

Forum:[Hack Forums](#)

Thread: [Help.What's wrong. something wrong when the customers name come out.](#)

imrankhan - December 25, 2013, 3:07 pm

School homework- a level computing

Design a data structure for the Natwest Bank.

It must hold the customers:

Name

Account Number

Balance

There will be 10 customers – Think back to your use of arrays. You must use an array [] of variables. Think about how you will access these.

here's my code.

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

```
int main(){
```

```
    struct student
    {
        int balance;
        int id;
        char name[20];
    };
    struct student st1[9];
```

```
    int i;
    for(i=0;i<=9;i++){
        printf("Enter the customers %d Name   ID and   balance", i+1);
        scanf("%s""%d""%d", &st1[i].name, &st1[i].id, &st1[i].balance );
    }
```

```
        int k;
        printf("what customer details do you want see");
        scanf("%d", &k);
```

```
switch(k){
```

```
case 1:
```

```
    printf("The customers 1 Name %d, ID %d and balance %d\n",st1[0].name, st1[0].id, st1[0].balance
```

```
);
    break;

    case 2:
printf("The customers 2 Name %d, ID %d and balance %d\n",st1[1].name, st1[1].id, st1[1].balance
);
    break;

    case 3:
printf("The customers 3 Name %d, ID %d and balance %d\n",st1[2].name, st1[2].id, st1[2].balance
);
    break;

case 4:
printf("The customers 4 Name %d, ID %d and balance %d\n",st1[3].name, st1[3].id, st1[3].balance
);
    break;

    case 5:
printf("The customers 5 Name %d, ID %d and balance %d\n",st1[4].name, st1[4].id, st1[4].balance
);
    break;

case 6:
printf("The customers 6 Name %d, ID %d and balance %d\n",st1[5].name, st1[5].id, st1[5].balance
);
    break;

    case 7:
printf("The customers 7 Name %d, ID %d and balance %d\n",st1[6].name, st1[6].id, st1[6].balance
);
    break;

    case 8:
printf("The customers 8 Name %d, ID %d and balance %d\n",st1[7].name, st1[7].id, st1[7].balance
);
    break;

    case 9:
printf("The customers 9 Name %d, ID %d and balance %d\n",st1[8].name, st1[8].id, st1[8].balance
```

```

);
    break;

    case 10:
        printf("The customers 10 Name %d, ID %s and balance %d\n",st1[9].name, st1[9].id,
st1[9].balance );
        break;

    default :
        printf("that's not a valid value");
        break;
    }

    getchar();
    getchar();
}

```

something wrong when the customers name come out.
something wrong when the customers name come out.

Sajuuk - December 25, 2013, 3:33 pm

Eww, put that damn code in code tags.

Code [code]Code Here[/ code]

imrankhan - December 25, 2013, 3:37 pm

i am new, so sorry.

how do you do that ?

but can you figure the problem out

g4143 - December 25, 2013, 3:38 pm

(Dec 25, 2013 04:07 PM)imrankhan Wrote: [\[CITE\]](#) something wrong when the customers name come out.

something wrong when the customers name come out.Are we supposed to guess what the problem is? Is it doubling like the above statements?

imrankhan - December 25, 2013, 3:41 pm

problem is that when the user wants the information like id,name and balance, the name comes out in integers.Obviously the name is meant to come in string.

LevelUp - December 25, 2013, 3:45 pm

(Dec 25, 2013 04:41 PM)imrankhan Wrote: [\[CITE\]](#) problem is that when the user wants the information like id,name and balance, the name comes out in integers.Obviously the name is meant to come in string.

You're using %d rather than %s when getting the name within your switch.

imrankhan - December 25, 2013, 3:45 pm

g4143 do you understand the promblem?

thank you
omg i was trying to avoid that problem

and how po i improve the code, any suggestions?

?lassy - December 25, 2013, 4:18 pm

(Dec 25, 2013 04:45 PM)imrankhan Wrote: [\[CITE\]](#) g4143 do you understand the promblem?

thank you
omg i was trying to avoid that problem

and how po i improve the code, any suggestions?
Don't use that big ass switch statement.

Do this:

PHP Code

```
<?php
int k;
printf("what customer details do you want see");
scanf("%d", &k);

printf("The customers %d Name %s, ID %d and balance %d\n", k, st1[k-1].name, st1[k-1].id, st1[k-1].balance );
^ This can go out of bounds, but just throw it in an if statement and you're good to go.
```

Ex.

PHP Code

```
<?php
int k;
printf("what customer details do you want see");
scanf("%d", &k);

if(k >= 1 && k <= 10)
    printf("The customers %d Name %s, ID %d and balance %d\n", k, st1[k-1].name, st1[k-1].id, st1[k-1].balance );
else
    printf("Invalid input %d", k);
```

0xDEAD10CC - December 26, 2013, 1:43 am

(Dec 25, 2013 05:18 PM)?lassy Wrote: [\[CITE\]](#) Don't use that big ass switch statement.

Do this:

PHP Code

```
<?php
int k;
printf("what customer details do you want see");
scanf("%d", &k);

printf("The customers %d Name %s, ID %d and balance %d\n", k, st1[k].name, st1[k].id, st1[k].
```

```
balance );
```

^ This can go out of bounds, but just throw it in an if statement and you're good to go.

Ex.

PHP Code

```
<?php
int k;
printf("what customer details do you want see");
scanf("%d", &k);

if(k >= 1 && <= 10)
    printf("The customers %d Name %s, ID %d and balance %d\n", k, st1[k].name, st1[k].id, st1[k].
balance );
else
    printf("Invalid input %d", k);
```

Why are you using k for indexing, if you continue with logic that dictates a 1-based array exists? If you're using k directly to index a value in the array and NOT k - 1, then shouldn't that be k >= 0, and k < 10?

Still, something seems off about this:

Code if(k >= 1 && <= 10)

Wouldn't you think? :) [hint]if(k >= 1 && <= 10)[/hint]

Code #include <stdio.h>

```
#define MAX_STUDENTS 10
```

```
typedef struct student
```

```
{
    int balance;
    int id;
    char name[20];
} student;
```

```
int main()
```

```
{
    student arr[MAX_STUDENTS];

    int i, k;
    for( i = 0; i < MAX_STUDENTS; i++ )
    {
        printf("Enter the customers %d Name ID and balance\n>>", i + 1);
        printf("Ex: John 1 250\n");
        scanf("%s %d %d", arr[i].name, &arr[i].id, &arr[i].balance);
    }
    getchar();
```

```
    printf("What customer details do you want see? [1-%d]\n", MAX_STUDENTS);
    if (!scanf("%d", &k) || (k > MAX_STUDENTS || k < 1))
    {
        printf("That is not a valid value...\n");
```

```

return 1;
}

int index = k - 1;
printf("The customers %d Name %s, ID %d and balance %d\n",
k, arr[index].name, arr[index].id, arr[index].balance);

getchar();
getchar();
}

```

I didn't include much handling for invalid input, this isn't my assignment. But here's how I would start going about it... Actually, personally I'd probably just ditch scanf() and go with fgets(). There's lots of functions for parsing data to a wanted datatype. I don't really program in C anyways.

?lassy - December 26, 2013, 2:03 am

(Dec 26, 2013 02:43 AM)TxTriton Wrote: [\[CITE\]](#) Why are you using k for indexing, if you continue with logic that dictates a 1-based array exists? If you're using k directly to index a value in the array and NOT k - 1, then shouldn't that be k >= 0, and k < 10?

Still, something seems off about this:

Code if(k >= 1 && <= 10)

Wouldn't you think? :) [hint]if(k >= 1 && <= 10)[/hint]

Code #include <stdio.h>

#define MAX_STUDENTS 10

typedef struct student

```

{
    int balance;
    int id;
    char name[20];
} student;

```

int main()

```

{
    student arr[MAX_STUDENTS];

    int i, k;
    for( i = 0; i < MAX_STUDENTS; i++ )
    {
        printf("Enter the customers %d Name ID and balance\n>>", i + 1);
        printf("Ex: John 1 250\n");
        scanf("%s %d %d", arr[i].name, &arr[i].id, &arr[i].balance);
    }
    getchar();

```

```

    printf("What customer details do you want see? [1-%d]\n", MAX_STUDENTS);
    if (!scanf("%d", &k) || (k > MAX_STUDENTS || k < 1))
    {
        printf("That is not a valid value...\n");
        return 1;
    }

```

```

int index = k - 1;
printf("The customers %d Name %s, ID %d and balance %d\n",
k, arr[index].name, arr[index].id, arr[index].balance);

getchar();
getchar();
}

```

I didn't include much handling for invalid input, this isn't my assignment. But here's how I would start going about it... Actually, personally I'd probably just ditch scanf() and go with fgets(). There's lots of functions for parsing data to a wanted datatype. I don't really program in C anyways.

Ah thanks for pointing those out, I was tired and misread OP's switch cases. My post is now fixed and should be equivalent to OP's switch statement.

I will refrain from posting after 2 AM. Nothing good happens after 2 AM. HIMYM Reference

hoer- - December 26, 2013, 3:06 am

st1[9] should be st1[10] because there are 10 customers :)

Sajuuk - December 26, 2013, 8:54 am

(Dec 26, 2013 04:06 AM)hoer- Wrote: [\[CITE\]](#) st1[9] should be st1[10] because there are 10 customers :)

Uhm, no? Because arrays start at the 0 position, so it should be st1[9].

hoer- - December 26, 2013, 10:07 am

(Dec 26, 2013 09:54 AM)Obsessed Wrote: [\[CITE\]](#) Uhm, no? Because arrays start at the 0 position, so it should be st1[9].

You must be confused, just because arrays start at 0 doesn't mean they should be defined like that. An array includes 0 as an element.

http://www.tutorialspoint.com/cprogramming/c_arrays.htm

InfectCode - December 26, 2013, 11:17 am

st1[9] is correct considering he only needs 10 customers. If he were to do st1[10], that position would hold a random integer. So why have an extra element in the array if its not needed?

Sajuuk - December 26, 2013, 11:23 am

(Dec 26, 2013 11:07 AM)hoer- Wrote: [\[CITE\]](#) You must be confused, just because arrays start at 0 doesn't mean they should be defined like that. An array includes 0 as an element.

http://www.tutorialspoint.com/cprogramming/c_arrays.htm

I'm confused? Huh, your post just proves my point. Look:

array[9] is array[0,1,2,3,4,5,6,7,8,9] <-- It's 10 elements, just what the OP needed.

0xDEAD10CC - December 26, 2013, 12:25 pm

Quote:Uhm, no? Because arrays start at the 0 position, so it should be st1[9].

Wrong. str1 9, gives you indexes 0-8, which is 9 customer slots total, NOT 10.

Quote:st1[9] is correct considering he only needs 10 customers. If he were to do st1[10], that position would hold a random integer. So why have an extra element in the array if its not needed?

Wrong again...

Quote:I'm confused? Huh, your post just proves my point. Look:

array[9] is array[0,1,2,3,4,5,6,7,8,9] <-- It's 10 elements, just what the OP needed.

Wrong yet again, index [9] doesn't belong to you, the result of using it, leads to undefined behavior.

Omg... ??? So many of you are wrong here... hoer- was right. For 10 students, the array should be:

Code student x[10];

Not [9]... The declaration is the # of elements! NOT, the last index.

When you write:

Code int arr[10];

You have 10 elements, indexed by the values from 0 to 9, which is 10 elements. Learn the basics? ? You can write at arr[10], but you have absolutely NO idea, what existed there before, which can lead to memory corruption, because that doesn't belong to you.

<http://www.cprogramming.com/tutorial/c/lesson8.html>

Quote:Let's look at the syntax for declaring an array.

```
int examplearray[100];
```

This would make an integer array with 100 slots (the places in which values of an array are stored). To access a specific part element of the array, you merely put the array name and, in brackets, an index number. This corresponds to a specific element of the array. The one trick is that the first index number, and thus the first element, is zero, and the last is the number of elements minus one. The indices for a 100 element array range from 0 to 99. Be careful not to "walk off the end" of the array by trying to access element 100!

Stop spreading false information.

edit: Perfect! It even gives you an example of a 10 element array on that page. ;)

Quote:Let me note again that you should never attempt to write data past the last element of the array, such as when you have a 10 element array, and you try to write to the [10] element. The memory for the array that was allocated for it will only be ten locations in memory, (the elements 0 through 9) but the next location could be anything. Writing to random memory could cause unpredictable effects--for example you might end up writing to the video buffer and change the video display, or you might write to memory being used by an open document and altering its contents. Usually, the operating system will not allow this kind of reckless behavior and will crash the program if it tries to write to unallocated memory.

I missed that he was using 9 in his original code, but I rewrote it with the correct # and seemed to have omitted that out in my response it seems..

hoer- - December 27, 2013, 4:12 am

(Dec 26, 2013 01:25 PM)TxTriton Wrote: [\[CITE\]](#) Wrong. str1 9, gives you indexes 0-8, which is 9 customer slots total, NOT 10.

Wrong again...

Wrong yet again, index [9] doesn't belong to you, the result of using it, leads to undefined behavior.

Omg... ??? So many of you are wrong here... hoer- was right. For 10 students, the array should be:

Code student x[10];
Not [9]... The declaration is the # of elements! NOT, the last index.

When you write:
Code int arr[10];

You have 10 elements, indexed by the values from 0 to 9, which is 10 elements. Learn the basics?
? You can write at arr[10], but you have absolutely NO idea, what existed there before, which can lead to memory corruption, because that doesn't belong to you.

<http://www.cprogramming.com/tutorial/c/lesson8.html>

Stop spreading false information.

edit: Perfect! It even gives you an example of a 10 element array on that page. ;)

I missed that he was using 9 in his original code, but I rewrote it with the correct # and seemed to have omitted that out in my response it seems..Thanks, He didn't even read that page I linked him, it doesn't anywhere declare array[9] for 10 elements. ?

imrankhan - December 28, 2013, 12:10 pm

thanks

i need to practice because half the code i don't even know

JamesOG - December 28, 2013, 1:26 pm

(Dec 26, 2013 12:17 PM)InfectCode Wrote: [\[CITE\]](#) st1[9] is correct considering he only needs 10 customers. If he were to do st1[10], that position would hold a random integer. So why have an extra element in the array if its not needed?

WHAT THE... Oh you're a "L33T member" ? ?

?lassy - December 28, 2013, 5:22 pm

(Dec 26, 2013 12:17 PM)InfectCode Wrote: [\[CITE\]](#) st1[9] is correct considering he only needs 10 customers. If he were to do st1[10], that position would hold a random integer. So why have an extra element in the array if its not needed?

Sir I must ask you to refrain from posting if you haven't researched the topic.

When you declare an array, the value you put inside the square brackets will indicate the amount of indexes the array has.

st1[9] would have 9 indexes, 0 - 8
st1[10] would have 10 indexes, 0 - 9

If you still don't believe me, try running this code.

PHP Code

```
<?php  
char test [10];  
printf("Size of the array: %d", (int)sizeof(test) );
```

Keep in mind, char takes up 1 byte of memory.