



1. a)

Task / Weeks	1	2	3	4	5	6	7	8	9	10	11	12
Defining and Understanding the Problem	Shaded	Shaded										
Planning and Design of Solution			Shaded	Shaded	Shaded	Shaded						
Implementation of Solution							Shaded	Shaded				
Testing of Solution								Shaded	Shaded			
Maintenance of Solution											Shaded	Shaded

### Gantt Chart

1. b)

The data dictionaries will aid the programmers in that they will describe the attributes and purposes of the data variables used in the programs algorithms source code. This will decrease production time by helping the programmers understand the existing system.

The test data can be used when testing the new system (which will be composed of make use of the old systems algorithms). This will also decrease production time as a significant portion of testing time is spent creating test data that will suit an algorithm.



1.c.i)

Code Line	total	gst - total	Transaction amount	Transaction. gst
1				
2				
3	0	0		
4	0	0		
5	0	0	22	2
6	0	0	22	2
7	22	0	22	2
8	22	2	22	2
9	22	2	33	3
10	22	2	33	3
11	55	2	33	3
12	55	5	33	3
13	55	5	11	1
14	55	5	11	1
15	66	5	11	1
16	66	6	11	1
17	66	6	222	0
18	66	6	222	0
19	66	6	222	0

Desk check of Algorithm



21.c.i) The program outputs 66 and 6, which are the expected results. It is assumed that 222, 0 is the sentinel record.

21.c.ii)

1. BEGIN
2. OPEN Transaction
3. total = 0
4. gst\_total = 0
5. ~~total-avg~~  
total\_avg = 0
6. count = 0
7. READ first record
8. WHILE record NOT sentinel
9. total = total + Transaction.amount
10. gst\_total = gst\_total + Transaction.gst
11. count = count + 1
12. READ next record
13. ENDWHILE
14. total\_avg = total / count
15. OUTPUT total, gst\_total, total\_avg.
16. END

1.d.i) The CASE tool provides a record of the different versions and revisions the program has gone through. It also provides the number of lines, giving an idea of the amount of change that the program has gone through between versions. (The lines could also be used to identify a particular source)

1.d.ii) It is assumed that the data is stored in the file as a record <sup>delimiters</sup> (delimiters separating version and line number pairs, which are in turn separated).

```
BEGIN Record_Longest  
  OPEN 'Results'  
  OPEN 'VERSION'  
  READ  
  Read first record and store in RVersion  
  IF RVersion.NoLines > 11000 THEN  
    WRITE RVersion.VersionNo to 'Results'
```

```
BEGIN Record - Longest  
  OPEN 'Results'
```



2.1.d.ii)

OPEN 'VERSION'

READ first record<sub>in 'VERSION'</sub> and store it in RVersion

WHILE RVersion NOT last record in file

IF RVersion.NoLines > 11000 THEN

WRITE RVersion.Version No to 'Results'

END IF

READ next record in 'VERSION' and store in RVersion

ENDWHILE

CLOSE 'VERSION'

CLOSE 'Results'

END