Assignment #2
CS3300

1. [5] **Activation records**: Draw the runtime activation records at the entry and exit to each function. Pay special attention to the following fields of the activation records: actual parameters, return value, control link, and local variables.

```
foo(){ // Returns the number of primes in x.
  int x[] = {2, 3, 4, 5, 6}, len = 5;
  return Eratosthenes(x, len, 0);
}
int Eratosthenes(int x[], int len, int index){
  int i, v = x[index];
  for (i=index+1; i<len; i+= v) x[i] = -1;
  for (i=index+1; i< len; i++)
    if (x[i] != -1) return Eratosthenes (x, len, i);
  return countNonNegative (x, len);
}
int countNonNegative(int x[], int len){
  int i, count=0;
  for (i=0;i<len;++i) if (x[i] > 0) count ++;
  return count;
}
```

2. [5] **Code generation**: Write an SDT to generate IR in three-address code (similar to the one discussed in the class) for the following grammar. Briefly explain about the attributes you use.

```
P → S
S → SwitchStmt S | Assignment; S  | ε
SwitchStmt → switch ( Id ) { CaseBlocks }
CaseBlocks → case Lit : Stmt2; CaseBlocks | default: Stmt2
Stmt2 → Assignment | Break
Assignment → x = E
E : RelEx | AddEx | Id
RelEx → E < E
AddEx → E + E
Break → break
```

3. [5] **Flow-graph**: Generate the three-address code and draw the flow graph.

```
void sort (int A[], int n){
  for (int i=0;i<n-1;++i)
    for (int j=i+1;j<n; ++j)
      if (A[i] > A[j]){ tmp = A[i]; A[i] = A[j]; A[j] = tmp; }
}
```

4. [5] **Register Allocation**: Prove that the live variable analysis discussed in the class computes the liveness information for each variable, conservatively. At runtime, if the value of a variable v is live at a program point L (that is used at a later point of L), then the computed $In(L)$ set includes v.

5. [5] **Personal!** Based on the portion covered after Quiz 1 in CS3300, make an interesting question and attempt an answer for the same. Credit will be given based on the creativity, ingenuity, and coolness of the question and the answer.