CCC 204 Data Structures and Algorithms
LABORATORY REPORT :
LAB 3# - Arrays

Samano, Rajel Johann
Information Technology Program, CABECS
Colegio San Agustin – Bacolod

I. INTRODUCTION

For Laboratory Activity Number three is to follow the objectives of the exercises and answer questions. The Three main objectives as follows:

- Define and initialize array data structures.
- Manipulate array elements.
- Develop programs that use the main topic or concept introduced in the exercise.

II. IMPLEMENTATION / APPROACH

Figure 1. histoGram.c with output

```c
#include <stdio.h>

int main() {
    //Initialize a 5-element array and have the values 5, 4, 3, 2, 1.
    int index[5] = {5, 4, 3, 2, 1};
    printf("Index     Value      Histogram\n");

    //Loop through the array to display the index and value of each element.
    for (int i = 0; i < 5; i++) {
        printf("  %d       %d       ", i, index[i]);

        //For loop to print out the amount of "$" it needs for each value.
        for (int j = 0; j < index[i]; j++) {
            printf("$");
        }
        printf("\n");
    }

    return 0;
}
```

```
C:\Codes\Coding\C>hg
Element/Index     Value      Histogram
     0              5          $         $         $         $         $
     1              4          $         $         $         $
     2              3          $         $         $
     3              2          $         $
     4              1          $
```

The task was to "**Display the array index, value, and histogram bar for value, represented by: '$' character. Limit array size to five (5)**." My approach towards the task was to start by making an array with the size of five then after that was done I went to Google to find some ideas on how to print an array using a for loop after I had an idea about it I began to experiment to find whether it was correct after multiple attempts to edit it finally showed an output that was close enough to the outcome that was showed in the table in the exercises array after modifying the code it was time to add another for loop were it is supposed to print out an "$" after trying by myself I went to google after to have a hint to what to do after finding a code that is close enough and after modifying my code it works as intended.

Figure 2. chessboard.c with output

```c
#include <stdio.h>

int main() {
    //Declaring variables Needed for the code.
    int rows[1], cols[1], process = 1;
    char chessBoard[100][100], input[2], char1, char2;

    while (process) {
        printf("ChessBoard Pattern\n");

        //Input for the number of rows and columns.
        printf("Number of Rows: ");
        scanf("%d", &rows[1]);
        printf("Number of Columns: ");
        scanf("%d", &cols[1]);

        //Input for characters to display on the chessboard.
        printf("Char1 to Display: ");
        scanf(" %c", &char1);
        printf("Char2 to Display: ");
        scanf(" %c", &char2);

        //Displaying the configuration of the chessboard.
        printf("Generating Size of Chessboard by %d x %d with the pattern of %c and %c.\n", rows[1], cols[1], char1, char2);
        printf("\n");

        //Generating the chessboard pattern.
        for (int i = 1; i <= rows[1]; i++) {
            for (int j = 1; j <= cols[1]; j++) {
                if ((i + j) % 2 == 0) {
                    chessBoard[i][j] = printf(" %c ", char1);
                } else {
                    chessBoard[i][j] = printf(" %c ", char2);
                }
            }
            printf("\n");
        }
        //While loop to repeat it again and change the size or character display.
        printf("\nDo you want to go again? (Y/N): ");
        scanf(" %c", input);

        if (input[0] == 'n' || input[0] == 'N') {
            process = 0;
            printf("\n");
        }
    }
    return 0;
}
```

```
C:\Codes\Coding\C>ch
ChessBoard Pattern
Number of Rows: 5
Number of Columns: 5
Char1 to Display: W
Char2 to Display: B
Generating Size of Chessboard by 5 x 5 with the pattern of W and B.

 W  B  W  B  W
 B  W  B  W  B
 W  B  W  B  W
 B  W  B  W  B
 W  B  W  B  W
 B  W  B  W  B

Do you want to go again? (Y/N): n
```

The task was to "**Print out a chessboard pattern, using two-dimensional array. Let character 'B' stand for black boxes and 'W' for white boxes. Final program should display row x column chessboard pattern. User input controls the size of pattern, and what character to display**." My approach to this was first to look at google as how to do this as I had no idea how to do it I search c language chessboard pattern it was almost there but after modifying and trying some other stuff then after trying a couple of times failing then modifying it little by little I also added a while loop that I had saved in my previous CCC subject I just had to do switch it over to c language as it was in java.

III. EXPERIMENTAL FINDINGS / DISCUSSIONS

Figure 3. Exercise1

```
#include <stdio.h>
int main(void)
{
 // use initializer list to initialize array n
int n[5] = {32, 27, 64, 18, 95};
printf("Element Value\n");
// output contents of array in tabular format
for (int i = 0; i < 5; ++i) {
printf("%d %d\n",i,n[i]);
}
}
```

1. What happens when array elements are more than five?

Ans: The code still works but it does not see the sixth element as it is over the array size when I tried to run the code it showed a "warning: excess elements in array initializer" and it had a "note: (near initialization for 'n')." that's what I had found.

2. What happens when the array is referenced with an index greater than five?

Ans: The code works perfectly if it is the modified code for question 1 then if it is not changed and the original code is used then it would work as it is over the placed elements in the array the number of initialized elements is less than the size of the array specified, then the rest of the elements will automatically be initialized to 0 by the compiler is what i had found in google.

Figure 4. Exercise2

```c
#include <stdio.h>
#define SIZE 12
int main(void)
{
 int a[SIZE] = {1, 3, 5, 4, 7, 2, 99, 16, 45, 67, 89, 45};
 int total = 0;
 for (size_t i = 0; i < SIZE; ++i)
 {
 total += a[i];
 }
 printf("Total of array element values is %d\n",total);
 printf("%d\n",SIZE);
}
```

1. What is the purpose of #define?

Ans: The #define is a directive that allows constant values to be declared for use throughout your code. You generally use this syntax when creating constants that represent numbers, strings, or expressions it cannot also be changed by your program code like variables.

2. In what situations is setting the array size using #define advantageous?

Ans: The use of #define is that it is like a variable but it cannot be changed and it also can be used multiple times as if there are more of the same variables then it would work if not modified to a degree.

Figure 5. Exercise3

```c
#include <stdio.h>
int main ()
{
 /* an array with 5 rows and 2 columns*/
 int a[5][2] = { 0 , 0, 1, 2, 2, 4, 3, 6, 4, 8};
 int i, j;
 /* output each array element's value */
 printf("Array [m][n] Value\n");
 for ( i = 0; i < 5; i++ )
 {
 for ( j = 0; j < 2; j++ )
 {
 printf("[%d] [%d] %d\n",i,j,a[i][j]);
 }
 }
 return 0;
}
```

1. What is the purpose of inner { } in array declaration and initialization?

Ans: I think what the use for the inner bracket the columns as from what i understand after searching google it is important when declaring 2 dimensional arrays it is need for the columns to have the inner brackets.

2. What happens when inner {} are omitted?

Ans: It would turn into an eror as it is a must for a 2-dimensional array is what it is stated in google but when I removed the inner brackets then it would work just fine so I don't know which is true or not.

Figure 6. Exercise 4

```c
#include <stdio.h>
#define SIZE1 5
double getAverage(double [],int); //average of values in the array
double getTotals(double *,int); //sum of all values in the array
int main ()
{
 double values[5] = {3,5,7,9,2};

 printf("Average value %.2f\n",getAverage(values,SIZE1));
 printf("Sum of values %.2f\n",getTotals(values,SIZE1));

 return 0;
}
double getAverage(double *arr,int size){
 int i;
 double avg;
 double sum = 0;
 for (i = 0; i < size; ++i) {
 sum += arr[i];
 }
 avg = sum / size;
 return avg;
}
double getTotals(double arr[],int size){
 int i;
 double sum = 0;
 for (i = 0; i < size; ++i) {
 sum += arr[i];
 }
 return sum;
}
```

1. Does the use of *arr or arr[]differ in their respective outputs? Why?
Ans: No, Because due to me testing both having switch the "*" and "[]" from each of the functions and at the function call. I believe it is just for the preference/style for the user that is used to.

IV. CONCLUSIONS

I learned a lot from this laboratory activity. I didn't know that you could create a pattern using a 2-dimensional array. I discovered a way to print these patterns using code that I found on the internet. This discovery was prompted by my initial failure to meet the requirements of the task, but it ultimately led me to learn more about for loops, especially when working with arrays. I realized that the key to printing array elements is using a for loop. When it comes to 2-dimensional arrays, you need to use nested for loops to display their contents.

References

GeeksforGeeks. (n.d.). "C Program Patterm." Retrieved September 12, 2023, from https://www.geeksforgeeks.org/c-program-print-pyramid-pattern/.

Stack Overflow. (2018). "C Histogram." Retrieved September 12, 2023, from https://stackoverflow.com/questions/49468519/generate-histogram-in-c.

Stack Overflow. (2016). "C chessboard." Retrieved September 12, 2023, from https://stackoverflow.com/questions/40252555/how-to-make-a-chessboard-in-c.

TechOnTheNet. (n.d.). "C constants." Retrieved September 12, 2023,from https://www.techonthenet.com/c_language/constants/create_define.php.

Tutorialspoint. (n.d.). "Array index Greater" Retrieved September 12, 2023, from https://www.tutorialspoint.com/what-happens-if-try-to-access-an-element-with-an-index-greater-than-the-size-of-the-array-in-java.

Stack Overflow. (2016). "Print Position of Array" Retrieved September 12, 2023, from https://stackoverflow.com/questions/40822959/how-to-print-the-position-of-a-value-in-an-array-in-c.

Stack Overflow. (2020). "How to Declare 2-dimensional Arrays" Retrieved September 12, 2023, from https://stackoverflow.com/questions/61464607/what-is-the-best-way-to-make-2-dimensional-array-in-c.

CodingBlocks. (2018). "Diff Between *arr and arr[]." Retrieved September 12, 2023, from https://discuss.codingblocks.com/t/whats-diff-between-int-arr-and-int-arr/56265.