

## **Practice Theory Questions**

### **Question 1**

A hash table has an index range of 1 to 900. The following pseudocode describes an algorithm for searching the table using the hashing function Hash. It is assumed that the key is present in the table.

```
1. Index <- Hash(Key)
2. WHILE Table[Index, 1] <> Key
3. Index <- Index + 1
4. ENDWHILE
5. Value <- Table[Index, 2]
```

**(a)** Explain the purpose of:

- (i)** line 3
- (ii)** line 5

[4]

**(b)** Describe a problem that might occur with a key which, when hashed, produces an index of 900.

[2]

**(c)** What modification to the algorithm is required to overcome this problem?

[3]

**(d)** Explain how a new item can be added to this hash table.

[4]

### **Question 2**

Bank customers are allowed to withdraw money from their accounts at an ATM. They cannot withdraw more than the current balance in their account. There is a daily limit on the amount that can be withdrawn. In some circumstances a charge is made for the transaction. The rules are:

- the transaction is rejected if the withdrawal amount requested is greater than the current balance
- the transaction is rejected if the withdrawal amount exceeds the daily limit
- if the current balance before the transaction is carried out is less than 50 dollars then any successful transaction incurs a fixed charge

**(a)** Create a decision table showing all the possible conditions and actions.

[4]

**(b)** Simplify your decision table by removing redundancies.

[4]

**(c)** Using your answer in (b) write a function using pseudocode. The function returns:

- -1 to indicate a rejection;
- 0 for a charge-free successful transaction;
- the charge for a chargeable successful transaction.

[5]

**(d)** State two ways in which your answer in (c) demonstrates clarity of code.

[2]

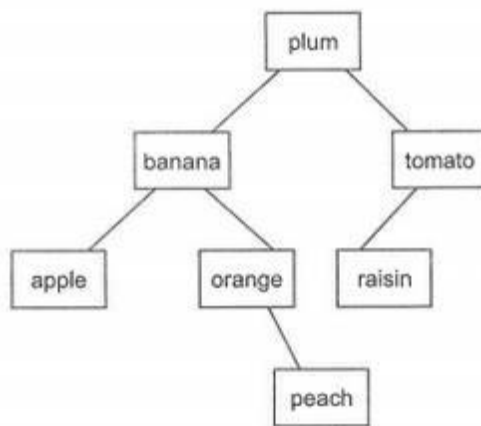
### Question 3

The ASCII code for the character 'Z', expressed as a denary integer, is 90.

- (a) Express the denary integer 90 as:
  - (i) an eight-bit binary number
  - (ii) a hexadecimal number [2]
- (b) Give two reasons why hexadecimal numbers are used in computing. [2]
- (c) State the ASCII code for 'X' in denary. Explain your answer. [2]
- (d) Explain why the Unicode encoding system has replaced ASCII. [2]
- (e) Describe a method of storing strings of characters of variable length in a computer. [2]

### Question 4

Consider the following binary tree:



- (a) List the nodes, in order, that are visited for a post-order traversal. [2]
- (b) List the nodes, in order, that are visited for an inorder traversal. [2]
- (c) What property is exhibited by the list of items produced in **part (b)**? [1]
- (d) Describe an algorithm, using pseudocode, to perform a binary tree search. The output should state whether or not the item is present in the tree. [5]

## Question 5

A function is to be written that returns the sum of all values held in an array that are greater than a minimum value. The function will be used with arrays of varying size, but never more than a maximum of 50 000 elements.

A first attempt at writing the program code for the function is given below:

```
1. FUNCTION TotalSum(Results : ARRAY[50000] OF REAL, ArraySize : INTEGER, MinValue : REAL)
    RETURNS REAL
2.   DECLARE Sum, Counter : INTEGER
3.   DECLARE Temp : Real
4.   Sum = 0.0
5.   FOR Counter = 1 TOO ArraySize
6.     Temp = Results[Counter]
7.     IF Temp > MinValue THEN Sum = Sum * Temp
8.   ENDEOR
9.   RETURN Sum
10. ENDFUNCTION
```

The function is included in a program specifically written to test the function. The main program outputs the value returned by the function. A compiler was used to compile the source program.

**(a)** The compiler reported an error at line 5 in the function. Identify the error and explain why it was flagged as a syntax error. [2]

**(b)** The compiler also reported an error at line 8. State the type of error reported by the compiler justifying your answer. [2]

The errors indicated in **parts (a) and (b)** were corrected. A successful compilation produces executable code. When the code was executed, the program failed to complete and reports an error at line 7.

**(c) (i)** State the type of error that occurred. Justify your answer. [2]

**(ii)** The error described in **part (c) (i)** depends on the detection of another type of error. Name this other type of error. How should the code be changed to correct this error? [2]

When the program finally runs without error, the test plan needs to be completed. The test plan uses data that tests different sizes of array, different array values and different minimum values.

The array `TempArray` is used in the main program as the array to be processed.

**(d)** Each element of `TempArray` stores a random value between 1.0 and 10.0.

**(i)** Explain why the function call:

`TotalSum(TempArray, 1000, 5.0)`

is not an appropriate black box test. [2]

**(ii)** Explain why the function call:

`TotalSum(TempArray, 10, 10.5)`

is not an appropriate white box test. [2]

**(e)** If each element of `TempArray` stores the value 1.0, state a function call that will be an appropriate black box test. Justify your answer. [3]

### Question 6

A program is to be written to test an insertion sort algorithm. A top-down approach was used in the design of the program. The program, `InsertionSortTester`, has a number of parts:

- input integer values into the array
- output the initial values in the array
- output the sorted values in the array
- perform the insertion sort
- validate the values

(a) Draw a diagram, which exhibits top-down design, for the InsertionSortTester program. [3]

A list of data items is stored in the array `Values`. The pseudocode for the insertion sort algorithm is:

```

01 FOR i ← 2 TO ArraySize
02     Temp ← Values[i]
03     j ← i-1
04     WHILE (j > 0) AND (Values[j] > Temp)
05         Values[j+1] ← Values[j]
06         j ← j-1
07     ENDWHILE
08     Values[j+1] ← Temp
09 ENDFOR

```

(b) The sort algorithm is to be tested using the sequence of numbers: 6, 8, 2 and 1. Copy and complete the trace table given below.

[illegible]

[6]

(c) Explain why this particular algorithm is inefficient for an array where the initial values are already in order. [2]

(d) Give **two** different test cases for the program. Justify your selection in each case. [4]

### Question 7

An algorithm for converting a number  $n$  from denary to octal uses the three built-in functions:

Function	Description	Example
INTMOD (Number, Divisor)	returns the remainder when the first parameter is divided by the second parameter.	INTMOD (7, 3) returns 1
INTDIV (Number, Divisor)	returns the integer part when the first parameter is divided by the second parameter.	INTDIV (7, 3) returns 2
SUBSTR (ThisString, Start, Length)	forms a substring from ThisString, starting at Start (with first index in string zero) and taking Length characters	SUBSTR ("abcd", 1, 2) returns "bc"

Study the following pseudocode:

```

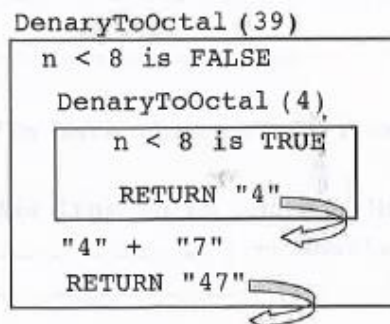
01 FUNCTION DenaryToOctal(n : INTEGER) RETURNS STRING
02   OctalDigits ← "01234567"
03   IF n < 8
04     THEN
05       TempString ← SUBSTR(OctalDigits, n, 1)
06     ELSE
07       // '+' is the concatenation operator
08       TempString ← DenaryToOctal(INTDIV(n, 8)) +
                     SUBSTR(OctalDigits, INTMOD(n, 8), 1)
09   ENDIF
10   RETURN TempString
11 ENDFUNCTION

```

(a) Identify where and why this is a recursive function.

[2]

The diagram shows the execution of the call DenaryToOctal (39).



(b) Draw a similar diagram to show the execution of the call DenaryToOctal (67).

[3]

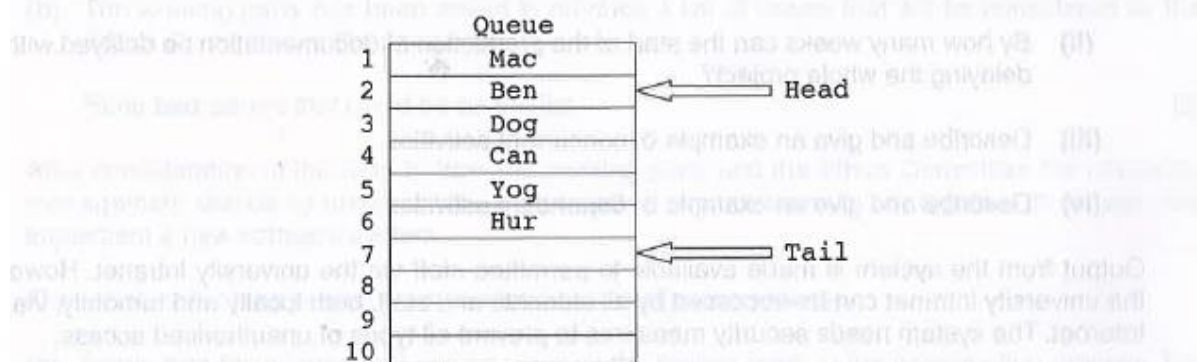
- (c) Changes are to be made to the function `DenaryToOctal()` so that it converts denary numbers to hexadecimal.

Describe the changes:

- that are essential to make the revised function work.
  - that are non-essential but would help with the clarity of the pseudocode.
- [5]

### Question 8

A simple queue data structure is implemented using a one-dimensional array and two pointers, `Head` and `Tail`, as shown:



- (a) Show the state of the above queue after:

- two items, Dap and Eck, are added (in that order)
  - one item is removed.
- [3]

When ten items have been added, this simple queue cannot accept any further items.

- (b) A first attempt at an algorithm for adding an item to this queue is:

```

01 IF .....
02 THEN .....
03     OUTPUT "No more room to add items"
04 ELSE .....
05     INPUT "New item to be added", NewItem
06     Queue[.....] ← NewItem
07     .....
08 ENDIF

```

Write the pseudocode to show the completed lines 01, 06, and 07. [3]

- (c) Give the initial value for `Tail` when the queue is created and justify your answer. [2]

The programmer can reuse the space released after removing an item. This maximises the available space.

- (d) Describe how the algorithm for adding an item should be amended so that the released space is made available. [2]



## Question 9

A real-estate management company owns a number of residential and business units. When the company first acquires a unit, it often requires renovation work. The company records the renovation cost.

A residential unit will be a house or an individual flat within a building. A business unit will be either an office building, a storage unit or a factory.

A residential unit is either advertised for sale, with the company looking to make a profit, or retained for rental. If the unit is sold, the sale price is recorded. If the unit is retained, the monthly rental charged, the start date and length of the rental (in months) are recorded.

A business unit has a long term lease, which is usually 10 years or longer. The company records the nature of the business. It does not offer any of its business units for sale.

Other data recorded for a unit include: purchase price, purchase date, number of rooms, floor space, whether or not a lift is present. The company records whether the house has a garage and whether it has a garden.

A programmer will develop an application, using object-oriented programming to store and process the company's data.

(a) Draw a class diagram, with base class `UNIT`, showing:

- appropriate sub-class(es)
- inheritance
- the properties required
- appropriate methods, including **one** pair of 'get' and 'set' methods for **one** of the properties. [8]

(b) The company has recently purchased a number of units that they want to renovate as a 'block of flats' (a number of self-contained flats in the same building). Once the renovation is complete, the company may offer a block of flats for sale. Alternatively, it may retain the unit and advertise each individual flat for rental.

Explain how this would affect the design in **part (a)**. [3]

(c) (i) Explain the meaning of the term encapsulation. [2]

(ii) Explain the meaning of the term polymorphism. [2]

### Question 10

A programmer implements a linked list of surnames with a start pointer, `StartPtr` and two one-dimensional arrays:


- Array `Data` stores the surnames.
- Array `Ptr` stores the link pointers.
- Both arrays have lower bound 1 and upper bound 3000.

The purpose of procedure `InsertListItem` is to insert a new surname to the linked list.

Assume a function `NextFree()` is available and returns:

- the index position for the array `Data` at which the new surname is to be inserted
- -1 when the `Data` array is full.

The programmer designs the algorithm as follows:

```
01 PROCEDURE InsertListItem(NewSurname : STRING)
02   IF NextFree() = -1
03     THEN
04       OUTPUT "List is full"
05     ELSE
06       // input the surname
07       IF StartPtr = 0
08         THEN
09           StartPtr ← NextFree()
10           Data[StartPtr] ← NewSurname
11         ELSE
12           // traverse the linked list to find the position
13           // at which to insert NewSurname
           
14             

15             

16             

17             

18             

19             

20             

21             

22             

23             

24             

25             

26             

27             

28             

29             

30             

31             

32             

33             

34             

35             

36             

37             

38             

39             

40             

41             

42             

43             

44             

45             

46             

47             

48             

49             

50             

51             

52             

53             

54             

55             

56             

57             

58             

59             

60             

61             

62             

63             

64             

65             

66             

67             

68             

69             

70             

71             

72             

73             

74             

75             

76             

77             

78             

79             

80             

81             

82             

83             

84             

85             

86             

87             

88             

89             

90             

91             

92             

93             

94             

95             

96             

97             

98             

99             

100            

101            

102            

103            

104            

105            

106            

107            

108            

109            

110            

111            

112            

113            

114            

115            

116            

117            

118            

119            

120            

121            

122            

123            

124            

125            

126            

127            

128            

129            

130            

131            

132            

133            

134            

135            

136            

137            

138            

139            

140            

141            

142            

143            

144            

145            

146            

147            

148            

149            

150            

151            

152            

153            

154            

155            

156            

157            

158            

159            

160            

161            

162            

163            

164            

165            

166            

167            

168            

169            

170            

171            

172            

173            

174            

175            

176            

177            

178            

179            

180            

181            

182            

183            

184            

185            

186            

187            

188            

189            

190            

191            

192            

193            

194            

195            

196            

197            

198            

199            

200            

201            

202            

203            

204            

205            

206            

207            

208            

209            

210            

211            

212            

213            

214            

215            

216            

217            

218            

219            

220            

221            

222            

223            

224            

225            

226            

227            

228            

229            

230            

231            

232            

233            

234            

235            

236            

237            

238            

239            

240            

241            

242            

243            

244            

245            

246            

247            

248            

249            

250            

251            

252            

253            

254            

255            

256            

257            

258            

259            

260            

261            

262            

263            

264            

265            

266            

267            

268            

269            

270            

271            

272            

273            

274            

275            

276            

277            

278            

279            

280            

281            

282            

283            

284            

285            

286            

287            

288            

289            

290            

291            

292            

293            

294            

295            

296            

297            

298            

299            

300            

301            

302            

303            

304            

305            

306            

307            

308            

309            

310            

311            

312            

313            

314            

315            

316            

317            

318            

319            

320            

321            

322            

323            

324            

325            

326            

327            

328            

329            

330            

331            

332            

333            

334            

335            

336            

337            

338            

339            

340            

341            

342            

343            

344            

345            

346            

347            

348            

349            

350            

351            

352            

353            

354            

355            

356            

357            

358            

359            

360            

361            

362            

363            

364            

365            

366            

367            

368            

369            

370            

371            

372            

373            

374            

375            

376            

377            

378            

379            

380            

381            

382            

383            

384            

385            

386            

387            

388            

389            

390            

391            

392            

393            

394            

395            

396            

397            

398            

399            

400            

401            

402            

403            

404            

405            

406            

407            

408            

409            

410            

411            

412            

413            

414            

415            

416            

417            

418            

419            

420            

421            

422            

423            

424            

425            

426            

427            

428            

429            

430            

431            

432            

433            

434            

435            

436            

437            

438            

439            

440            

441            

442            

443            

444            

445            

446            

447            

448            

449            

450            

451            

452            

453            

454            

455            

456            

457            

458            

459            

460            

461            

462            

463            

464            

465            

466            

467            

468            

469            

470            

471            

472            

473            

474            

475            

476            

477            

478            

479            

480            

481            

482            

483            

484            

485            

486            

487            

488            

489            

490            

491            

492            

493            

494            

495            

496            

497            

498            

499            

500            

501            

502            

503            

504            

505            

506            

507            

508            

509            

510            

511            

512            

513            

514            

515            

516            

517            

518            

519            

520            

521            

522            

523            

524            

525            

526            

527            

528            

529            

530            

531            

532            

533            

534            

535            

536            

537            

538            

539            

540            

541            

542            

543            

544            

545            

546            

547            

548            

549            

550            

551            

552            

553            

554            

555            

556            

557            

558            

559            

560            

561            

562            

563            

564            

565            

566            

567            

568            

569            

570            

571            

572            

573            

574            

575            

576            

577            

578            

579            

580            

581            

582            

583            

584            

585            

586            

587            

588            

589            

590            

591            

592            

593            

594            

595            

596            

597            

598            

599            

600            

601            

602            

603            

604            

605            

606            

607            

608            

609            

610            

611            

612            

613            

614            

615            

616            

617            

618            

619            

620            

621            

622            

623            

624            

625            

626            

627            

628            

629            

630            

631            

632            

633            

634            

635            

636            

637            

638            

639            

640            

641            

642            

643            

644            

645            

646            

647            

648            

649            

650            

651            

652            

653            

654            

655            

656            

657            

658            

659            

660            

661            

662            

663            

664            

665            

666            

667            

668            

669            

670            

671            

672            

673            

674            

675            

676            

677            

678            

679            

680            

681            

682            

683            

684            

685            

686            

687            

688            

689            

690            

691            

692            

693            

694            

695            

696            

697            

698            

699            

700            

701            

702            

703            

704            

705            

706            

707            

708            

709            

710            

711            

712            

713            

714            

715            

716            

717            

718            

719            

720            

721            

722            

723            

724            

725            

726            

727            

728            

729            

730            

731            

732            

733            

734            

735            

736            

737            

738            

739            

740            

741            

742            

743            

744            

745            

746            

747            

748            

749            

750            

751            

752            

753            

754            

755            

756            

757            

758            

759            

760            

761            

762            

763            

764            

765            

766            

767            

768            

769            

770            

771            

772            

773            

774            

775            

776            

777            

778            

779            

780            

781            

782            

783            

784            

785            

786            

787            

788            

789            

790            

791            

792            

793            

794            

795            

796            

797            

798            

799            

800            

801            

802            

803            

804            

805            

806            

807            

808            

809            

810            

811            

812            

813            

814            

815            

816            

817            

818            

819            

820            

821            

822            

823            

824            

825            

826            

827            

828            

829            

830            

831            

832            

833            

834            

835            

836            

837            

838            

839            

840            

841            

842            

843            

844            

845            

846            

847            

848            

849            

850            

851            

852            

853            

854            

855            

856            

857            

858            

859            

860            

861            

862            

863            

864            

865            

866            

867            

868            

869            

870            

871            

872            

873            

874            

875            

876            

877            

878            

879            

880            

881            

882            

883            

884            

885            

886            

887            

888            

889            

890            

891            

892            

893            

894            

895            

896            

897            

898            

899            

900            

901            

902            

903            

904            

905            

906            

907            

908            

909            

910            

911            

912            

913            

914            

915            

916            

917            

918            

919            

920            

921            

922            

923            

924            

925            

926            

927            

928            

929            

930            

931            

932            

933            

934            

935            

936            

937            

938            

939            

940            

941            

942            

943            

944            

945            

946            

947            

948            

949            

950            

951            

952            

953            

954            

955            

956            

957            

958            

959            

960            

961            

962            

963            

964            

965            

966            

967            

968            

969            

970            

971            

972            

973            

974            

975            

976            

977            

978            

979            

980            

981            

982            

983            

984            

985            

986            

987            

988            

989            

990            

991            

992            

993            

994            

995            

996            

997            

998            

999            

1000           

1001           

1002           

1003           

1004           

1005           

1006           

1007           

1008           

1009           

1010           

1011           

1012           

1013           

1014           

1015           

1016           

1017           

1018           

1019           

1020           

1021           

1022           

1023           

1024           

1025           

1026           

1027           

1028           

1029           

1030           

1031           

1032           

1033           

1034           

1035           

1036           

1037           

1038           

1039           

1040           

1041           

1042           

1043           

1044           

1045           

1046           

1047           

1048           

1049           

1050           

1051           

1052           

1053           

1054           

1055           

1056           

1057           

1058           

1059           

1060           

1061           

1062           

1063           

1064           

1065           

1066           

1067           

1068           

1069           

1070           

1071           

1072           

1073           

1074           

1075           

1076           

1077           

1078           

1079           

1080           

1081           

1082           

1083           

1084           

1085           

1086           

1087           

1088           

1089           

1090           

1091           

1092           

1093           

1094           

1095           

1096           

1097           

1098           

1099           

1100           

1101           

1102           

1103           

1104           

1105           

1106           

1107           

1108           

1109           

1110           

1111           

1112           

1113           

1114           

1115           

1116           

1117           

1118           

1119           

1120           

1121           

1122           

1123           

1124           

1125           

1126           

1127           

1128           

1129           

1130           

1131           

1132           

1133           

1134           

1135           

1136           

1137           

1138           

1139           

1140           

1141           

1142           

1143           

1144           

1145           

1146           

1147           

1148           

1149           

1150           

1151           

1152           

1153           

1154           

1155           

1156           

1157           

1158           

1159           

1160           

1161           

1162           

1163           

1164           

1165           

1166           

1167           

1168           

1169           

1170           

1171           

1172           

1173           

1174           

1175           

1176           

1177           

1178           

1179           

1180           

1181           

1182           

1183           

1184           

1185           

1186           

1187           

1188           

1189           

1190           

1191           

1192           

1193           

1194           

1195           

1196           

1197           

1198           

1199           

1200           

1201           

1202           

1203           

1204           

1205           

1206           

1207           

1208           

1209           

1210           

1211           

1212           

1213           

1214           

1215           

1216           

1217           

1218           

1219           

1220           

1221           

1222           

1223           

1224           

1225           

1226           

1227           

1228           

1229           

1230           

1231           

1232           

1233           

1234           

1235           

1236           

1237           

1238           

1239           

1240           

1241           

1242           

1243           

1244           

1245           

1246           

1247           

1248           

1249           

1250           

1251           

1252           

1253           

1254           

1255           

1256           

1257           

1258           

1259           

1260           

1261           

1262           

1263           

1264           

1265           

1266           

1267           

1268           

1269           

1270           

1271           

1272           

1273           

1274           

1275           

1276           

1277           

1278           

1279           

1280           

1281           

1282           

1283           

1284           

1285           

1286           

1287           

1288           

1289           

1290           

1291           

1292           

1293           

1294           

1295           

1296           

1297           

1298           

1299           

1300           

1301           

1302           

1303           

1304           

1305           

1306           

1307           

1308           

1309           

1310           

1311           

1312           

1313           

1314           

1315           

1316           

1317           

1318           

1319           

1320           

1321           

1322           

1323           

1324           

1325           

1326           

1327           

1328           

1329           

1330           

1331           

1332           

1333           

1334           

1335           

1336           

1337           

1338           

1339           

1340           

1341           

1342           

1343           

1344           

1345           

1346           

1347           

1348           

1349           

1350           

1351           

1352           

1353           

1354           

1355           

1356           

1357           

1358           

1359           

1360           

1361           

1362           

1363           

1364           

1365           

1366           

1367           

1368           

1369           

1370           

1371           

1372           

1373           

1374           

1375           

1376           

1377           

1378           

1379           

1380           

1381           

1382           

1383           

1384           

1385           

1386           

1387           

1388           

1389           

1390           

1391           

1392           

1393           

1394           

1395           

1396           

1397           

1398           

1399           

1400           

1401           

1402           

1403           

1404           

1405           

1406           

1407           

1408           

1409           

1410           

1411           

1412           

1413           

1414           

1415           

1416           

1417           

1418           

1419           

1420           

1421           

1422           

1423           

1424           

1425           

1426           

1427           

1428           

1429           

1430           

1431           

1432           

1433           

1434           

1435           

1436           

1437           

1438           

1439           

1440           

1441         
```



### Question 11

The recursive procedure X has two parameters, Value and Index. The procedure processes the contents of an array, T.

```

01 PROCEDURE X(Value, Index)
02   IF T[Index] > 0
03     THEN
04       IF T[Index] > Value
05         THEN
06           X(Value, Index * 2)
07         ENDIF
08       IF T[Index] < Value
09         THEN
10         X(Value, Index * 2 + 1)
11       ENDIF
12       IF T[Index] = Value
13         THEN
14         OUTPUT "True"
15       ENDIF
16     ENDIF
17 ENDPROCEDURE

```

- (a) (i) State what is meant by a recursive procedure. [1]
- (ii) Give the **two** line numbers which indicate that procedure X is recursive. [1]
- (b) An array T is used to store the data for a binary tree. A program places items in the array in the order in which they joined the tree structure.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
T	17	11	19	9	12	18	23	0	4	0	0	0	0	0	0

- (i) Draw the binary tree for the array T dataset. [3]
- (ii) Copy and then complete the trace table for the procedure call X(18, 1).

Procedure call	Value	Index	Output
1	18	1	

- (iii) Describe the purpose of procedure X. [2]