SIT102 Introduction to Programming

Pass Task 6.2: Hand Execution of Arrays

Overview

In this task you will demonstrate how arrays work by hand executing a number of small code snippets.

Submission Details

Use the instructions on the following pages to demonstrate how arrays work.

Submit the following files to OnTrack.

- Images of your hand execution (fit program 1 and 2 on one page, and 3 and 4 on another)

The focus of this task is on understanding and demonstrating the use of arrays, the use of indexes, and control flow associated with these.

You will need to discuss your answer to this task with the tutor in order for it to be signed off, so leave an audio comment or discuss this with them in class.

Instructions

Watch the Hand Execution with Arrays video, this shows how to step through code that makes use of arrays. You can apply the same process and visualisations for working with vectors as well.

Use the process shown to demonstrate how the following code snippets execute with the provided input values.

Program 1:

Demonstrate how the following code works with the supplied values:

```cpp
double _____ ( const vector<double> &data )
{
    int result = 0;

    for(int i = 0; i < data.size(); i++)
    {
        if ( data[i] == 3 )
        {
            result++;
        }
    }

    return result;
}
```

1. Include the following on the first image:

    1. Hand execute the above with a vector that has the following data

        `data` :

        ```
        |----|----|----|----|----|
        |  3 |  2 |  8 |  0 |  0 |
        |----|----|----|----|----|
        ```

    2. Hand execute the above with a vector that has the following data

        `data` :

        ```
        |----|----|----|----|----|
        |  1 | -7 |  3 |  2 |  3 |
        |----|----|----|----|----|
        ```

    3. In the image, note an appropriate name you could use for this function

Program 2:

```cpp
void _____ ( vector<double> &data )
{
    for(int i = 0; i < data.size(); i++)
    {
        if ( data[i] < 0 )
        {
            data[i] = data[i] * -1;
        }
    }
}
```

Note that this procedure *updates* the passed value(s) of the vector.

1. Include the following on the second image:

    1. Hand execute the above with a vector that has the following data

        data :

        ```
        |----|----|----|----|----|
        |  2 | -5 |  0 |  3 | -9 |
        |----|----|----|----|----|
        ```

    2. Hand execute the above with a vector that has the following data

        data :

        ```
        |----|----|----|----|----|
        |  0 |  1 |  3 | -10|  7 |
        |----|----|----|----|----|
        ```

    3. In the image, note an appropriate name you could use for this function

Program 3:

```cpp
void _____ ( vector<double> &data, int idx )
{
    if ( idx >= 0 and idx < data.size() )
    {
        int last = data.size() - 1;
        data[idx] = data[last];
        data.pop_back();
    }
}
```

Note that this procedure *updates* the passed value(s) of the vector. Remember that `pop_back()` removes the last element from the vector. You can represent this by crossing out the whole box from the vector.

1. Include the following on the third image:

    1. Hand execute the above with the following values:

        data :

        ```
        |----|----|----|----|----|
        |  3 | -9 |  0 |  1 | 33 |
        |----|----|----|----|----|
        ```

        idx :

        ```
        2
        ```

    2. Hand execute the above with the following values:

        data :

        ```
        |----|----|
        |  0 | -4 |
        |----|----|
        ```

        idx :

        ```
        1
        ```

    3. Hand execute the above with the following values:

        data :

        ```
        |----|----|
        |  2 |  1 |
        |----|----|
        ```

        idx :

4

4. In the image, note an appropriate name you could use for this function

Program 4:

```cpp
void _____ ( vector<double> &data, int idx, double value )
{
    data.push_back(value);

    if ( idx >= data.size() - 1 ) return;
    if ( idx < 0 ) idx = 0;

    for(int i = data.size() - 1; i > idx; i--)
    {
        data[i] = data[i - 1];
        data[i - 1] = value;
    }
}
```

Note that this procedure *updates* the passed in vector. This will change the value in a few of the vector elements, so make sure to give yourself lots of space in those boxes. Take care with this one...

1. Include the following on the fourth image:

    1. Hand execute the above with the following values:

        data :

        ```
        |----|----|----|----|----|
        |  5 | -5 |  1 |  7 | -2 |
        |----|----|----|----|----|
        ```

        idx :

        2

        value :

        -6

    2. In the image, note an appropriate name you could use for this function

Task Discussion

Discuss the following with your tutor to demonstrate your understanding of the concepts covered.

- Explain how an element can be removed from an array.
- Reflect upon different ways that the index can be used to work with values in the array.