Review Test Submission: Midterm Sample #3: Functions

User	Yihsun Lee
Course	Object-Oriented Software Development Using C++
Test	Midterm Sample #3: Functions
Started	6/22/21 8:59 PM
Submitted	6/22/21 8:59 PM
Due Date	6/23/21 11:59 PM
Status	Needs Grading
Attempt Score	Grade not available.
Time Elapsed	0 minute out of 30 minutes
Instructions	You are allowed to use any tool and material available to you during the test. The following are considered plagiarism: • exchanging messages in any form with another person during the test. • allowing access to your questions/solutions to somebody else before the test due date. • acquiring the test questions/solutions that somebody else had before the test due date.
Results Displayed	Submitted Answers

Question 1

Needs Grading

Define a family of functions (a templated function) named `insertAtEnd` that inserts in a **dynamically-allocated array** of elements of any type another element at the end (resize the array). The function should receive as parameters:

- the dynamically-allocated array
- the size of the array
- the element to insert

The function should return the resulted array.

Specialize the function for the type `char`. In this specialization the array must be null-terminated; the size parameter doesn't count the null-byte.

The client code listed below uses your templated function, and should not contain memory leaks. The comments next to each statement shows the content the array should have **after** the statement is executed.

A type that uses this template must include in its definition certain functions and/or operators. **Identify each function and/or operator that your template assumes is defined.** You may do so in the form of an exact prototype or an English descriptive phrase.

Write your solution in the textbox below.

```
// assume all necessary headers have been included

int main()
{
    {
        int* arrI = nullptr;
        arrI = insertAtEnd(arrI, 0, 1); // 1
        arrI = insertAtEnd(arrI, 1, 5); // 1, 5
        arrI = insertAtEnd(arrI, 2, -3);// 1, 5, -3
        delete[] arrI;
    }
    {
        double* arrD = nullptr;
        arrD = insertAtEnd(arrD, 0, 1.2);// 1.2
        arrD = insertAtEnd(arrD, 1, 2.3);// 1.2, 2.3
        arrD = insertAtEnd(arrD, 2, 3.4);// 1.2, 2.3, 3.4
        delete[] arrD;
    }
    {
        char* arrC = nullptr;
        arrC = insertAtEnd(arrC, 0, 'a');// a\0
        arrC = insertAtEnd(arrC, 1, 'b');// ab\0
        arrC = insertAtEnd(arrC, 2, 'c');// abc\0
        cout << arrC;
        delete[] arrC;
    }
}
```

Selected Answer:

```
//
//  insertAtEnd.hpp
//  practice_functions
//
//  Created by YiHsun  on 2021-06-21.
//

#ifndef SDDS_insertAtEnd_h
#define SDDS_insertAtEnd_h

#include <iostream>
namespace sdds{

template<typename T, typename V>
T* insertAtEnd(T* arr, size_t size, V val){
   T* newArr = new T[size+1];
   for(size_t i = 0; i < size; i++){
      newArr[i] = arr[i];
   }
   newArr[size] = val;
   delete[] arr;
   arr = newArr;
   for(size_t i = 0; i < size+1; i++){
      std::cout << arr[i] << " ";
   }
   std::cout << std::endl;
   return arr;
}

template <>
char* insertAtEnd(char* arr, size_t size, char val){
   char* temp = new char[size+2];
   for(size_t i = 0; i < size; i++){
      temp[i] = arr[i];
   }
   temp[size] = val;
   temp[size+1] = '\0';
   delete[] arr;
   arr = temp;
   return arr;
}

}



#endif /* SDDS_insertAtEnd_h */
```

← OK