

## LAB # 12

Monday, April 24 (L1) and Wednesday, April 26 (L2)

### OBJECTIVES

- To practice arrays, strings, and pointers

### SUBMISSION REQUIREMENT

1. Please follow “[Submission Guidelines](#)” in the lab section of the course website to submit your solution (program files) to the class M: drive by [5pm, Wednesday, April 26](#)
2. Suggested format for naming your solution files: [lab#-your last name-p#.cpp](#)  
For example: [lab12-xing-p1.cpp](#) for problem 1; [lab12-xing-p2.cpp](#) for problem 2; ...

### EXERCISES

1. Correct errors in the following program and then run the program to understand what task the program can perform. (**Hint: 15 errors**)

```
/* This is a debugging problem in Exam #3 */
#include "stdlib.h"
#include "stdio.h"
#include "time.h"
#define array_size 7;

INT myfunction(void)

void main(void);
{
    int myarray[array_size] = (1, 2, 3, 4, 5, 6, 7, 8, 9);
    int yourarray[array_size] = (0);
    int k = 0;

    srand(time(NULL));

    for (k = 0; k <= array_size; k++)
    {
        myarray[k] = myfunction(void);
        yourarray = myarray;
    }

    for (k = 0; k < array_size; k++)
        fprintf("The element %d in your array is %d \n", k, &yourarray[k]);

    return 0;
}

int myfunction(void);
{
    int rand1;
    rand1 = rand() % 101 + 30;
    return rand1
}
```

2. Write down the output of the following program. Then run the program to verify the result. If you have difficulty with understanding some of them, please ask the lab assistants for the help with an explanation.

```
#include "stdio.h"

int main(void)
{
    char s[20] = "Life is beautiful";
    char* ps = s;
    int k;

    for (k = 0; k < 4; k++)
        printf("%c", s[k]);

    printf("\n");

    printf("%c", ps[16]);
    printf("%c", ps[3]);
    printf("%c", ps[10]);
    printf("%c", ps[14]);

    printf("\n");
    printf("%d", s[17]);
    printf("%d", s[18]);
    printf("%d", s[19]);

    return 0;
}
```

3. Write down the output of the following program. Then run the program to verify the result. If you have difficulty with understanding some of them, please ask the lab assistants for the help with an explanation.

```
#include "stdio.h"

void add1(int m, int n);
void add2(int* d, int e);

void main(void)
{
    int k;
    int num[6] = { 1,2,3,4,5,6 };
    int s = 10;

    for (k = 0; k < 6; k++)
    {
        if (num[k] % 2 == 0)
            add1(num[k], s);
        else
            add2(&num[k], s);
    }
}
```

```

        for (k = 0; k < 6; k++)
            printf("%d\n", num[k]);
    }

void add1(int m, int n)
{
    m = m + n;
}

void add2(int* d, int e)
{
    *d = *d + e;
}

```

4. Write a complete C program that can perform the following tasks:
- 1) Read 12 integers into a 1-D array from the keyboard using a **for** loop.
  - 2) After all the integers have been read, do the following update for the array elements:
    - adds 10 if the array element is an even number,
    - adds 60 if the array element is an odd number.
  - 3) Finally, print out the first and the last array elements (**one per line**) after the update.

**For example:** If you input 1 2 3 4 5 6 7 8 9 10 11 12 from the keyboard, they will be updated to **61 12 63 14 65 16 67 18 69 20 71 22**. The following numbers should be displayed on the screen when the program runs.

61  
22

5. The following program is supposed to display the string “Great job!” (one letter per line) on the screen, as follows:

```

G
r
e
a
t

j
o
b
!

```

Run the program to see the errors. Then correct the errors so that the program can generate the correct output.

```

#include "stdio.h"

void main(void)
{
    char a[11] = "Great job!";
    int i;

    for (i = 1; i <= 11; i++)
    {
        printf("%d\n", a[i]);
    }
}

```

6. Write down the output of the following program. Then run the program to verify the result. If you have difficulty with understanding some of them, please ask the lab assistants for the help with an explanation.

```

#include "stdio.h"

void main(void)
{
    int x = 13;
    int* p;
    p = &x;
    int** q;
    q = &p;

    printf("%d\n", x);
    printf("%d\n", *p);
    printf("%d\n", **q);

    printf("%d\n", x++);
    printf("%d\n", ++x);

    printf("%d\n", p);
    printf("%p\n", p);
    printf("%d\n", q);
    printf("%p\n", q);
}

```