0; n < MAX_ARRAY_ELEMENTS; ++n) {
        scanf("%d", &arrayOfInts[n]);
    }

    printf("\nThe entered data string is: ");
    for (int r = 0; r < MAX_ARRAY_ELEMENTS; ++r) {
        printf("%d", arrayOfInts[r]);
    }

    return 0;
}
```

I hope this helps. ?

icko1996 - November 9, 2018, 11:04 am

First make sure you validate the string:

- trim / ignore any whitespace
- trim / ignore characters which are not a digit

If the character encoding of the string is ASCII, which is very likely, you can use this trick.

```
Code size_t len = strlen(str);
for (int i = 0; i < len; ++i)
{
    array[ i ] = str[ i ] - '0';
}
```