Assignment-8

Due : 29th March, 2023

In this assignment, your task is to generate intermediate code. You may assume that $if - else$ and *while* statements are not present in the input code. The intermediate code is to be written in LLVM IR language. A sample input and output codes for the intermediate code generator is given below.

Input (input) :

```
int main()
{
    int a, b;
    a = 6;
    b = a / 7;
}
```

Output (out.ll) :

```
define i32 @main() #0
{
    %1 = alloca i32, align 4
    %2 = alloca i32, align 4
    store i32 6, i32* %1, align 4
    %3 = load i32, i32* %1, align 4
    %4 = sdiv nsw i32 %3, 7
    store i32 %4, i32* %2, align 4
    ret i32 0
}
```

Note that the sample output does not have header and footer for a typical llvm IR (.ll) file. It is expected that the generated file out.ll can be subjected to the llvm interpreter *lli* without any error. More precisely, the command
lli out.ll
will not output any error. You may allow type integer and operators '-', '/', copy only for generating intermediate code. Check with the following input and output code also.

```
int main()
{
    int a, b, c, d;
    a = 5;
    b = a / 2;
    c = a - b;
    d = c;
}
```

Output (out.ll) :

```
 define i32 @main() #0 {
  %1 = alloca i32, align 4
  %2 = alloca i32, align 4
  %3 = alloca i32, align 4
  %4 = alloca i32, align 4
  store i32 5, i32* %1, align 4
  %5 = load i32, i32* %1, align 4
  %6 = sdiv i32 %5, 2
  store i32 %6, i32* %2, align 4
  %7 = load i32, i32* %1, align 4
  %8 = load i32, i32* %2, align 4
  %9 = sub nsw i32 %7, %8
  store i32 %9, i32* %3, align 4
  %10 = load i32, i32* %3, align 4
  store i32 %10, i32* %4, align 4
  ret i32 0
}
```