LAB 3

Code : Problem A

Lab03_ProbA.c > copy_by_pointer(int *, int *)

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

#define N 5

void copy_by_array(int arr[N], int arr2[N])
{

    for (int i = 0; i < N; i++)
    {
        arr2[i] = arr[i]; // copying arr to arr2
    }
}

void copy_by_pointer(int *x, int *y)
{
    for(int i = 0; i < N; i++)
    {
        *(y + i) =  *(x + i); // copy by using pointer notation
    }

}

//additional function just to print the elements of an array
void print_array(int arr[N])
{
    for (int i= 0; i < N; i++)
    {
        printf("%d\n", arr[i]);
    }

}

int main()
{
    int arr[N], arr2[N], arr3[N]; //declaring the three arrays we are going to use
    printf("The elements of the array before any changes are:\n");
    srand(time(0));
    for (int i =0; i< N; i++)
    {
        arr[i] = rand() % 100; // Random numbers between 0 and 99
        printf("%d\n", arr[i]); //printing the elements as soon as they are assigned to the array
    }

    // demonstrating copy by array
    copy_by_array(arr, arr2); //copying to arr2
    printf("The elements of the second array are : \n");
    print_array(arr2);

    // demonstrating copy by using pointer notation
    copy_by_pointer(arr, arr3);
    printf("The elements the third array are:\n");
    print_array(arr3);
    return 0;
}
```

output : Problem A

```
→  Lab03 ./a.out
The elements of the array before any changes are:
7
39
29
75
23
The elements of the second array are :
7
39
29
75
23
The elements the third array are:
7
39
29
75
23
```

Output : Problem B

```
→  Lab03 gcc -Wall Lab03_ProbB.c
→  Lab03 ./a.out
Here is the array BEFORE the transpose operation.
81  96  70  23
76  11  97  23
61  58  27  83
Here is the array AFTER the transpose operation.
81  76  61
96  11  58
70  97  27
23  23  83
```

Problem B : Code

Lab03_ProbB.c > main()

```c
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#define N 3
#define M 4

void transpose(int arr[N][M], int arr2[M][N])
{
    for(int i = 0; i < N; i++)
    {
        for(int j = 0; j < M; j++)
        {
            arr2[j][i] = arr[i][j]; //transpose
        }
    }

}


int main()
{
    int arr[N][M], arr2[M][N];
    srand(time(0)); // seed the random generator
    printf("Here is the array BEFORE the transpose operation.\n"); // showing the array BEFORE transposing it
    for (int i = 0; i < N; i++)
    {
        for(int j = 0; j< M ; j++)
        {
            arr[i][j] = rand() % 100; //randomly assign values between 0 and 99
            printf("%d  ", arr[i][j]);
        }
    printf("\n");
    }
    //Transpose
    transpose(arr, arr2);
    printf("Here is the array AFTER the transpose operation.\n"); // showing the array AFTER transposing it
    for (int i = 0; i < M; i++)
    {
        for (int j = 0; j < N; j++)
            printf("%d  ", arr2[i][j]);
        printf("\n");
    }

    return 0;
}
```

Problem D => Code

Lab03_ProbD.c > main()

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

int main()
{
    int array[50][100] = {}; //all elements are 0
    srand(time(0));
    int i = rand() % 50; // rows
    int j = rand() % 100;// columns
    int *position = &array[i][j]; // get a random position
    int x= ((*position + i) + j) / 100; //using pointer notation to get the row
    int y = ((*position + i) + j) % 100; // using pointer notation to get the column
    printf(" The position found is: (%d, %d)\n", x, y);
    return 0;
}
```

Output: Problem D

```
→  Lab03 gcc Lab03_ProbD.c
→  Lab03 ./a.out
 The position found is: (0, 92)
```

Problem E

The elements stored in A to G include:

$A[5] = \{1, 5, 9, 0, 0\};$

$B[3] = \{0, 0, 0\}$

$C[2] = \{0, 0\};$

$D[5] = [C[1], 2\};$ $\Rightarrow \{0, 2, 0, 0, 0\}$

$E = D[C[0] + A[4]];$ $\Rightarrow D[0] \Rightarrow \{0\}$

$F[3][2] = \{\{1, 2\}, \{3, 4\}, \{5, 6\}$

$G[6] = \{1, 2, 3, 4, 5, 6\}$

Problem F : Code

Lab03_ProbF.c > getter(int [], int, int)

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


#define M 20
#define N 5
int getter(int ar[], int k, int j);
void setter(int arr[]);

int getter(int ar[], int k, int j)
{
    printf("Here is the array BEFORE modifying it to a 20X5 array:\n");
    setter(ar);
    printf("\n And now, here is the array AFTER : \n");
    for(int i = 0; i < M; i++)
    {
        for (int j = 0; j< N; j++)
        {
            printf("%d ", ar[(i * (N-1)) + j]);
        }
        printf("\n");
    }
    return ar[(k * (N-1)) + j];//calculating and returning the(k, j) element
}

void setter(int arr[N])
{
    srand(time(0));
    for (int i= 0; i < (M*N); i ++)
    {
        arr[i] = rand() % 100; // generate numbers between 0 and 99
        printf("%d \n", arr[i]); // print the random numbers
    }
}

int main()
{

    int arr[M * N], x, y;
    printf("Enter the position you want access: \n");
    scanf("%d ",&x); //first digit
    scanf(" %d", &y); // second digit

    int z = getter(arr, x, y); //to store the element (k, j)
    printf("The element at [%d, %d] is: %d\n", x, y, z );
    return 0;
}
```

Problem F : output

Note: the array always starts at zero (0);

```
[→  Lab03 gcc Lab03_ProbF.c
[→  Lab03 ./a.out
Enter the position you want access:
4
4
Here is the array BEFORE modifying it to a 20X5 array:
84
92
78
40
67
14
33
18
73
57
12
82
70
66
86
0
74
48
2
28
78
89
36
49
57
75
76
67
94
94
22
81
35
25
70
7
17
96
10
42
81
15
98
83
24
69
79
67
25
15
50
30
```

```
96
48
32
66
20
37
55
45
70
89
87
13
86
36
6
78
65
55
4
19
47
63
31
6
89
18
88
48
44
47
92
28
90
86
58
69
73
59
81
38
73
17
66
71

 And now, here is the array AFTER :
84 92 78 40 67
67 14 33 18 73
73 57 12 82 70
70 66 86 0 74
74 48 2 28 78
78 89 36 49 57
57 75 76 67 94
94 94 22 81 35
35 25 70 7 17
17 96 10 42 81
81 15 98 83 24
24 69 79 67 25
25 15 50 30 37
37 2 11 41 96
96 48 32 66 20
20 37 55 45 70
70 89 87 13 86
86 36 6 78 65
65 55 4 19 47
47 63 31 6 89
The element at [4, 4] is: 78
```