lame:	Student ID:	Date:
-------	-------------	-------

Worksheet 8

while and do...while statements

Objectives

After completing this worksheet, you should be able to

- Understand the concept of repetition control structures
- Comprehend the usage of *while* and *do...while* statements
- Appreciate the usage of *break* statement
- 1. Open a new editor and write Program 8.1.

Program 8.1

2. Run program 8.1 and record the results in Table 8.1.

Result from step #2	Result from step #3

- 3. From Program 8.1, replace the statement $\mathbf{x} = \mathbf{x} + \mathbf{1}$; in ① with $\mathbf{x} = \mathbf{x} + \mathbf{2}$;. Then run the modified program and record the results in Table 8.1.
- 4. Suppose that we replace the statement while(x <= 5) with while(1) or any non-zero value in the parentheses of while() statement. What does the statement do?

5. Open a new editor and write Program 8.2.

Program 8.2

6. Run Program 8.2. Then enter the input values given in Table 8.2 and record the result.

Input values	Result from step #6
3	Total of i =
7	Total of i =
0	Total of i =
10	Total of i =
20	Total of i =

Table 8.2

From Table 7.2, when we entered **0** (zero) as an input to the program. Was the set of statements between the curly brackets executed?

How about the negative value?

7. Open a new editor and write program 8.3.

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    int a = 0;
    while( a < 10) {
        a++;
        if(a == 6) { // If the condition is true, the statement break; will be executed.
            printf("I like this number.!!!\n");
            break;
        }
        printf("The value of a is %d\n",a);
    }
    printf("Good bye \n");
    return 0;
}</pre>
```

Program 8.3

8. Run Program 8.3 and record the results in Table 8.3.

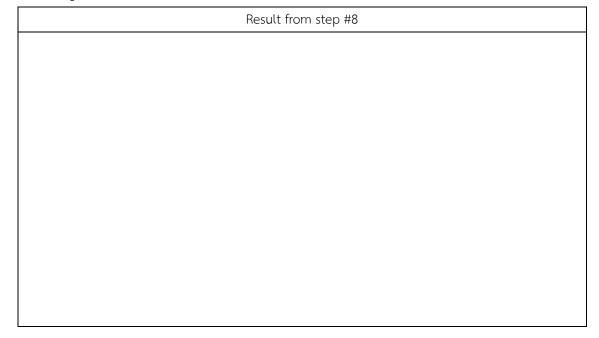


Table 8.3

```
9. Open a new editor and write Program 8.4
    #include <stdio.h>
    int main(int argc, char *argv[]) {
      int x = 1;
                           // Initialize the value of x which is the repetition control variable
      do {
          printf("Born to Code in C. \n");
         x = x + 1; // Increase the value of x by 1 (The reduced form is x++)
      } while (x \le 5); // If the condition is true, the statements after do will be done.
      return 0:
   }
                                           Program 8.4
    Run Program 8.4 and compare with the results produced by Program 8.1. Are the results
    different?_____
    Suppose that we replace the statement while (x \le 5) with while (1) or any non-zero value in
    the parentheses of while() statement. What does the statement do?
10. Open a new editor and write Program 8.5. Then, run and observe the results.
#include <stdio.h>
int main(int argc, char *argv[]) {
   char key;
   do {
       printf("Press any key -> ");
       key = getch();
       printf("\nAscii code : %d\n",key);
   } while(key != 27);
                                           // Press Esc key to terminate the program
   return 0;
}
                                       Program 8.5
   What does the program do?
```

Student ID:	Date:	
	Student ID:	

1. From the question 1 in Homework 7, improve the ability of your program by using the repetition statements. Suppose that if users press 4, the program will terminate otherwise (users choose 1 to 3) the program will redisplay the menu and wait for input from users again.