[redacted]

21.5

Compiler Construction Quiz-1

3

1. Which compiler phase will produce errors in the C code snippets given below, and what will be those errors? Please specify the exact line that will produce the said error. [8 Marks]

a)

```
1  int main() {
2      int x = 5;
3      int x = 6;
4      return 0;
5  }
6
```

Line 3 which will be given by ~~semantic analyzer~~ and that error will be that x has been redeclared.

b)

```
1  int main() {
2      for (int i=0; i<=5; i++) {
3          printf("Hello");
4      }
5      printf(i);
6      return 0;
7  }
8
```

c)

```
1  int main() {
2      int i = 57$;
3      if (i > 3) printf("i =%d", i);
4      return 0;
5  }
6
```

d)

```
1  int main() {
2      int j = 30;
3      if (j > 3) {
4          printf("j =%d", j);
5          printf("3");
6      return 0;
7  }
8
```

a) The error will be in line 3 and will be given by ~~syntax analyzer.~~ The error will be that the x variable has been redeclared. (1)

b) The error will be in line 5 and will be given by semantic analyzer. The error will be that i has not been declared as it is out of scope of the for loop thus i is undeclared outside the loop. (2)

c) The error is in line 2 and is given by ~~syntax~~ ~~semantic~~ analyzer. The error is that the i variable is not int type. (0)

d) (0)

2. Write all the 1+7=8 phases of the compiler while showing the input and output of each phase. [16 Marks]

→ Stream of characters ✓

Lexical analyzer ✓

↓ ← stream of tokens ✓

Syntax analyzer ✓

↓ ← Syntax tree ✓

(19)

Semantic analyzer

↓ ← ~~annotated~~ syntax tree

Intermediate Code

Generator ✓

↓ ← ~~3AC~~ (IR) ✓

Machine Independent

Code optimizer ✓

↓ 3AC (IR) optimized ✓

Target Code ✓

Generator

↓ ← target code ↑ assembly

Machine Dependent ✓

Code optimizer

↓ ← optimized target code → assembly

one phase is missing

3. Write in fewer than 100 words what the role of linker/loader is in a compiler. [5 Marks]

It converts the sorce code into object files and then links them to form a single executable to run on the operating system's memory.

3

4. True/false and short answer: 3.5

- Compiler translates code once. Subsequently, the translated code can be executed with translation again and again. (1) (.5)
- Typically, ~~Compilers~~ Compiler translates code faster than Interpreter and ~~Assembly~~ Machine Independent Code Ophimizer phase is language and machine-independent. (1) (1)
- Two different languages using the same backend must generate the same IR code. (0)
- Interpreted code is usually considered more portable than Compiled code. (1)