

Programming Lab Report

Vivid Padungkiatsakul 6601023620026¹

¹Robotic Engineering and Automation System, Faculty of Engineering, King
Mongkut's University of Technology North Bangkok

[010243107] C Programming 2023

Lab Week 6

Array [II]: Two Dimensional

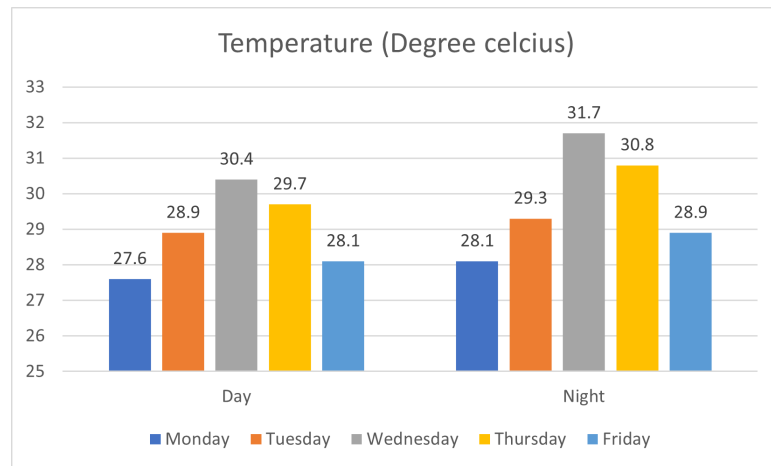
Topic

- Two dimensional (2D) Array
-

Experiment I

1 Problem description

- By using **2D array**, write a program that stores temperature value in array and finds average temperature of day and night.



2 Program design

- From the graph, we will get his value:

Day	27.6	28.9	30.4	29.7	28.1
Night	28.1	29.3	31.7	30.8	28.9

- We will collect all value in array call temp[]
- In this Experiment, we have to use 2D array. So the matrix of this array will be 2x5.
- For calculate the average of temperature in day and night, We will use function call Avgn().
- Avgn() function will output average from calulating the matrix in same row by getting matrix array and lenth of column in the row that we want.
- For calculate average or *Mean* $[\bar{x}]$ can be calculate by $\bar{x} = \frac{\sum_{n=1}^n x_n}{n}$

Part 3: Program text is on the next page.

3 Program text

```
#include <stdio.h>

float temp[2][5]={27.6,28.9,30.4,29.7,28.1},
               {28.1,29.3,31.7,30.8,28.9}};

float avge=0;
int i,j;
int row = sizeof(temp)/sizeof(temp[0]);
int column = sizeof(temp[0])/sizeof(temp[0][0]);

float Avge(float Sheet[][5], int row, int column){
    float sum=0;
    int count=0;
    for (int i = row-1; i < row; i++){
        for (int j = 0; j < column; j++){
            // printf("%f %f\n",sum,Sheet[i][j]);
            sum = sum + Sheet[i][j];
            count++;
        }
        // printf("%f %d\n", sum, count);
        avge = (float)sum/count;
        count = 0;
    }
    return avge;
}

int main(){
    printf("column = %d, row = %d\n", column, row);
    printf("Average [Day] = %.2f\n", Avge(temp, 1, column));
    printf("Average [Night] = %.2f", Avge(temp, 2, column));
}
```

4 Terminal output

```
column = 5, row = 2
Average [Day] = 28.94
Average [Night] = 29.76
```

Part 5: Problem that occur is on the next page.

5 Problem that occur

- The problem that we get is when we compare between hand calculate and programming calculate. The decimal are not same.
- This problem could be related to how the compiler or the environment is handling floating-point precision.
- You can see the comparison between hand calculate and programming calculate and full calculate of the program.

```
column = 5, row = 2
0.000000 27.600000
27.600000 28.900000
56.500000 30.400000
86.900002 29.700001
116.600006 28.100000
144.700012 5
Average [Day] = 28.94
0.000000 28.100000
28.100000 29.299999
57.400002 31.700001
89.100006 30.799999
119.900009 28.900000
148.800003 5
Average [Night] = 29.76
```

Programming output

First column [Day]

$$\bar{x} = \frac{27.6 + 28.9 + 30.4 + 29.7 + 28.1}{5}$$
$$\bar{x} = \frac{145.7}{5}$$
$$\bar{x} = 29.14$$

Second column [Night]

$$\bar{x} = \frac{28.1 + 29.3 + 31.7 + 30.8 + 28.9}{5}$$
$$\bar{x} = \frac{149.8}{5}$$
$$\bar{x} = 29.96$$

Hand Calculate

Experiment II

1 Problem description

- By using **2D array**, write a program that gets two of 3x3 matrix from user, adds them up and displays the result. Also, verify your program.

2 Program design

- In this experiment, we have to create matrix 3x3 from user input.
- So we can fix the parameter call column and row in 3.
- we can fix 2D array in 3x3 call matrix1[row][column] and matrix2[row][column]
- For getting user input to create in matrix, we use UserInputMatrix() function.
- The UserInputMatrix() function takes user input to populate a 2D matrix. The function uses nested loops to iterate over each element of the matrix, prompting the user to enter a value for each element. The matrix size is determined by the column and row parameters that are fix.
- For output matrix, we use MatrixOutput() function.
- The MatrixOutput() function prints a 2D matrix with square brackets, commas, and newlines to make it readable.
- For add 2 Matrices together, We use AddMatrix() function.
- The AddMatrix() function adds two matrices (matrix1[][] and matrix2[][]) element-wise and stores the result in another matrix.

Part 3: Program text is on the next page.

3 Program text

```
#include <stdio.h>

#define row 3
#define column 3

int matrix1[row][column] = {};
int matrix2[row][column] = {};
int result[row][column] = {};
int i, j;

void UserInputMatrix(int matrix[row][column]) {
    for (i = 0; i < row; i++) {
        for (j = 0; j < column; j++) {
            printf("Enter value: ");
            scanf("%d", &matrix[i][j]);
        }
    }
}

void MatrixOutput(int matrix[row][column]) {
    printf("[");
    for (i = 0; i < row; i++) {
        printf("[");
        for (j = 0; j < column; j++) {
            printf("%d", matrix[i][j]);
            if (j < column - 1) {
                printf(",");
            }
        }
        printf("]");
        if (i < row - 1) {
            printf("\n");
        }
    }
    printf("]");
}

void AddMatrix(int matrix1[row][column], int matrix2[row][column], int result[row][column]) {
    for (i = 0; i < row; i++) {
        for (j = 0; j < column; j++) {
            result[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }
}

int main() {
    printf("Enter values for matrix1:\n");
    UserInputMatrix(matrix1);
    printf("Enter values for matrix2:\n");
    UserInputMatrix(matrix2);

    printf("\nMatrix1:\n");
    MatrixOutput(matrix1);
    printf("\nMatrix2:\n");
    MatrixOutput(matrix2);

    AddMatrix(matrix1, matrix2, result);
}
```

```
printf("\nSum of the matrices:\n");  
MatrixOutput(result);  
  
return 0;  
}
```

4 Terminal output

```
Enter values for matrix1:  
Enter value: 1  
Enter value: 2  
Enter value: 3  
Enter value: 1  
Enter value: 2  
Enter value: 3  
Enter value: 1  
Enter value: 2  
Enter value: 3  
Enter values for matrix2:  
Enter value: 1  
Enter value: 2  
Enter value: 3  
Enter value: 1  
Enter value: 2  
Enter value: 3  
Enter value: 1  
Enter value: 2  
Enter value: 3  
  
Matrix1:  
[[1,2,3]  
[1,2,3]  
[1,2,3]]  
Matrix2:  
[[1,2,3]  
[1,2,3]  
[1,2,3]]  
Sum of the matrices:  
[[2,4,6]  
[2,4,6]  
[2,4,6]]
```

Experiment III

1 Problem description

- By using **2D array**, write a program that displays total number of student who get grade A, B, C, D, and F. Also, verify your program.

	A	B	C	D	F
Math	20	15	30	2	0
English	25	10	20	10	2
Physics	15	20	10	5	3

2 Program design

- In this experiment, we have to calculate total people in that grade.
- So, we have to create matrix 3x5 call ScoreSheet[3][5] to store the value.
- We create Grade[] to set grade value(length of array will same to ScoreSheet[][] column length) and TotalScoreGrade[] to get final value.
- For counting the total number of student who get grade, we use ScoreCount() function.
- The ScoreCount() function calculates total number of student who get grade based on the input RawSheet array (ScoreSheet[][]). The function iterates through the 2D array, accumulating the count of students for each grade in the output Sheet array (TotalScoreGrade[]).
- For output the total number of student who get grade, we use ScoreOutput() function.
- The ScoreOutput() function displays the total number of students for each grade. The function takes arrays of grade labels (Grade[]) and corresponding student counts (TotalScoreGrade[]) as parameters, iterating through them to print each grade and its associated student count.

Part 3: Program text is on the next page.

3 Program text

```
#include <stdio.h>

int ScoreSheet[3][5]={20,15,30,2,0},
                      {25,10,20,10,2},
                      {15,20,10,5,3}};
char *Grade[5] = {"A","B","C","D","F"};
int TotalScoreGrade[5]={};
int i,j;
int row = sizeof(ScoreSheet)/sizeof(ScoreSheet[0]);
int column = sizeof(ScoreSheet[0])/sizeof(ScoreSheet[0][0]);

void ScoreOutput(char *Grade[], int Sheet[], int lenght){
    for (i = 0; i < lenght; i++){
        printf("Grade %s: %d\n", Grade[i], Sheet[i]);
    }
}

int ScoreCount(int RawSheet[][5], int Sheet[], int column, int row){
    for (i = 0; i < row; i++){
        for (j = 0; j < column; j++){
            Sheet[j] = Sheet[j]+RawSheet[i][j];
        }
    }
}

int main(){
    ScoreCount(ScoreSheet,TotalScoreGrade,column,row);
    ScoreOutput(Grade,TotalScoreGrade,column);
}
```

4 Terminal output

```
Grade A: 60
Grade B: 45
Grade C: 60
Grade D: 17
Grade F: 5
```

Experiment IV

1 Problem description

- By using **2D array**, write a program that stores your first and last name in the array. Then, counts and displays number of character 'A' and 'a' in the array. For example,

J	o	h	n	
S	m	i	t	h

2 Program design

- In this experiment, we have to find how many upper letter and lower letter in name.
- In this time, we include string.h for easy to split text.
- So, we have to create matrix call Name and create array call User to collect user input after split.
- For collect firstname and lastname in matrix, we use StringToMatrix() function.
- The StringToMatrix() function converts the input string into a 2D matrix, where each row corresponds to a word and each column to a letter in the word. In this case is firstname and lastname.
- For counting upper and lower letter, we use CountLetterMatrix() function.
- The CountLetterMatrix() function counts the number of uppercase and lowercase letters in the matrix. The function utilizes nested loops to iterate through each element in the matrix.

Part 3: Program text is on the next page.

3 Program text

```
#include <stdio.h>
#include <string.h>

#define MAXLENGHT 50

char Name[2][MAXLENGHT];
char User[MAXLENGHT];
int i, j;
int count = 0, row = 0;
int countUpper = 0, countLower = 0;

void StringToMatrix(char str[], char matrix[][MAXLENGHT]){
    for (i = 0; User[i] != '\0'; i++) {
        if (str[i] != ' ') {
            matrix[row][count] = str[i];
            count++;
        } else if (str[i] == ' ') {
            row++;
            count = 0;
        }
    }
}

void CountLetterMatrix(char matrix[][MAXLENGHT]){
    for (i = 0; i < 2; i++) {
        for (j = 0; matrix[i][j] != '\0'; j++) {
            if (matrix[i][j] >= 'A' && matrix[i][j] <= 'Z') {
                countUpper++;
            } else if (matrix[i][j] >= 'a' && matrix[i][j] <= 'z') {
                countLower++;
            }
        }
    }
}

int main() {
    printf("Enter Name: ");
    fgets(User, sizeof(User), stdin);
    User[strcspn(User, "\n")] = '\0';

    StringToMatrix(User, Name);
    CountLetterMatrix(Name);

    printf("Number of 'A' characters: %d\n", countUpper);
    printf("Number of 'a' characters: %d\n", countLower);
}
```

4 Terminal output

```
Enter Name: John Smith
Number of 'A' characters: 2
Number of 'a' characters: 7
```