Holy Family Canossian College  
Mock Examination 2002-2003

Computer Studies Paper 1

2 hours  
This paper must be answered in English

1. There are THREE sections in this paper,  
   Section A, Section B and Section C.

2. Section A carries 40 marks,  
   Section B carries 40 marks, and  
   Section C carries 20 marks

3. All questions are COMPULSORY.  
   Write your answers in the spaces provided in this  
   Question/Answer Book.

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>A2</td>
<td>A3</td>
<td>A4</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total</td>
</tr>
<tr>
<td>B5</td>
<td>B6</td>
<td>B7</td>
<td>B8</td>
<td></td>
<td></td>
<td>B</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Overall</td>
</tr>
</tbody>
</table>
There are three sections in this paper.
Answer all questions in the spaces provided in these question-answer sheets.

Section A (40 marks)
1. a) Explain the following terms.
   i) Register
   (1 mark)
   ii) Accumulator
   (1 mark)

   b) Give 2 advantages and one disadvantage of high-level languages over low-level language.
   (6 marks)

   c) Give 2 differences between compilers and interpreters.
   (2 marks)

   i) Register - storage unit in which data or instruction are stored temporarily for a specific purpose.
   (1 mark)
   ii) Accumulator - in sides the ALU, use to hold the data item to be processed and the results of arithmetic and logical operations. (1 mark)

2. The following spreadsheet is used to keep the assessment results of 100 students.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>name</td>
<td>Maths</td>
<td>English</td>
<td>Chinese</td>
<td>Total mark</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Peter</td>
<td>36</td>
<td>28</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Rose</td>
<td>47</td>
<td>69</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Jack</td>
<td>28</td>
<td>40</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   (a) Write down the formula that should be stored in cell E2?
   (2 marks)

   (b) Column F is used to show the average marks of Chinese, English and Mathematics. Write down the formula that should be stored in cell F2?
   (2 marks)

   (c) Suppose the formula in cell F2 was entered. Instead of setting the formula one by one for the other cells in column F, what commands that can be used such that the average marks for each student can be calculated?
   (2 marks)

   (d) Suppose all the formula in the spreadsheet were entered. What will happen if the content in cell B2 is changed from 36 to 50?
   (2 marks)
(2 marks)

(e) Give one advantage of using an electronic spreadsheet, instead of a calculator, to find the
assessment results of students. 

(2 marks)

a) \( =B2+C2+D2 \) or \( =\text{SUM}(B2..D2) \) 

b) \( \frac{E}{3} \) or \( =\text{SUM}(B2..D2)/3 \) or \( =(B2+C2+D2)/3 \) or \( =\text{average}(B2..D2) \) 

c) copy the formula in F2 and paste the formula of F2 to other cells in column F 

(2 marks)
d) content in Cell E2 and F2 will change to 116 and 38.667 respectively. 

(2 marks)
e) if the data were changed, the computer will update the results automatically, results can
be saved for future use. 

(2 marks)

3. The following are the instruction codes of an assembly language:

<table>
<thead>
<tr>
<th>Mnemonics</th>
<th>Operand</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA</td>
<td>XXX</td>
<td>clear the contents of the accumulator to zero.</td>
</tr>
<tr>
<td>INP</td>
<td>XXX</td>
<td>input data to address XXX</td>
</tr>
<tr>
<td>STA</td>
<td>XXX</td>
<td>store contents of accumulator to address XXX</td>
</tr>
<tr>
<td>LDA</td>
<td>XXX</td>
<td>load accumulator with the contents of address XXX</td>
</tr>
<tr>
<td>JMP</td>
<td>XXX</td>
<td>jump to address XXX</td>
</tr>
<tr>
<td>JPN</td>
<td>XXX</td>
<td>branch to address XXX if the contents of accumulator are negative.</td>
</tr>
<tr>
<td>JPZ</td>
<td>XXX</td>
<td>branch to address XXX if the contents of accumulator are zero.</td>
</tr>
<tr>
<td>INC</td>
<td>XXX</td>
<td>increase the contents of address XXX by 1.</td>
</tr>
<tr>
<td>STP</td>
<td>XXX</td>
<td>stop.</td>
</tr>
</tbody>
</table>

The following program is written in the assembly language defined above:

<table>
<thead>
<tr>
<th>Memory Location</th>
<th>Instruction Code</th>
<th>_operand</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>CLA</td>
<td>0000</td>
<td></td>
</tr>
<tr>
<td>0001</td>
<td>STA</td>
<td>1101</td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td>STA</td>
<td>1110</td>
<td></td>
</tr>
<tr>
<td>0011</td>
<td>INP</td>
<td>1100</td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>LDA</td>
<td>1100</td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td>JPN</td>
<td>1011</td>
<td></td>
</tr>
<tr>
<td>0110</td>
<td>INC</td>
<td>1001</td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td>INC</td>
<td>1101</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>JMP</td>
<td>0011</td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td>JMP</td>
<td>0011</td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td>STP</td>
<td>0000</td>
<td></td>
</tr>
</tbody>
</table>

The input values are in order 3, -4, 8, -2, -4, -6, 0. 

(a) Fill in the following table to show the contents in the memory location 1100, memory location 1101
and memory location 1110 immediately after the execution of the instruction in location 0011 for
each of the passes.

<table>
<thead>
<tr>
<th>Pass</th>
<th>Contents in Memory location 1100</th>
<th>Memory location 1101</th>
<th>Memory location 1110</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Pass</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Second Pass</td>
<td>-4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Third Pass</td>
<td>8</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
(b) What are the contents (in decimals) of the memory locations 1101 and 1110 at the end of the program execution? The contents of the memory locations 1101 and 1110 are 2 and 4, respectively. (2 marks)

(c) Briefly describe the purpose of the program.

To accept a list of input numbers and count the number of positive numbers and negative numbers and store the number in address 1101 and 1110 respectively. The program stops when the input numbers is zero. (1 mark)

4. In a 10-bit word computer, it uses 6 bits to code the mantissa and 4-bits to code the exponent of the floating point number in sign and magnitude form. (bit 3 and bit 9 is the sign bit of mantissa and exponent respectively).

<table>
<thead>
<tr>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M</td>
<td></td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a) Write down the smallest positive number that can be represented by this computer in both denary and binary number. (2 marks)

b) Write down the largest positive number that can be represented by this computer in both denary and binary number. (2 marks)

c) Show how you convert the denary number -9.125 into binary floating-point number stored in the above computer. (5 marks)

d) Name the type of error (if any) found in part c) (1 mark)

a) smallest positive number : bit pattern 010000 1111 (0000011111 is wrong as the number is normalized, the number next to the binary point should be equal to 1)

in binary 0.100001111 = 0.1 x 2^-7 = 2^-8 =1/256 (2 marks)

b) largest positive number: 0.111110111 = 0.11111 x 2^7 = 4+8+16+32+64 =124 (2 marks)

c) -.9125

-1001.001

= -0.1001001 x 2^4

= -0.1001001 x 2^0100

= 1100100100 (5 marks)

d) Truncation error (1 mark)
Section B (40 marks)
Answer all questions in the spaces provided in these question-answer sheets.

5. John wrote a Pascal program so as to count the total number of vowels in a text file named as 'text.txt'.

Suppose the content of the text.txt are as follows:

```
aeiouAEIOU
```

Below is a sample run of the program:

```
The number of vowels in the text file are 12
```

However, there are several mistakes in the main program. Fill in the following table to show the locations of all the mistakes and their corrections. (10 marks)

<table>
<thead>
<tr>
<th>Line Number</th>
<th>Corrected Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>Begin</td>
</tr>
<tr>
<td>90</td>
<td>reset (F);</td>
</tr>
<tr>
<td>100</td>
<td>while not eof (F) do</td>
</tr>
<tr>
<td>120</td>
<td>While not eoln (F) do</td>
</tr>
<tr>
<td>140</td>
<td>Readln (F,C);</td>
</tr>
<tr>
<td>150</td>
<td>Case C of</td>
</tr>
<tr>
<td>180</td>
<td>if (ord(C)=97) or (ord(C)=101) or (ord(C)=105) or (ord(C)=111) or (ord(C)=117)</td>
</tr>
<tr>
<td>190</td>
<td>then count := count + 1</td>
</tr>
<tr>
<td>240</td>
<td>write('The number of vowels in the text file are ', count)</td>
</tr>
</tbody>
</table>

10  Program q6;
20  Var
30  F : text;
40  C : char;
50  Count: integer;
60  Start
70  Count := 0;
80  Assign (F, 'text.txt');
90  Reset;
100 while not eof do
110 begin
120 While not eoln (F) do
130  Begin
140  Readln (F,C);
150  Case F of
160   'A', 'E', 'I', 'O', 'U' : count := count + 1
170   end;
180  if (val(C)=97) or (val(C)=101) or (val(C)=105) or (val(C)=111) or (val(C)=116)
190  then count +1 := count
200 end;
210  readln (F)
220  end;
230  close(F);
240  write('The number of vowels in the text file are count')
250  end.
6. Procedures KR and KL are given below:

Procedure KR(var IPSTRING:string);
begin
    IPSTRING:=copy(IPSTRING,1,length(IPSTRING)-1)
end;

Procedure KL(var IPSTRING:string);
begin
    IPSTRING:=copy(IPSTRING,2,length(IPSTRING)-1)
end;

(a) Suppose IPSTRING is a string variable storing the string ‘123456’, write down the contents of
IPSTRING after the execution of the following statements:
(i) KR(IPSTRING)
(ii) KL(IPSTRING)

(b) Suppose IPSTRING is a string variable storing the string ‘123456’, write down the contents of
IPSTRING after the execution of the following statement:
KRN(IPSTRING,2)

(c) What will be the contents of IPSTRING after the execution of the statement
KRN(IPSTRING,N) for each of the following cases?
(i) N <= 0
(ii) N >= length (IPSTRING)

7. The algorithm below asks the user to input a sequence of marks and then output the maximum
mark of the sequence.

Step 1: Declare K, Mark, Num and MaxMark as integer variable.
Step 2: Display 'How many marks are there?'
Step 3: read a number into variable Num
Step 4: initialize MaxMark to 0
Step 5: use for loop to repeat steps 6-9 (use variable K, start from K = 1 to K=Num-1)
Step 6: repeat steps 7-8 until Mark is within the range 0-100
Step 7: read a number into variable Mark
Step 8: if Mark is not within the range 0-100 then display 'out of range'
Step 9: if Mark is greater than MaxMark then assign the value of Mark to MaxMark
Step 10: output MaxMark

Convert the algorithm into a Pascal program. (10 marks)

```pascal
program q7;
var K, Mark, Num, MaxMark : integer;
begin
    writeln ('How many marks are there?');
    readln (Num);
    MaxMark := 0;
    For K := 1 to Num-1 do
        Begin
```
Repeat
Readln (Mark);
If (Mark < 0) or (Mark > 100) then writeln (‘out of range’)
Until (Mark >= 0) and (Mark <= 100)
If Mark > MaxMark then MaxMark := Mark;
End;
Writeln (MaxMark);
End.

1 mark for each step

8. A school, with at most 1500 students, carried out a survey to find out how hard working its students are. The students were asked the number of hours they spent on study per week. The data were entered into a computer system. A program counted the number of students who studied 15 hours or more per week. These students were classified as hard working students. Students who studied 5 to 14 hours per week were classified as average students. Those who studied 4 hours or less a week were classified as lazy students. The three frequencies, together with the average hours spent on studying per week, were printed to the screen.

The program SURVEY is to carry out the above task. A sample output of the program is shown below. The data following the question marks are entered by the user. All other items are outputs from the program.

Enter the number -1 to stop data entry
Enter the number of hours spent on studying
? 16
? 12
? 4
? 11
? -1
1 of the students are hard working
2 of the students are average
1 of the students are lazy
These 4 students spent an average of 10.8 hours per week on study

The program consists of subprograms INIT, INPUT_HOURS, CALCULATE, and SHOW_RESULT. The basic structure of the program is shown below:

```pascal
program SURVEY(input, output);
var student, chard, cave, claz, sum, jj : integer;
hour: array [1..1500] of integer;
aver :real;

procedure Init;
begin
student := 0; sum:=0;chard :=0; cave := 0; claz :=0;
for JJ :=1 to 1500 do hour[jj] :=0;
end;

procedure InputHours;
begin
end;

procedure Calculate;
begin
end;

procedure ShowResults;
begin
writeln (chard, ' of the students are hard working');
writeln (cave, ' of the students are average');
writeln (claz, ' of the students are lazy');
writeln;
write ('These ', student,' students spent an average of ');
```
writeln (aver:4:1, ' hours per week on study.'); end;

begin {main program}  
Init;  
InputHours;  
Calculate;  
ShowResults;  
end;

The variable to be used are listed below:

<table>
<thead>
<tr>
<th>Variables</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>hour</td>
<td>the number of hours spent on studying be each student</td>
</tr>
<tr>
<td>aver</td>
<td>the average number of hours spent by all the students</td>
</tr>
<tr>
<td>student</td>
<td>the number of students who took part in this survey</td>
</tr>
<tr>
<td>nhard</td>
<td>number of hard working students</td>
</tr>
<tr>
<td>nave</td>
<td>number of average students</td>
</tr>
<tr>
<td>nlaz</td>
<td>number of lazy students</td>
</tr>
<tr>
<td>jj</td>
<td>loop counter</td>
</tr>
<tr>
<td>sum</td>
<td>the total number of study hours of all students</td>
</tr>
</tbody>
</table>

Using the above variables only, write the following parts of the program in Pascal. 

a) Write the subprogram InputHours which shows the necessary prompt message, accepts inputs for number of hours of each student and updates the total number of students who took part in this survey. The input should be terminated by -1. (5 marks)

```pascal
procedure input_hours;
begin
    writeln ('Enter the number -1 to stop data entry');
    writeln ('Enter the number of hours spent on studying');
    for JJ := 1 to 1500 do
    begin
        write ('?');
        readln (hour[jj]);
        if Hour[jj] <> -1 then student := student + 1 else JJ := 1500;
    end;
end;
```

b) Write the subprogram calculate which sorts out the inputs into the three classifications, and finds the average studying hours. (5 marks)

```pascal
procedure calculate;
begin
    for jj := 1 to student do
    begin
        if hour[jj] >= 15 then chard := chard +1;
        if (hour[jj] >= 5) and (hour[jj] < 15) then cave := cave +1;
        if hour[jj] < 5 then claz := claz +1;
    end;
    sum := sum + hour[jj];
    aver := sum/ student;
end;
```
9. An alumni association collects membership fees from members once every year and sends newsletters to its members from time to time.

The association does not use a computer for its operation. In its manual system, there are
- a mailing list,
- member registration cards, and
- a fee record list.

The mailing list consists of address labels as shown below:

Mr Chan Tai Man
136 High Street
3/F, Hong Kong

Miss Cheung Siu Ling
149 Prince Edward Road
2/F, Kowloon

... 

The registration cards are sorted in alphabetical order of members’ names. Each card is in the following format:

<table>
<thead>
<tr>
<th>Name</th>
<th>Sex</th>
<th>Address</th>
<th>Telephone</th>
<th>HK ID Number</th>
<th>Year of Graduation</th>
<th>Occupation</th>
<th>Membership Number</th>
</tr>
</thead>
</table>

The fee record list is used to keep a record of the payment of fees. It has the following format:

<table>
<thead>
<tr>
<th>Name</th>
<th>92</th>
<th>93</th>
<th>94</th>
<th>95</th>
<th>96</th>
<th>97</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chan Tai Man</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Au Kin Leung</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Poon Tak Kin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When a person applies for membership in the association, he needs to fill in a registration card and pay the membership fee. After verifying the information on the card, the association will place the card in the right place and add the necessary information to the mailing list and the fee record list for the new member.

A member is classified as INACTIVE if

the member has not paid the membership fee for three consecutive years;
or the member has not paid the membership fee for one year and is out of contact with the association.

The association removes the membership of all inactive members at the end of each year.
The association is now planning to computerize its manual system.

(a) The person applying for membership should be required to produce some relevant documents for the verification of information on his registration card. Name two such documents and explain how they can be used for verification.

(4 marks)
(b) Name two serious problems in the design of the fee record list. Suggest how to overcome these problems.

(4 marks)

(c) Suppose the association has computerized its system using an application software. Give two reasons why a database package would be more suitable for the association to use than a word processing package.

(2 marks)

(d) A database file called MEMBER is now used to hold all members’ information. The following shows part of the structure of MEMBER:

<table>
<thead>
<tr>
<th>FIELD</th>
<th>NAME</th>
<th>TYPE</th>
<th>WIDTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>NAME</td>
<td>CHARACTER</td>
<td>25</td>
</tr>
<tr>
<td>002</td>
<td>SEX</td>
<td>Numeric</td>
<td>01</td>
</tr>
</tbody>
</table>

(i) Distinguish between a character field and a numeric field.
(ii) Explain why SEX can be either a character field or a numeric field.

(4 marks)

(e) It happens that the chairperson of the association is the manager of a credit card company. The company has printed pamphlets for the promotion of the company’s credit card. Explain why it is an invasion of privacy for the chairperson to make use of the association’s mailing list to send the pamphlets directly to members of the association.

a) Document 1: HK ID card 1
   Document 2: graduation certificate 1

   Name, sex, ID number are checked against those in the ID card 1
   Year of graduation is checked with the certificate 1

b) Problem 1: It does not distinguish members of the same name 1
   Problem 2: Difficult to find a member because they are not arrange in order. 1
   Solution: add membership number/ID number column 1
   put names in order 1

c) There are at least two files and they are related if records put onto database. They can be accessed efficiently and data redundancy is minimized. any two 1 mark each

d) (i) a character field stores symbol(s) but numeric field stores value which can do arithmetic operations 1+1
   (ii) Sex has only two value (M/F) (1/0); therefore, either a character field or numeric field can store those values 1+1

e) The purpose of the mailing list is violated. Its original purpose is for communication between the association and the members only. 2

   Effective communication 4 well organized 2
   grammar 1
   relevant 1

(2 marks)
A Partial Character List for ASCII

<table>
<thead>
<tr>
<th>Character</th>
<th>ASCII</th>
<th>Character</th>
<th>ASCII</th>
<th>Character</th>
<th>ASCII</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>48</td>
<td>J</td>
<td>74</td>
<td>d</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>49</td>
<td>K</td>
<td>75</td>
<td>e</td>
<td>101</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>L</td>
<td>76</td>
<td>f</td>
<td>102</td>
</tr>
<tr>
<td>3</td>
<td>51</td>
<td>M</td>
<td>77</td>
<td>g</td>
<td>103</td>
</tr>
<tr>
<td>4</td>
<td>52</td>
<td>N</td>
<td>78</td>
<td>h</td>
<td>104</td>
</tr>
<tr>
<td>5</td>
<td>53</td>
<td>O</td>
<td>79</td>
<td>i</td>
<td>105</td>
</tr>
<tr>
<td>6</td>
<td>54</td>
<td>P</td>
<td>80</td>
<td>j</td>
<td>106</td>
</tr>
<tr>
<td>7</td>
<td>55</td>
<td>Q</td>
<td>81</td>
<td>k</td>
<td>107</td>
</tr>
<tr>
<td>8</td>
<td>56</td>
<td>R</td>
<td>82</td>
<td>l</td>
<td>108</td>
</tr>
<tr>
<td>9</td>
<td>57</td>
<td>S</td>
<td>83</td>
<td>m</td>
<td>109</td>
</tr>
<tr>
<td>;</td>
<td>58</td>
<td>T</td>
<td>84</td>
<td>n</td>
<td>110</td>
</tr>
<tr>
<td>;</td>
<td>59</td>
<td>U</td>
<td>85</td>
<td>o</td>
<td>111</td>
</tr>
<tr>
<td>&lt;</td>
<td>60</td>
<td>V</td>
<td>86</td>
<td>p</td>
<td>112</td>
</tr>
<tr>
<td>=</td>
<td>61</td>
<td>W</td>
<td>87</td>
<td>q</td>
<td>113</td>
</tr>
<tr>
<td>&gt;</td>
<td>62</td>
<td>X</td>
<td>88</td>
<td>r</td>
<td>114</td>
</tr>
<tr>
<td>?</td>
<td>63</td>
<td>Y</td>
<td>89</td>
<td>s</td>
<td>115</td>
</tr>
<tr>
<td>@</td>
<td>64</td>
<td>Z</td>
<td>90</td>
<td>t</td>
<td>116</td>
</tr>
<tr>
<td>A</td>
<td>65</td>
<td>[</td>
<td>91</td>
<td>u</td>
<td>117</td>
</tr>
<tr>
<td>B</td>
<td>66</td>
<td>\</td>
<td>92</td>
<td>v</td>
<td>118</td>
</tr>
<tr>
<td>C</td>
<td>67</td>
<td>]</td>
<td>93</td>
<td>w</td>
<td>119</td>
</tr>
<tr>
<td>D</td>
<td>68</td>
<td>^</td>
<td>94</td>
<td>x</td>
<td>120</td>
</tr>
<tr>
<td>E</td>
<td>69</td>
<td>–</td>
<td>95</td>
<td>y</td>
<td>121</td>
</tr>
<tr>
<td>F</td>
<td>70</td>
<td>‘</td>
<td>96</td>
<td>z</td>
<td>122</td>
</tr>
<tr>
<td>G</td>
<td>71</td>
<td>a</td>
<td>97</td>
<td>{</td>
<td>123</td>
</tr>
<tr>
<td>H</td>
<td>72</td>
<td>b</td>
<td>98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>73</td>
<td>c</td>
<td>99</td>
<td>}</td>
<td>125</td>
</tr>
</tbody>
</table>

List of Operators and Reserved Words

+, -, *, /, div, mod, >, <, >=, <=, <>, and, or, not, sqrt, trunc, round, random, abs, +(string concatenation), length, ord, chr, val, str, copy, const, type, procedure, var, integer, real, char, string, boolean, true, false, text, array...of, program, input, output, begin...end, :=, for...to/downto... do, (*...*), if...then...else, while...do, repeat...until, case...of...end, write, writeln, read, readln, assign, reset, rewrite, eof, eoln, close, sqr, sin, cos, arctan, ln, exp, with...do

END OF PAPER