Assignment for data structure

First step is make a header file of array data structure:

First of all we make a header file of array data structure which requires a suitable code for creating a header file in the program which are as follows:

#include "array.h"

I

int main() {

Array myArray = create\_array(5);

for (size\_t i = 0; i < get\_array\_size(&myArray); i++) {

set\_value(&myArray, i, (int)(i \* 10));

}

for (size\_t i = 0; i < get\_array\_size(&myArray); i++) {

printf("Element at index %zu: %d\n", i, get\_value(&myArray, i));

}

free\_array(&myArray);

return 0;

}

2. Header File array.ccp:

Now we make a header file of header file array.ccp by writing the following code are as follows;

#include "array.cpp"

int main() {

Array MyArray(5);

for (size\_t i = 0; i < myArray.getSize(); i++) {

myArray.setValue(i, static\_cast<int>(i \* 10));

}

for (size\_t i = 0; i < myArray.getSize(); i++) {

std::cout << "Element at index " << i << ": " << myArray.getValue(i) << std::endl;

}

return 0;

}

3. Header File of LinkedList.h:

Now we make a header file by using linked list data structure.

First we make a header file of linkedlist data structure which have name Linkedlist.h are as follow:

code:

#include "linkedlist.h"

int main() {

LinkedList<int> list;

// Inserting elements

list.insert(10);

list.insert(20);

list.insert(30);

// Displaying the list

std::cout << "Linked List: ";

list.display();

// Removing an element

list.remove(20);

std::cout << "After removing 20: ";

list.display();

// Displaying the size of the list

std::cout << "Size of the list: " << list.getSize() << std::endl;

return 0;

}

**4 header file of linkedlist.ccp:**

Here are code of creating header file linkedlist.ccp are as follows:

#include "linkedlist.ccp"

int main() {

LinkedList<int> list;

// Insert elements

list.insert(10);

list.insert(20);

list.insert(30);

// Display the list

std::cout << "Linked List: ";

list.display();

// Remove an element

list.remove(20);

std::cout << "After removing 20: ";

list.display();

// Display the size of the list

std::cout << "Size of the list: " << list.getSize() << std::endl;

return 0;

}