# Table of Contents

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statement of the Problem</td>
<td>(2)</td>
</tr>
<tr>
<td>Assumptions and Limitations</td>
<td>(3)</td>
</tr>
<tr>
<td>List of Variables</td>
<td>(4)</td>
</tr>
<tr>
<td>Flowchart</td>
<td>(5)</td>
</tr>
<tr>
<td>Trace Table</td>
<td>(12)</td>
</tr>
<tr>
<td>Program Listing</td>
<td>(13)</td>
</tr>
<tr>
<td>Test Data</td>
<td>(17)</td>
</tr>
<tr>
<td>Results</td>
<td>(18)</td>
</tr>
</tbody>
</table>
Statement of the Problem

The program accepts thirty students who completed the Common Entrance Examination, stores; their first name, last name, primary school, identification number, score and two secondary school choices. It verifies whether a score is correct or not and assigns each student into a secondary school depending on their score and the school's pass mark. It counts the number of students who are assigned to each school. It stores each child's name & IDNO, in the secondary school list for which he passed. If a student does not pass for any school, it prints their result as, "Unassigned". After, it calculates and prints the average or national mean. Finally, it prints the school lists and the number of students assigned to each school.
Assumptions and Limitations

Limitations

➢ The program does not cater for the first and second choice with the same character.

➢ The program will not detect wrong identification numbers. Furthermore, the identification numbers were not checked for correct format; even copies can be entered with error.

Assumptions

➢ The program assumes that no student gets a score of; 0 and 100.

➢ The program assumes that each Secondary School can hold up to 30 students.

➢ The program assumes that choices are entered in upper case only and does not specifically cater for choices in lower case.
List of Variables

Fname: holds the student first name
Lname: holds the student last name
Pschool: holds the student primary school
IDNO: holds the student identification number
Score: holds the student score percentage
Fchoice: holds the student first secondary school choice
Schoice: holds the student second secondary school choice
Validscore: determines whether a score is valid or not
ValidFc: determines whether the first choice is valid or not
ValidSc: determines whether the second choice is valid or not
QC_list: stores the first name, last name and identification number of all students assigned to
Queens College.
SVH_list: stores the first name, last name and identification number of all students assigned to
Southern Valley High
HV_list: stores the first name, last name and identification number of all students assigned to
Harbour View
QC_Count: counts and holds the number of students who passed for Queens College
SVH_Count: counts and holds the number of students who passed for Southern Valley High
HV_Count: counts and holds the number of students who passed for Harbour View
Student: identifies a student by number (used with a For loop as a counter variable).
TScore: holds the total score (all scores of the students combined)
Avg: holds the average score (the total score divided by the total number of students)
Flowchart

START

QC_Count ← 0
SVH_Count ← 0
HV_Count ← 0
TScore ← 0
Student ← 0

Print "Enter First Name, Last Name, Primary School, IDNO."
Read Fname, Lname, Pschool, IDNO

Print "Enter Score"  
Read Score

If Score <= 0 or >= 100
Selection:
No

TScore ← TScore + Score

Print "Enter First Choice"  
Read Fchoice

P1

{ 5 }
If Score >= 55 and Schoice = ‘H’

Output
Print Fname, Lname, Pschool, Score
Print ‘Unassigned’

Print Fname, Lname, Pschool, Score
Print ‘Queens College’

QC_Count ← QC_Count + 1

QC_list[QC_Count] ← IDNO + Fname + Lname

Storage in arrays

Print Fname, Lname, Pschool, Score
Print ‘Southern Valley High’

SVH_Count ← SVH_Count + 1

SVH_list[SVH_Count] ← IDNO + Fname + Lname

CG

CG
Print Fname, Lname, Pschool, Score
Print 'Harbour View'

HV_Count ← HV_Count + 1

HV_list[HV_Count] ← IDNO + Fname + Lname

CG

Student ← Student + 1

If Student = 30

RE

Yes

No

C1
C1

Avg ← TScore/30

Print Avg

Student ← 0

Student ← Student + 1

Print QC List[Student]

If Student = QC_Count

Yes

Print QC_Count, "passed for Queens College!"

No

B1

{ 9 }
B1

Student ← 0

Student ← Student + 1

Print SVH_List[Student]

If Student = SVH_Count

Yes

Print SVH_Count, "passed for Southern Valley High!"

No

C1
C1

Student ← 0

Student ← Student + 1

Print HV_List[Student]

If Student = HV_Count

Yes

Print HV_Count, "passed for Harbour View!"

No

END
## Test Data

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>IDNO</th>
<th>Primary School</th>
<th>Score %</th>
<th>FC</th>
<th>SC</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Gabriel</td>
<td>GJ01</td>
<td>Anglican Primary</td>
<td>77.3</td>
<td>Q</td>
<td>H</td>
<td>Pass</td>
</tr>
<tr>
<td>Kain</td>
<td>Godspell</td>
<td>GK02</td>
<td>Anglican Primary</td>
<td>84.6</td>
<td>Q</td>
<td>S</td>
<td>Pass</td>
</tr>
<tr>
<td>Laura</td>
<td>Smith</td>
<td>SL03</td>
<td>Anglican Primary</td>
<td>59.2</td>
<td>S</td>
<td>H</td>
<td>Fail</td>
</tr>
<tr>
<td>Andrea</td>
<td>Williams</td>
<td>WA04</td>
<td>Anglican Primary</td>
<td>60.8</td>
<td>S</td>
<td>H</td>
<td>Pass</td>
</tr>
<tr>
<td>Anita</td>
<td>Paul</td>
<td>PA05</td>
<td>Anglican Primary</td>
<td>89.2</td>
<td>Q</td>
<td>S</td>
<td>Pass</td>
</tr>
<tr>
<td>Lucky</td>
<td>Acer</td>
<td>AL06</td>
<td>Anglican Primary</td>
<td>84.6</td>
<td>Q</td>
<td>S</td>
<td>Pass</td>
</tr>
<tr>
<td>Eledoya</td>
<td>Civelle</td>
<td>CE07</td>
<td>Anglican Primary</td>
<td>52.3</td>
<td>Q</td>
<td>S</td>
<td>Pass</td>
</tr>
</tbody>
</table>
Trace Table

<table>
<thead>
<tr>
<th>Student IDNO</th>
<th>Fname</th>
<th>Lname</th>
<th>Score</th>
<th>FC</th>
<th>SC</th>
<th>School Assigned</th>
<th>QC_Count</th>
<th>SVH_Count</th>
<th>HV_Count</th>
<th>TScore</th>
<th>Avg</th>
<th>Print</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GJ01</td>
<td>John</td>
<td>Gabe</td>
<td>Q</td>
<td>H</td>
<td>Queen's College</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>77.3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>GK02</td>
<td>Kain</td>
<td>Godspell</td>
<td>Q</td>
<td>S</td>
<td>Queen's College</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>161.9</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>SL03</td>
<td>Laura</td>
<td>Smith</td>
<td>S</td>
<td>H</td>
<td>Harbour View</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>221.1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>WA04</td>
<td>Andrea</td>
<td>Williams</td>
<td>S</td>
<td>H</td>
<td>Harbour View</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>281.9</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>PA05</td>
<td>Anita</td>
<td>Paul</td>
<td>Q</td>
<td>S</td>
<td>Queen's College</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>371.1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>AL06</td>
<td>Lucky</td>
<td>Acer</td>
<td>S</td>
<td>H</td>
<td>Queen's College</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>455.7</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>CE07</td>
<td>Eledoya</td>
<td>Civelle</td>
<td>Q</td>
<td>S</td>
<td>Unassigned</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>508.0</td>
<td>0</td>
</tr>
</tbody>
</table>

QC = Queens College
SVH = Southern Valley High
HV = Harbour View

Some variables identified.

Some changes correctly demonstrated.

Missing from the trace table is the columns indicating selection control and looping control structures.
Program Listing

Program CE_School_Assigned;

{This program uses the scores and the two choices of students who have completed the Common Entrance Examination to assign each student to a secondary school. It also prints the average score and the three secondary school lists.}

Uses WinCrt;

Var

Fname,Lname,Pschool,IDNO: string;
Fchoice,Schoice: char;
Validscore,ValidFc,ValidSc: string;
QC_list: array[1..30] of string;
SVH_list: array[1..30] of string;
HV_list: array[1..30] of string;
Student,QC_Count,SVH_Count,HV_Count: integer;
Score,Avg,TScore: real;

Begin

{Initializations}
QC_Count:= 0;
SVH_Count:= 0; TScore:=0;
HV_Count:= 0;

For Student:= 1 to 30 do

{Input of student data}
Begin

writeln;
writeln('Enter The First Name:');
readln(Fname);
writeln('Enter The Last Name:');
readln(Lname);
writeln('Enter The IDNO:');
readln(IDNO);
writeln('Enter The Primary School:');
readln(Pschool);

Repeat

writeln('Enter The Score:');
readln(Score);

until {End of student data}


if (Score <= 0) or (Score > 100) Then
Begin
Validscore := 'No';
Writein ('invalid Score!'):
EndElse
Validscore := 'Yes';

FScore := TScore + Score;

Repeat
Writeln ('Enter The First Choice and Second Choice:');
readin (Fchoice);
readln (Schoice);

Case Fchoice of
'S': ValldFc := 'Yes';
End;

Case Schoice of
'S': ValldSc := 'Yes';
End;

End;
Else if (Score >= 55) and Begin

  Writeln(Fname, ',Lname, ',Pschool, ',Score);
  Writeln('Congratulations, You Are Accepted To Harbour View!');
  HV_Count:= HV_Count +1;
  HV_list[HV_Count] := IDNO + ' ' + Fname + ' ' + Lname;

End
Else

If (Score >= 62) And ((Fchoice = 'S') or (Schoice = 'S')) Then

Begin

  Writeln(Fname, ',Lname, ',Pschool, ',Score);
  Writeln('Congratulations, You Are Accepted To Southern Valley High!');
  SVH_Count:= SVH_Count +1;
  SVH_list[SVH_Count] := IDNO + ' ' + Fname + ' ' + Lname;

End
Else

If (Score >= 55) and (Schoice = 'H') Then

Begin

  Writeln(Fname, ',Lname, ',Pschool, ',Score);
  Writeln('Congratulations, You Are Accepted To Harbour View!');
  HV_Count:= HV_Count +1;
  HV_list[HV_Count] := IDNO + ' ' + Fname + ' ' + Lname;

END
Else

Writeln('Unassigned');

End;

{Average calculations and printing of secondary school lists.}

Writeln;
Avg:= TScore/30;
Writeln('Average = ', Avg:2:1);

Writeln;
For Student:= 1 to QC_Count do
Writeln(QC_list[Student]);

Writeln;
Writeln(QC_Count, ',Students Have Been Assigned To Queens College!');

Loop entering data into an array

manipulating an array
WriteIn;
For Student:= 1 to SVH_Count do
WriteIn (SVH_list [Student]);

WriteIn;
WriteIn (SVH_Count,' ', 'Students Have Been Assigned To Southern Valley High!');

WriteIn;
For Student:= 1 to HV_Count do
WriteIn (HV_list [Student]);

WriteIn;
WriteIn (HV_Count,' ', 'Students Have Been Assigned To Harbour View!');
End.
Results

Enter The First Name: John
Enter The Last Name: Gabriel
Enter The IDNO: GJ01
Enter The Primary School: Anglican Primary
Enter The Score: 77.3
Enter The First Choice and Second Choice: QH
Congratulations, You Are Accepted To Queens College!

Enter The First Name: Rain
Enter The Last Name: Godspell
Enter The IDNO: GKO2
Enter The Primary School: Anglican Primary
Enter The Score: 84.6
Enter The First Choice and Second Choice: QS
Congratulations, You Are Accepted To Queens College!

Enter The First Name: Laura
Enter The Last Name: Smith
Enter The IDNO: SL03
Enter The Primary School: Anglican Primary
Enter The Score: 59.2
Enter The First Choice and Second Choice: SH
Congratulations, You Are Accepted To Harbour View!

Program compiled —
Output correct for most values —

User-friendly —
Enter The First Name: Andrea
Enter The Last Name: Williams
Enter The EDNA: Wf1114
Enter The Primary School: Anglican Primary
Enter The Score: 60.8
Enter The First Choice and Second Choice: 

Congratulations, You Are Accepted To Harbour View!

Enter The First Name: Anitta
Enter The Last Name: Paul
Enter The EDNA: PAO5
Enter The Primary School: Anglican Primary
Enter The Score: 89.2
Enter The First Choice and Second Choice: 

Congratulations, You Are Accepted To Queens College!

Enter The First Name: Lucky
Enter The Last Name: Acer
Enter The EDNA: ALO6
Enter The Primary School: Anglican Primary
Enter The Score: 101
Invalid Score!
Enter The Score: 84.6
Enter The First Choice and Second Choice: 

Congratulations, You Are Accepted To Queens College!
Eledopa
Civelle
CE07
Anglican Primary
52.3
H
O
Invalid First Choice!
Invalid Second Choice!
Q
S
Unassigned
Average = 72.6
GJ01 John Gabriel
GK02 Kain Godspell
PA05 Anitta Paul
AL06 Lucky Acer
4 Students Have Been Assigned To Queens College!
0 Students Have Been Assigned To Southern Valley High!
SL03 Laura Smith
WA04 Andrea Williams
2 Students Have Been Assigned To Harbour View!
Comments

Program Implementation

The candidate provided evidence of program code and program output. Based on the program output, it was clear that the program had compiled successfully and the candidate inputted data to test every condition and constraint.

The candidate used more than three variable types and the appropriate variables were initiated.

The program was well-sequenced.

The candidate used the various forms of the IF statement. The candidate also included a CASE statement, even though that was not required as stated in the syllabus.

Different types of looping structures were used. Some were used to input/output data into arrays.

The program was not well documented. The candidate had a statement of the problem, however, it did not include the author’s name, date created, and very little internal documentation throughout the program.

The program code was well indented.

The program was very user-friendly. The instructions given were easy to follow.

In the program implementation section, this candidate would have received 14 out of the possible 15 marks.

Algorithm

This candidate’s problem solving and programming sample was well presented. The candidate submitted the sample with a cover page and table of contents with page numbers.

The candidate provided a detailed problem statement which also included limitations and assumptions made by the programmer.

The candidate printed a variable listing with descriptions. Missing was the intended data types (character, numeric).

The flow chart was well presented. It had selection structures but no looping structure was presented.

For the algorithm section, this candidate would have received 9 out of the possible 10 marks.
## Trace Table

The attempt of tracing the flowchart by the candidate was good. A few variable names were missing and the selection structures were not represented in the trace table to show logic flow. A column for each selection structure should have been present. This would have made the trace table more robust. The candidate would have received 3 out of the possible 5 marks.