Data Structure Project Report

On

" Convert an Array to a Doubly Linked List"

Submitted in the Partial fulfillment of the requirement for the Award of Degree of

Bachelor of Technology

in

COMPUTER SCIENCE & ENGINEERING

Batch
(2020-24)

Submitted to:	**Submitted by:**
Er.Navneet Bawa	**Sumit Kumar Giri 2000213** **Sanjay Kumar Sah 2000193**

Amritsar College of Engineering & Technology, Amritsar

(Autonomous college under UGC Act – 1956[2(f) and 12(B)])

ACKNOWLEDGEMENTS

This is a humble effort to express our sincere gratitude towards those who have guided and helped us to complete this project

A project is major milestone during the study period of a student. As such this project was a challenge to us and was an opportunity to prove our caliber. we are highly grateful and obliged to each and everyone making us help out of problems being faced by us.

It would not have been possible to see through the undertaken project without the guidance of **Navneet Bawa**. It was purely on the basis of their experience and knowledge that we able to clear all the theoretical and technical hurdles during the development phases of this project work.

Last but not the least we are very thankful to our Head of Department and all Members of Computer Science Deptt. who gave us an opportunity to face real time problems while fulfilling need of an organization by making projects for them.

DECLARATION

We Name hereby declare that the project work entitled **"Convert an Array to a Doubly Linked List "** is an authentic record of my own work carried out as requirements of Institutional Training project for the award of degree of B.Tech(CSE), **Amritsar College of Engg. And Technology, Amritsar,** under the guidance of **Navneet Bawa.**

Name of Students:

Sumit kumar giri 2000213

Sanjay Kumar sah 2000193

Certified that the above statement made by the student is correct to the best of our knowledge and belief.

Faculty Coordinator:

Er.Navneet Bawa(CSE Department)

INDEX PAGE

Introduction to Data Structures.

• ***Data structure is representation of the logical relationship existing between individual elements of data.***

• ***In other words, a data structure is a way of organizing all data items that considers not only the elements stored but also their relationship to each other.***

• Data structure affects the design of both structural & functional aspects of a program.

• Program=algorithm + Data Structure

• You know that a algorithm is a step by step procedure to solve a particular function.

• That means, algorithm is a set of instruction written to carry out certain tasks & the data structure is the way of organizing the data with their logical relationship retained.

• To develop a program of an algorithm, we should select an appropriate data structure for that algorithm.

• Therefore algorithm and its associated data structures from a program.

Data structure are normally divided into two broad categories:

• Primitive Data Structure

• Non-Primitive Data Structure

-PRIMITIVE DATA STRUCTURE:

• There are basic structures and directly operated upon by the machine instructions.

• In general, there are different representation on different computers.

• Integer, Floating-point number, Character constants, string constants, pointers etc, fall in this category.

-NON-PRIMITIVE DATA STRUCTURE:

• There are more sophisticated data structures.

• These are derived from the primitive data structures.

• The non-primitive data structures emphasize on structuring of a group of homogeneous (same type) or heterogeneous (different type) data items.

• Lists, Stack, Queue, Tree, Graph are example of non-primitive data structures.

• The design of an efficient data structure must take operations to be performed on the data structure.

The most commonly used operation on data structure are broadly categorized into following types:

• Create

• Selection

• Updating

• Searching

- Sorting
- Merging
- Destroy or Delete

DIFFERENT BETWEEN THEM

- A primitive data structure is generally a basic structure that is usually built into the language, such as an integer, a float.
- A non-primitive data structure is built out of primitive data structures linked together in meaningful ways, such as a or a linked-list, binary search tree, AVL Tree, graph etc.

OBJECTIVE OF THE PROJECT:

The Objective of this algorithm is to convert an Array into a Doubly Linked List

Working:

Create a for loop to go through the array's of numbers. You will be able use the index variable to track what value you want. Take that value and save it to the current node you are working with. Make sure to set the current node's previous and next nodes as well

Algorithm

The idea is to start traversing the array and for every array element create a new list node and assign the prev and next pointers of this node accordingly.

Create a pointer p_head to point to the starting of the list which will initially point to NULL(Empty list).

For the first element of the array, create a new node and put that
node's prev and next pointers to point to p_head to maintain the fashion of the list.

For the rest of the array elements, insert those elements to the end of the created doubly linked list.

Code :

```
#include<bits/stdc++.h>
using namespace std;

struct DoubleLinkedListItem
{
   int data;
   struct DoubleLinkedListItem *p_prev;
   struct DoubleLinkedListItem *p_next;
};

struct DoubleLinkedListItem * converArrayToDoubleLinkedList(const int arr[], size_t arr_size)
{
   struct DoubleLinkedListItem *p_head = NULL;
   do
   {
      struct DoubleLinkedListItem *p_current = NULL;
      size_t i = 0;
      if (arr_size <= 0) //simple check
      {
         cout<<"Warning: size of array equal or less 0!";
         break;
      }
      for(; i < arr_size; i++)
      {
         struct DoubleLinkedListItem *p_item = new DoubleLinkedListItem;
         if (NULL == p_item) // simple check
         {
            cout<<"Some error occur during allocating, exit";
            break;
         }

         p_item->data = arr[i];
         if (NULL == p_current)
```

```
  {
            p_current = p_item;
            p_current->p_prev = NULL;
          p_current->p_next = NULL;
            p_head = p_current;
        } else
        {
            p_current->p_next = p_item;
            p_item->p_prev = p_current;
            p_item->p_next = NULL;

            p_current = p_item;
        }
      }
   } while(0);

   return p_head;
}

int main()
{
   int test_arr[] = {1,2,3,4,5,6,7};
   int i = 0;
   struct DoubleLinkedListItem *p_head = NULL;
   cout<<"Elements in the test Array are :\n";
   for (; i < sizeof(test_arr)/sizeof(test_arr[0]); i++)
   {
      cout<<test_arr[i]<<" ";
   }
   cout<<endl;

   p_head = converArrayToDoubleLinkedList(test_arr, sizeof(test_arr)/sizeof(test_arr[0]));

   cout<<"After Processing Elements in the Doubly Linked List are :\n";
   if (NULL != p_head)
   {
      do
      {
         struct DoubleLinkedListItem *p_for_free = p_head;
```

```
            cout<<p_head->data<<"   ";
            p_head = p_head->p_next;
            free(p_for_free);
        } while(p_head);
        p_head = NULL;

    } else
    {
        cout<<"empty linked list\n";
    }
    cout<<endl;
    return 0;
}
```

OUTPUT:

```
Elements in the test Array are :
1 2 3 4 5 6 7
After Processing Elements in the Doubly Linked List are :
1 2 3 4 5 6 7
```

References

- **Books**
  - Data structures by Anshuman Sharma
- **Web URLs**

  https://www.upgrad.com/blog/data-structure-project-ideas-beginners/

  https://www.geeksforgeeks.org/create-linked-list-from-a-given-array/