



Submission

Practical 6 to 7

Module Code

CS102.3

Module Name

Programming with C language

Student Name

Udawaththa UT

Student ID

30136

Faculty of Computing

NSBM Green University Twon

Practical 6

Q1)

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

int main()
{
    int array[10];
    int i;

    printf("Enter 10 integer values:\n");
    for (i = 0; i < 10; i++) {
        scanf("%d", &array[i]);
    }

    int min = array[0];
    for (i = 1; i < 10; i++) {
        if (array[i] < min) {
            min = array[i];
        }
    }

    int max = array[0];
    for (i = 1; i < 10; i++) {
        if (array[i] > max) {
            max = array[i];
        }
    }

    int sum = 0;
    for (i = 0; i < 10; i++) {
        sum += array[i];
    }
    float average_value = (float)sum / 10;

    int reversed_array[10];
    for (i = 0; i < 10; i++) {
        reversed_array[i] = array[9 - i];
    }

    printf("\nArray values: ");
    for (i = 0; i < 10; i++) {
        printf("%d ", array[i]);
    }
    printf("\n");
    printf("Minimum value: %d\n", min);
```

```

printf("Maximum value: %d\n", max);
printf("Average value: %.2f\n", average_value);

printf("Array in reverse order: ");
for (i = 0; i < 10; i++) {
    printf("%d ", reversed_array[i]);
}
printf("\n");
return 0;
}

```

The screenshot shows the Code::Blocks IDE with the project 'practical6.1' open. The source file 'main.c' is displayed, showing the implementation of an array processing program. The program calculates the minimum, maximum, and average values of an array, and prints the array in reverse order. The execution output is shown in a separate window, displaying the input values and the calculated statistics.

Source Code (main.c):

```

27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

```

```

int sum = 0;
for (i = 0; i < 10; i++) {
    sum += array[i];
}
float average_value = (float)sum / 10;

int reversed_array[10];
for (i = 0; i < 10; i++) {
    reversed_array[i] = array[9 - i];
}

printf("\nArray values: ");
for (i = 0; i < 10; i++) {
    printf("%d ", array[i]);
}
printf("\n");
printf("Minimum value: %d\n", min);
printf("Maximum value: %d\n", max);
printf("Average value: %.2f\n", average_value);

printf("Array in reverse order: ");
for (i = 0; i < 10; i++) {
    printf("%d ", reversed_array[i]);
}
printf("\n");
return 0;
}

```

Execution Output:

```

C:\Users\3PCT\Desktop\C codes\practical6.1\bin\Release\practical6.1.exe
Enter 10 integer values:
59
63
12
9
452
65
23
986
145
96

Array values: 59 63 12 9 452 65 23 986 145 96
Minimum value: 9
Maximum value: 986
Average value: 191.00
Array in reverse order: 96 145 986 23 65 452 9 12 63 59

Process returned 0 (0x0)   execution time : 13.412 s
Press any key to continue.

```

```
Q2) #include <stdio.h>
#include <stdlib.h>

int main()
{
    int size;
    printf("Enter the size of the arrays: ");
    scanf("%d", &size);

    int array1[size], array2[size], vector_sum[size];
    int scalar_sum = 0;
    int i;

    printf("Enter %d integer values for array1:\n", size);
    for (i = 0; i < size; i++) {
        scanf("%d", &array1[i]);
    }

    printf("Enter %d integer values for array2:\n", size);
    for (i = 0; i < size; i++) {
        scanf("%d", &array2[i]);
    }

    for (i = 0; i < size; i++) {
        scalar_sum += array1[i];
    }

    for (i = 0; i < size; i++) {
        vector_sum[i] = array1[i] + array2[i];
    }

    printf("\nScalar Sum: %d\n", scalar_sum);

    printf("Vector Sum:\n");
    for (i = 0; i < size; i++) {
        printf("%d ", vector_sum[i]);
    }
    printf("\n");

    return 0;
}
```

Practical 7

Q1)

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

```
#define ROWS 3
```

```
#define COLS 3
```

```
void inputMatrix(int matrix[ROWS][COLS]) {  
    for (int i = 0; i < ROWS; i++) {  
        for (int j = 0; j < COLS; j++) {  
            scanf("%d", &matrix[i][j]);  
        }  
    }  
}
```

```
void displayMatrix(int matrix[ROWS][COLS]) {  
    for (int i = 0; i < ROWS; i++) {  
        for (int j = 0; j < COLS; j++) {  
            printf("%d ", matrix[i][j]);  
        }  
        printf("\n");  
    }  
}
```

```
void addMatrices(int matri1[ROWS][COLS], int matri2[ROWS][COLS], int result[ROWS][COLS]) {  
    for (int i = 0; i < ROWS; i++) {  
        for (int j = 0; j < COLS; j++) {  
            result[i][j] = matri1[i][j] + matri2[i][j];  
        }  
    }  
}
```

```
int main() {  
    int matri1[ROWS][COLS];  
    int matri2[ROWS][COLS];  
    int sumMatrix[ROWS][COLS];  
  
    printf("Enter elements for the first 3x3 matrix:\n");  
    inputMatrix(matri1);  
  
    printf("Enter elements for the second 3x3 matrix:\n");  
    inputMatrix(matri2);  
  
    addMatrices(matri1, matri2, sumMatrix);  
  
    printf("\nFirst Matrix:\n");
```

```
displayMatrix(matri1);
```

```
printf("\nSecond Matrix:\n");  
displayMatrix(matri2);
```

```
printf("\nSum of Matrices:\n");  
displayMatrix(sumMatrix);  
return 0;
```

```
}
```

The screenshot shows a C code editor with a project named 'practical7'. The code defines two 3x3 matrices, 'matri1' and 'matri2', and calculates their sum. The output console shows the program's execution, including prompts for matrix elements and the resulting sum.

```
main.c [practical7] - Code::Blocks 20.03  
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help  
<global>  
Management  
Workspace  
practical6.1  
Sources  
main.c  
PRACTICAL6.2  
Sources  
main.c  
practical7  
Sources  
main.c  
main.c  
1 #include <stdio.h>  
2 #include <stdlib.h>  
3 #define ROWS 3  
4 #define COLS 3  
5  
6 void inputMatrix(int matrix[ROWS][COLS]) {  
7     for (int i = 0; i < ROWS; i++) {  
8         for (int j = 0; j < COLS; j++) {  
9             scanf("%d", &matrix[i][j]);  
10        }  
11    }  
12 }  
13  
14 void displayMatrix(int matrix[ROWS][COLS]) {  
15     for (int i = 0; i < ROWS; i++) {  
16         for (int j = 0; j < COLS; j++) {  
17             printf("%d ", matrix[i][j]);  
18         }  
19         printf("\n");  
20     }  
21 }  
22  
23 void addMatrices(int matri1[ROWS][COLS], int matri2[ROWS][COLS], int result[ROWS][COLS]) {  
24     for (int i = 0; i < ROWS; i++) {  
25         for (int j = 0; j < COLS; j++) {  
26             result[i][j] = matri1[i][j] + matri2[i][j];  
27         }  
28     }  
29 }  
30  
31 int main() {  
32     int matri1[ROWS][COLS];  
33     int matri2[ROWS][COLS];  
34     int sumMatrix[ROWS][COLS];  
35     inputMatrix(matri1);  
36     inputMatrix(matri2);  
37     displayMatrix(matri1);  
38     displayMatrix(matri2);  
39     addMatrices(matri1, matri2, sumMatrix);  
40     displayMatrix(sumMatrix);  
41     return 0;  
42 }
```

Output Console: "C:\Users\3PCT\Desktop\practical7\bin\Release\practical7.exe"

```
Enter elements for the first 3x3 matrix:  
12 53 98  
12 69 85  
98 36 46  
Enter elements for the second 3x3 matrix:  
12 65 96  
45 23 52  
15 69 45  
First Matrix:  
12 53 98  
12 69 85  
98 36 46  
Second Matrix:  
12 65 96  
45 23 52  
15 69 45  
Sum of Matrices:  
24 118 194  
57 92 137  
113 105 91  
Process returned 0 (0x0) execution time : 34.445 s  
Press any key to continue.
```