

Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Tư, 16 tháng 4 2025, 5:06 PM
Kết thúc lúc	Thứ Tư, 16 tháng 4 2025, 5:12 PM
Thời gian thực hiện	6 phút 30 giây

Câu hỏi 1

Đúng

Hiện thực hàm sau:

int calcSum(int *ptr, int n);

Tính và trả về tổng của các phần tử trong mảng 1 chiều được cho bởi con trỏ.

Trong đó:

ptr là con trỏ tới phần tử đầu tiên trong mảng.

n là kích thước của mảng.

Lưu ý: Bạn cần phải dùng dereference operator (*) để lấy giá trị của các phần tử trong mảng. Không được dùng subscript operator ([]).

Implement the following function:

int calcSum(int *ptr, int n);

Calculate and return the sum of elements of a 1-dimension array given by a pointer.

Where:

ptr is a pointer to the first element in the array.

n is the size of the array.

Note: You need to use the dereference operator (*) to get the values of the elements in the array. The subscript operator ([]) cannot be used.

For example:

Test	Result
int arr[] = {1, 2, 3, 4, 5}; cout << calcSum(arr, sizeof(arr) / sizeof(arr[0]));	15
int arr[] = {0, -1, 5, 6, -5, 1, -9, -10, -6, 3}; cout << calcSum(arr, sizeof(arr) / sizeof(arr[0]));	-16

Answer: (penalty regime: 0 %)

Reset answer

```

1 int calcSum(int *ptr, int n) {
2     int sum = 0;
3     int *endPtr = ptr + n; // Con trỏ chỉ vị trí sau phần tử cuối cùng
4
5     while (ptr < endPtr) {
6         sum += *ptr; // Cộng giá trị tại vị trí ptr vào tổng
7         ptr++;      // Di chuyển con trỏ đến phần tử tiếp theo
8     }
9
10    return sum;
11 }
```

	Test	Expected	Got	
✓	int arr[] = {1, 2, 3, 4, 5}; cout << calcSum(arr, sizeof(arr) / sizeof(arr[0]));	15	15	✓
✓	int arr[] = {0, -1, 5, 6, -5, 1, -9, -10, -6, 3}; cout << calcSum(arr, sizeof(arr) / sizeof(arr[0]));	-16	-16	✓

Passed all tests! ✓



Câu hỏi 2

Đúng

Hiện thực hàm sau:

```
void add(int *ptr, int n, int k);
```

Thực hiện thêm phần tử vào cuối của mảng 1 chiều được cho bởi con trỏ.

Trong đó:

ptr là con trỏ tới phần tử đầu tiên trong mảng.

n, k lần lượt là kích thước của mảng và phần tử cần được thêm vào.

Implement the following function:

```
void add(int *ptr, int n, int k);
```

Insert element to the end of the 1-dimension array given by a pointer.

Where:

ptr is a pointer to the first element in the array.

n, k respectively is the size of the array and the element that need to be added.

For example:

Test	Result
<pre>int arr[100] = {1, 2, 3, 4, 5}; int n = 5; int k = 7; add(arr, n, k); cout << arr[n];</pre>	7
<pre>int arr[100] = {3, 9, 20, 6, 18, 0, 16, 8, 15, 14}; int n = 10; int k = 0; add(arr, n, k); cout << arr[n];</pre>	0
<pre>int arr[100] = {15, 8, -14, 13, -17, -12, 10, -15, -9, 4, 1, 0, 16, -11, -5, 19, 17, -13, -18, -4, 0}; int n = 22; int k = -100; add(arr, n, k); cout << arr[n];</pre>	-100

Answer: (penalty regime: 0 %)

Reset answer

```
1 void add(int *ptr, int n, int k) {
2     // Di chuyển con trỏ đến vị trí cuối mảng (vị trí thứ n)
3     ptr = ptr + n;
4
5     // Gán giá trị k vào vị trí cuối mảng
6     *ptr = k;
7 }
```



	Test	Expected	Got	
✓	<pre>int arr[100] = {1, 2, 3, 4, 5}; int n = 5; int k = 7; add(arr, n, k); cout << arr[n];</pre>	7	7	✓
✓	<pre>int arr[100] = {3, 9, 20, 6, 18, 0, 16, 8, 15, 14}; int n = 10; int k = 0; add(arr, n, k); cout << arr[n];</pre>	0	0	✓

Passed all tests! ✓

