Home Work No.12 (Arrays)

Before going into homework codes, you must have following concepts regarding arrays:

Arrays:

- Are ***Grouped*** data. It have more than one data cells or ***elements***.
- All elements have of ***similar data*** type i.e., we can create an array of multiple elements with similar data type.
- Hold ***Contiguous/continuous/adjacent*** Memory locations
- Are **static** and ***fixed*** sized. These occupy memory at compile time. Always constants or literals are used to create an array.
- Each array maintains an internal addressing called ***indexing***. Each element of an array is assigned an index or subscript number. In C++ indices are started from zero to size-1. Do not confuse indexing with physical memory addresses these both are entirely different. Physical memory address where an array occupy memory.
- Indexing is one the ***power*** of any array which makes a programmer to access and use and array very easily and conveniently.
- ***Subscript operator []*** is used to access each or any element of an array.

Things to Remember:

- ✓ Array **size** is always be **constant** and of **integer** type.
- ✓ Each element/cell size of an array depends upon type of an array.
- ✓ In C++, indices are started from **zero** to **size-1**.
- ✓ Array name is **Constant Pointer** [i.e., it always point to the first element of an array]. We will discuss later on what is meant by constant pointer.
- ✓ When we print name of array its will print starting bytes physical/memory address of array (except character array where it prints content/data or character of array until null character]
- ✓ In C++ no bound checks so be careful during writing you code. Always write a loop from starting index i.e. zero to ending index i.e. size-1.
- ✓ Be careful of ***off-by-one*** error.

///... code1

```
int main()
{
        const int NUM_EMPLOYEES = 6; // Number of employees
        int hours[NUM_EMPLOYEES]; // creating and array of hours using number employees as 
size

        // Assigning the hours worked using subscript Notation

        hours[0] = 11; // assigning value to 1st element of array at index 0
        hours[1] = 22; // assigning value to 2nd element of array at index 1
        hours[2] = 36; // assigning value to 3rd element of array at index 2
        hours[3] = 14; // assigning value to 4th element of array at index 3
        hours[4] = 15; // assigning value to 5th element of array at index 4
        hours[5] = 65; // assigning value to 6th element of array at index 5


        // Display the contents of the array.
        cout << "The hours you entered are:"<<endl;

        cout << hours[0] << endl;
        cout << hours[1] << endl;
        cout << hours[2] << endl;
        cout << hours[3] << endl;
        cout << hours[4] << endl;
        cout << hours[5] << endl;

        return 0;
}
```

///... code1 use of initializer list for initializing hours array

```
int main()
{
        const int NUM_EMPLOYEES = 6; // Number of employees
        // creating and array of hours using number employees as size and initialized 
using initilizer List

        int hours[NUM_EMPLOYEES] = { 2,33,4,55,6,7 };

        // Display the contents of the array.
        cout << "The hours you entered are:"<<endl;

        cout << hours[0] << endl;
        cout << hours[1] << endl;
        cout << hours[2] << endl;
        cout << hours[3] << endl;
        cout << hours[4] << endl;
        cout << hours[5] << endl;



        return 0;
}
```

///... code3 loops are used to Read array and print array elements

```
int main()
{

	const int NUM_EMPLOYEES = 6; // Number of employees
	int hours[NUM_EMPLOYEES]; // creating and array of hours using number employees as 
size

	//using loop to take input in hours array

	for (int i = 0; i < NUM_EMPLOYEES; i++)
	{
		cout << endl << "IPUT  " << i + 1 << "  Element  : ";
		cin >> hours[i];
	}

	//Printing values hours using loop

	for (int i = 0; i < NUM_EMPLOYEES; i++)
		cout << "\n\nHours of  Employee " << i + 1 << "  is ::::   "<< hours[i];


	return 0;

}
```

///... code4 calculating total hours worked by adding all array elements

```
int main()
{
	const int NUM_EMPLOYEES = 6; // Number of employees
	int hours[NUM_EMPLOYEES]; // creating and array of hours using number employees as 
size

	int total_hours = 0; //Variable to calculate total hours


	//using loop to take input in hours array

	for (int i = 0; i < NUM_EMPLOYEES; i++)
	{
		cout << endl << "IPUT  " << i + 1 << "  Element  : ";
		cin >> hours[i];
	}

	//calculating Total number of hours for each employee

	for (int i = 0; i < NUM_EMPLOYEES; i++)
		total_hours += hours[i];

	cout << "\n\nTotal Hours Worked are  =  " << total_hours;
```

```
        return 0;

}
```

Problem 1:

Convert code 4 to find average hours of work.

Problem 2:

Convert code 4 to find how many employee have even hours and how many have odd hours of work.

Problem 3:

Convert code 4 Sort array hours array into ascending and descending order.

Problem 4:

Convert code 4 to find highest and lowest from the hours array.

///………

1) Answer the following:

A) Define the following arrays and print them using while loops:

a) ages, a 10-element array of ints initialized with the values 5, 7, 9, 14, 15, 17, 18, 19, 21, and 23.
b) temps, a 7-element array of floats initialized with the values 14.7, 16.3, 18.43, 21.09, 17.9, 18.76, and 26.7.
c) alpha, an 8-element array of chars initialized with the values J , B , L, A,* , $, H , and M .

B) Is each of the following a valid or invalid array definition? (If a definition is invalid, explain why.)

a) int numbers[10] = {0, 0, 1, 0, 0, 1, 0, 0, 1, 1};
b) int matrix[5] = {1, 2, 3, 4, 5, 6, 7};
c) double radii[10] = {3.2, 4.7};
d) int table[7] = {2, , , 27, , 45, 39};
e) char codes[] = {'A', 'X', '1', '2', 's'};
f) int blanks[];

C) Given the following array definition:

int nums[5] = {1, 2, 3};

What will the following statement display?
cout << nums[2]<<" " << nums[4];

///..

2) What is the problems/Errors with following codes.

1)

```
int a = 10;

int array1[a];

for (int i = 0; i < 10; i++)
{
        cout << array1[i];
}
```

....

2)

```
const int size = 10;
int array[size];

//initializing array  using loops

for (int i = 1; i <= 10; i++)
{
        cin >> array[i];
}
```

....

3)

```
int array2[5] = { 1,2,3,4,5,6 };

for (int i = 0; i < 5; i++)
{
        cin >> array2[i];
}
```

....

4)

```
int array[0] = {};
```

....

5)

```
const int SIZE = 4;
int oldValues[SIZE] = { 10, 100, 200, 300 };
int newValues[SIZE];


newValues = oldValues;// assigning all the values of oldValues array to a newValues array
```

....

How to correct in code segment 5)

....

6)

```
int table[10];
for (int x = 0; x < 20; x++)
{
cout << "Enter the next value: ";
cin >> table[x];
}
```

....

How to correct in code segment 6)

....

7)

```
char greeting[] = {'H', 'e', 'l', 'l', 'o'};
cout << greeting;
```

....

How to correct in code segment 7)

....

///..

3) Dry run following codes and show what will be output of following codes!

```
////////Code No.1 ///////////////////////////////////////////

const int MONTHS = 12;
int days[MONTHS] = { 31, 28, 31, 30,
       31, 30, 31, 31,
       30, 31, 30, 31 };
```

```
    for (int count = 0; count < MONTHS; count++)
    {
        cout << "Month " << (count + 1) << " has ";
        cout << days[count] << " days.\n";
    }



    ////////Code No.2 ///////////////////////////////////////////

    const int size = 7;
    int numbers[size] = { 1, 2, 4, 8 };

    cout << endl << "Size of numbers array is  : " << sizeof(numbers);
    cout << endl<< "Address of First Byte of numbers array is  : " <<numbers;
    cout<<endl;
    for (int i = 0; i < size; i++)
    {
        cout << "Value at index" << i + 1 << "  is  :";
        cout << numbers[i] << endl;
    }




    ////////Code No.3 ///////////////////////////////////////////


        const int size = 10;
        int array[size];

        //initializing array  between 10 to 30 with random functions

        srand(time(0));

        for (int i = 0; i < size; i++)
        {

            array[i] = 10 + rand() % 20;
        }

        cout << endl << endl;

        for (int i = 0; i < size; i++)
        {
            cout <<  array[i] << "  ";
        }



////////Code No.4 ///////////////////////////////////////////
// Challenge code can u guess what following code is doing

int main()
{
	const int SIZE = 10;
	int array1[SIZE];

	int value1, value2;
	cout << "\n Enter 10 values for array\n\n";

	for (int i = 0; i <= 9; i++)
	{
		cout << "Enter  " << i + 1 << " Value of array 1 :  ";
		cin >> array1[i];

	}

	cout << "\n\nYour Entered array is ";
	for (int i = 0; i <= 9; i++)
	{
		cout << " " << array1[i] << " ";

	}
	//// ..... Guess what following code is doing

	value1 = array1[0];

	for (int count1 = 1; count1 < SIZE; count1++)
	{
		if (array1[count1] > value1)
			value1 = array1[count1];
	}


	cout << "\n\nFirst Mysterious Value is .....  " << value1;

	value2 = array1[0];

	for (int count2 = 1; count2 < SIZE; count2++)
	{
		if (array1[count2] < value2)
			value2 = array1[count2];
	}


	cout << "\n\nSecond Mysterious Value is .....  " << value2;


	cout << "\n\n\n\t\t CAN YOU GUESS WHAT ARE THESE TWO MESTERIOUS VALUES!!!\n\n\n";

	return 0;
}


////////Code No.5 ///////////////////////////////////////////
// Challenge code can u guess what following code is doing

int main()
{

	const int SIZE = 7;
```

```
        int array2[SIZE];


        int low = 0, high = 6;

        cout << "\n Enter 7 values for array\n\n";

        for (int i = 0; i <= 6; i++)
        {
                cout << "Enter  " << i + 1 << " Value of array 1 :  ";
                cin >> array2[i];

        }

        cout << "\n\n Array at  start\n ";

        for (int i = 0; i <= 6; i++)
        {
                cout << " " << array2[i] << " ";

        }

        //// ..... Guess what following code is doing
        while (high >= low)
        {
                int temp = array2[low];
                array2[low] = array2[high];
                array2[high] = temp;
                high--;
                low++;
        }
        //////................

        cout << "\n\n Array After processig\n ";

        for (int i = 0; i <= 6; i++)
        {
                cout << " " << array2[i] << " ";

        }

        cout << endl;


        return 0;
}
//Convert above code for 8 sized array
```

///...

/////// Questions from Some Previous Exams

Understand and then explain what is happening in all codes below.

1) What will be output of following code?

```
int main()
{

        int count = 0;
        int array1[] = { 119,101,108,108,32,100,111,110,101,33 };
        char array2[11];

        cout << "Size of integer array is " << sizeof(array1) << "    bytes." << endl;

        cout << "Size of Character array is " << sizeof(array2) << "  bytes." << endl;

        while (count <= 9)
        {
                array2[count] = array1[count];
                count++;
        }
        array2[count] = '\0';


        cout << array2 << endl;


}
```

2) What will be output of following code?

```
        int number = 3;
        int array1[20];
        for (int i = 0; i < 20; i++)
        {
                array1[i] = number;
                number += 3;
        }

        for (int i = 0; i < 20; i++)
        {
                if (array1[i] % 2 == 0 && array1[i] % 3 == 0 && array1[i] % 5 == 0)
                cout << array1[i] << " : This number is divisible by 2, 3 and 5. " << endl;
        }
```

3) What will be output of following code?

```
        int main() {
                int nrows = 3, ncols = 4;
                int A[2][3][4] = { {

                        { 1, 3, 2 },
                        { 4, 5 },
                        { 7, 8, 9 } },
                        { { 4 },
                        { 5, 5, 7 },
                        { -2, 3, 4 } }
                };
```

```
    int b[2][4] = { { 0 } };
    for (int i = 0; i < 2; ++i) {

        for (int j = 0; j < ncols; ++j)
            for (int k = 0; k < nrows; ++k)
                b[i][j] += A[i][k][j];
    }
    for (int i = 0; i < 2; ++i) {
        for (int j = 0; j < nrows; ++j) {
            for (int k = 0; k < ncols; ++k)
                cout << A[i][j][k] << " ";
            cout << endl;
        }
        for (int j = 0; j < ncols; ++j)
            cout << b[i][j] << " ";
        cout << endl;
    }
    return 0;
}
```

Int a[5];

Int b[5];

Int c[5];