Programming Languages Homework 1
(September 2016)

Please write or print your solutions on A4 papers. Also, remember to write down your department, student ID, and name. The maximum score for this exercise is 90 points.

1. (10 points) Provide a brief discussion on how the internet affected the development of programming languages.

2. (10 points) Consider the C program below. Categorize the three assignments as legal or illegal, providing a justification for your answer.

```c
int o, *r, z[3];
&o = (int *)malloc(sizeof(int));
z = (int *)malloc(sizeof(int)*3);
*r = 3;
```

3. (10 points) Which compiler phase (e.g. scanner, parser, semantic analyzer, code generator) would be responsible for catching each of the following errors, respectively? Explain why.
   1. An illegal character.
   2. Premature end-of-line.
   3. A undeclared variable.
   5. Division by zero.

4. (20 points) Similarity between two programs can be measured via it parse tree. Let us trace the GCD program in the textbook by doing Exercise 1.5 in the textbook.

5. (10 points) Do exercise 1.7 in the textbook.

6. A Turing machine is an abstract “machine” that manipulates symbols on a strip of tape according to a table of rules; to be more exact, it is a mathematical model that defines such a device. Despite its simplicity, a Turing machine can simulate the logic of any computer algorithm. Programs written in this model is just as tedious as writing assembly or machine codes. BrainF**k (BF) is a language that simulates the behaviour of turing machine with only 7 instructions: input, output, increment, decrement, wind, rewind, and loop. The BF language specification can be found at https://en.wikipedia.org/wiki/Brainfuck and a simulator can be found at http://www.nayuki.io/page/brainfuck-interpreter-javascript.
(a) (5 points) Given an example of the BF language that prints *Hello world!*, modify (or rewrite) the following code to print *Yours truly!*

```
1 ++++++++[>+++++>++++>++++>++++>+<<<<]>+<<<<>>[<++++++++++.>+.<<<<.>]<+-----%
2 >>.>----+.++++++++++.>++>.<-. <<.++++-+-.----------.>>>++.>++.>++
```

(b) (5 points) Write a BF program which reads the input and outputs the content read backwards, i.e., reading *abcde* will output *edcba*.

(c) (5 points) Write a BF program that lets the user input a character and copy the character value to the next cell and maintain the original value. In C, the program is

```
1 char a[2];
2 scanf("%c", a[0]);
3 a[1] = a[0];
```

(d) (15 points) (*) Write a BF program that can add all digits of the input and print the resulting integer, i.e., input is *123456789*, the output is *45*. You can assume the test case will not overflow a 8-bit cell. Remember the information stored in cells are in ASCII code, you may need to write a function similar to the C library `printf("%d")` to convert ASCII to integers.