IT Programming
SBA
May/June (2011)

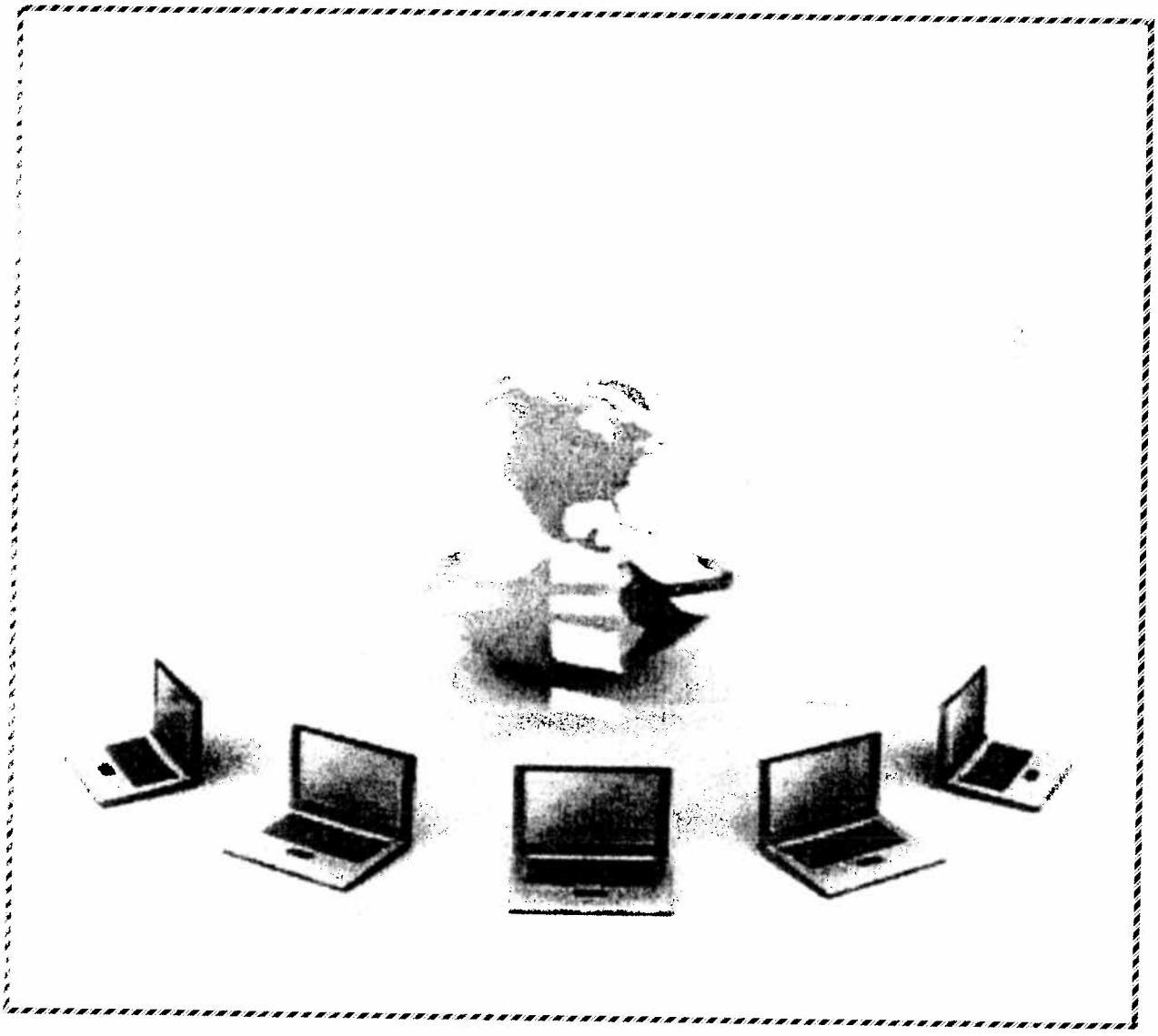


Table of Contents

Contents

Page

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 Habour 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

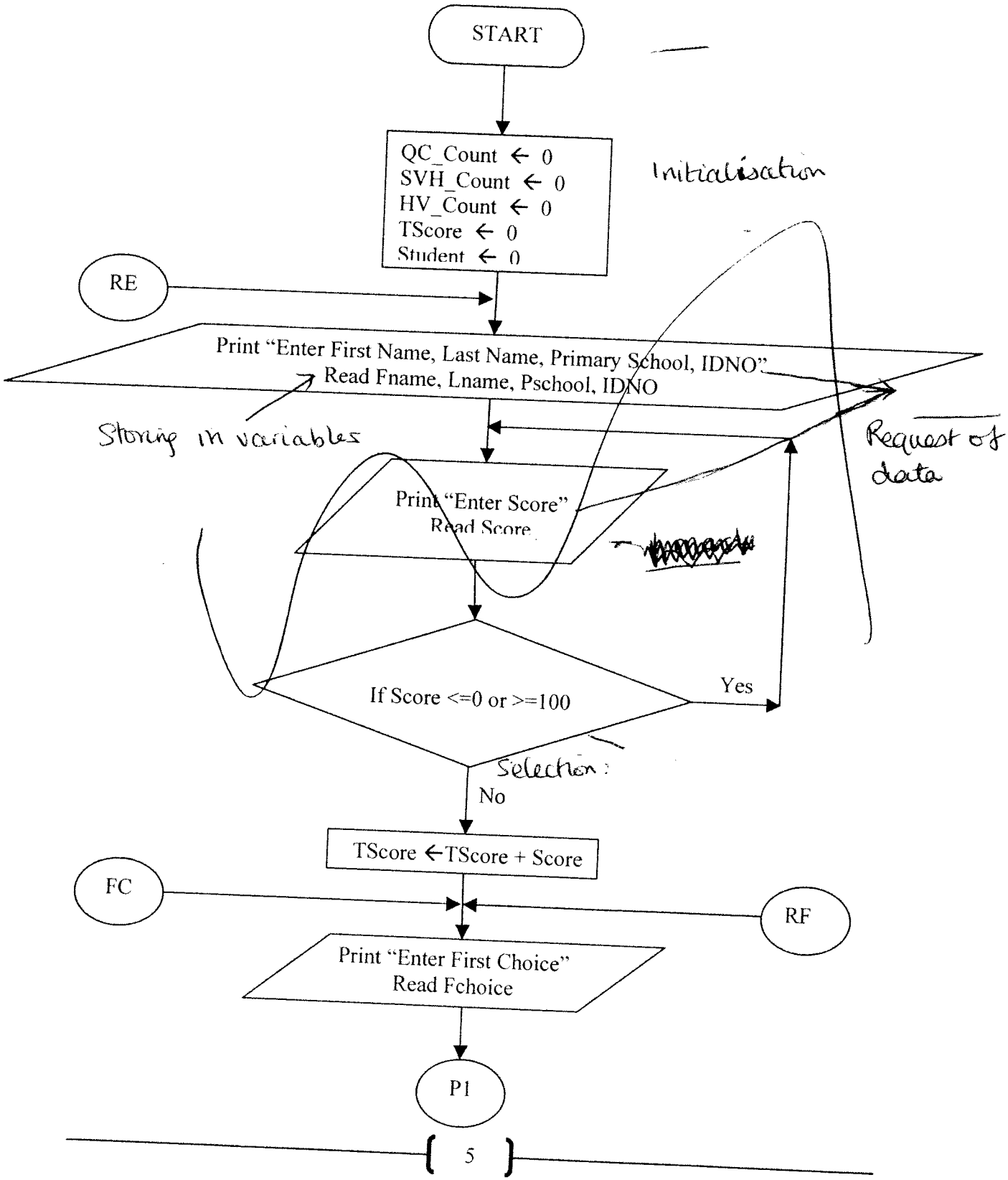
HV_Count: counts and holds the number of students who passed for Habour View

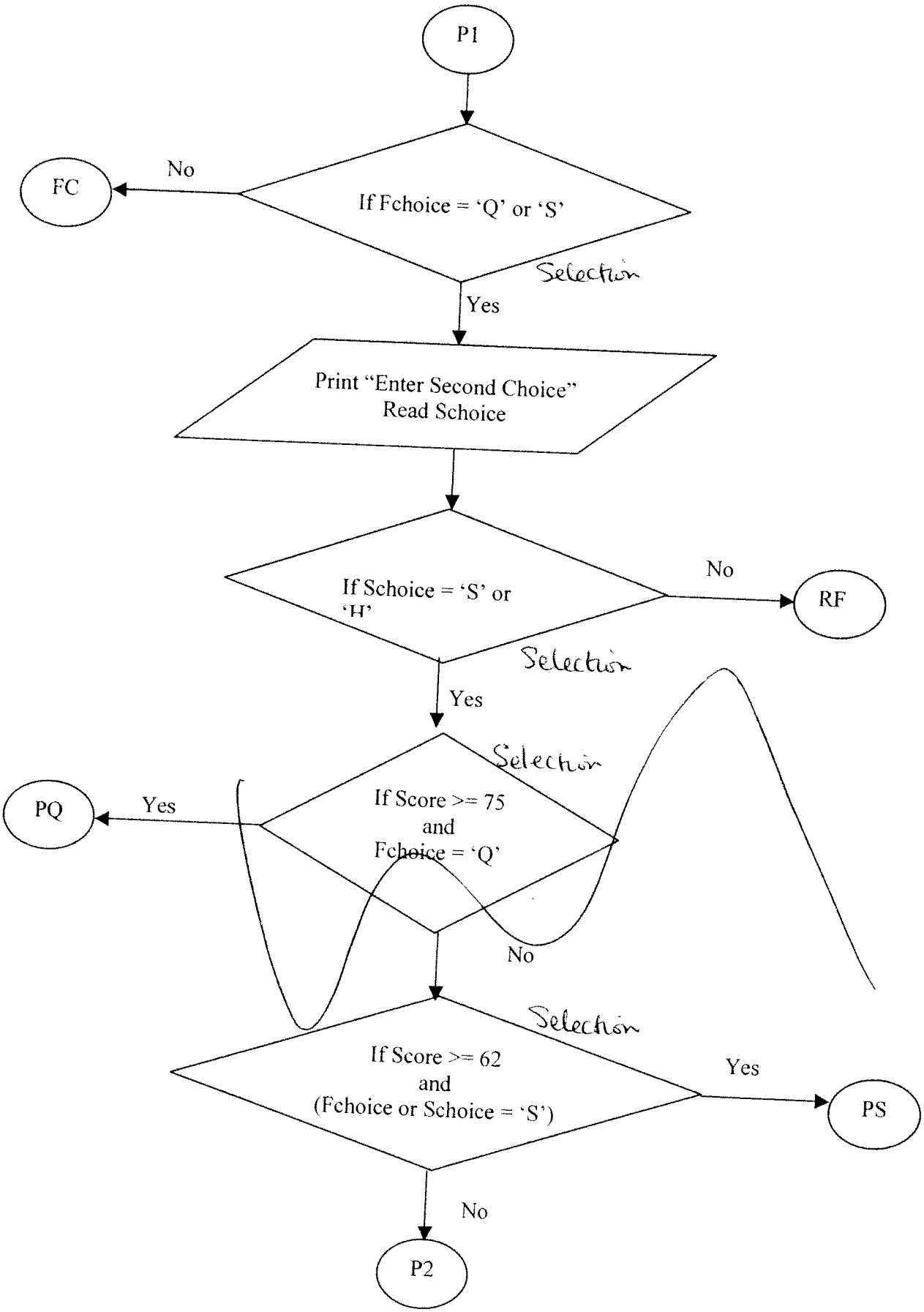
Student: identifies a student by number (*used with a For loop as a counter variable*).

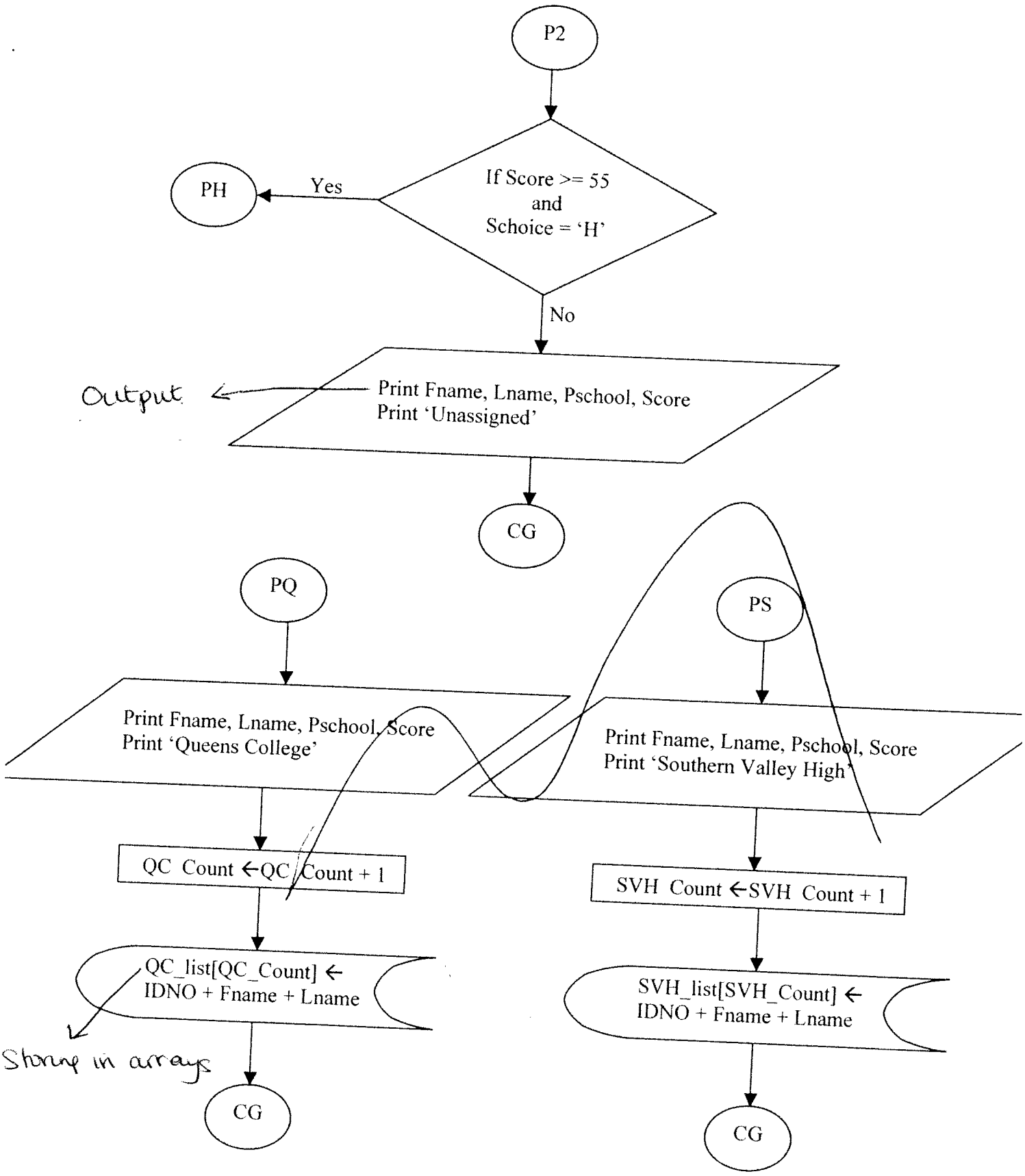
TScore: holds the total score (*all scores of the students combined*)

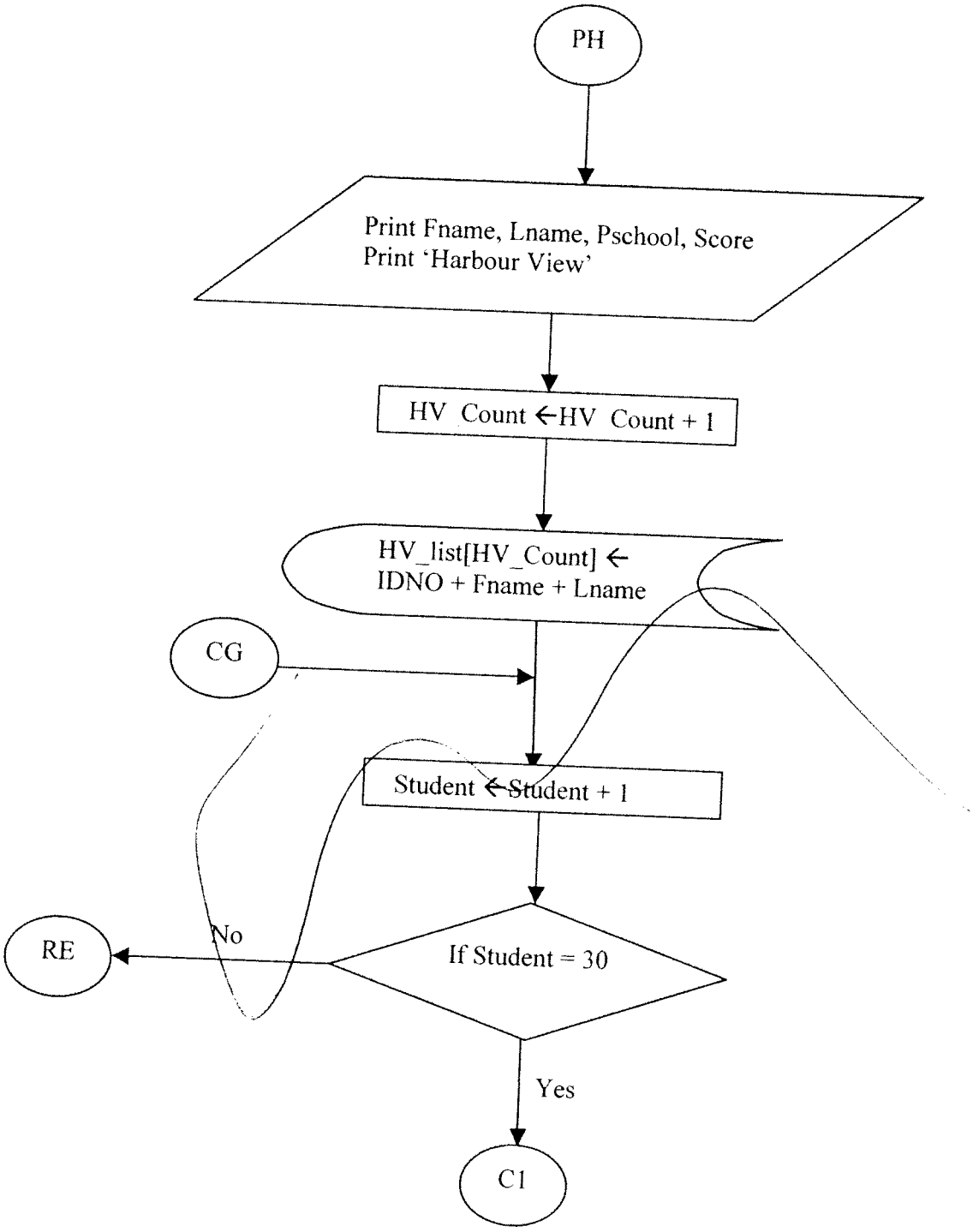
Avg: holds the average score (*the total score divided by the total number of students*)

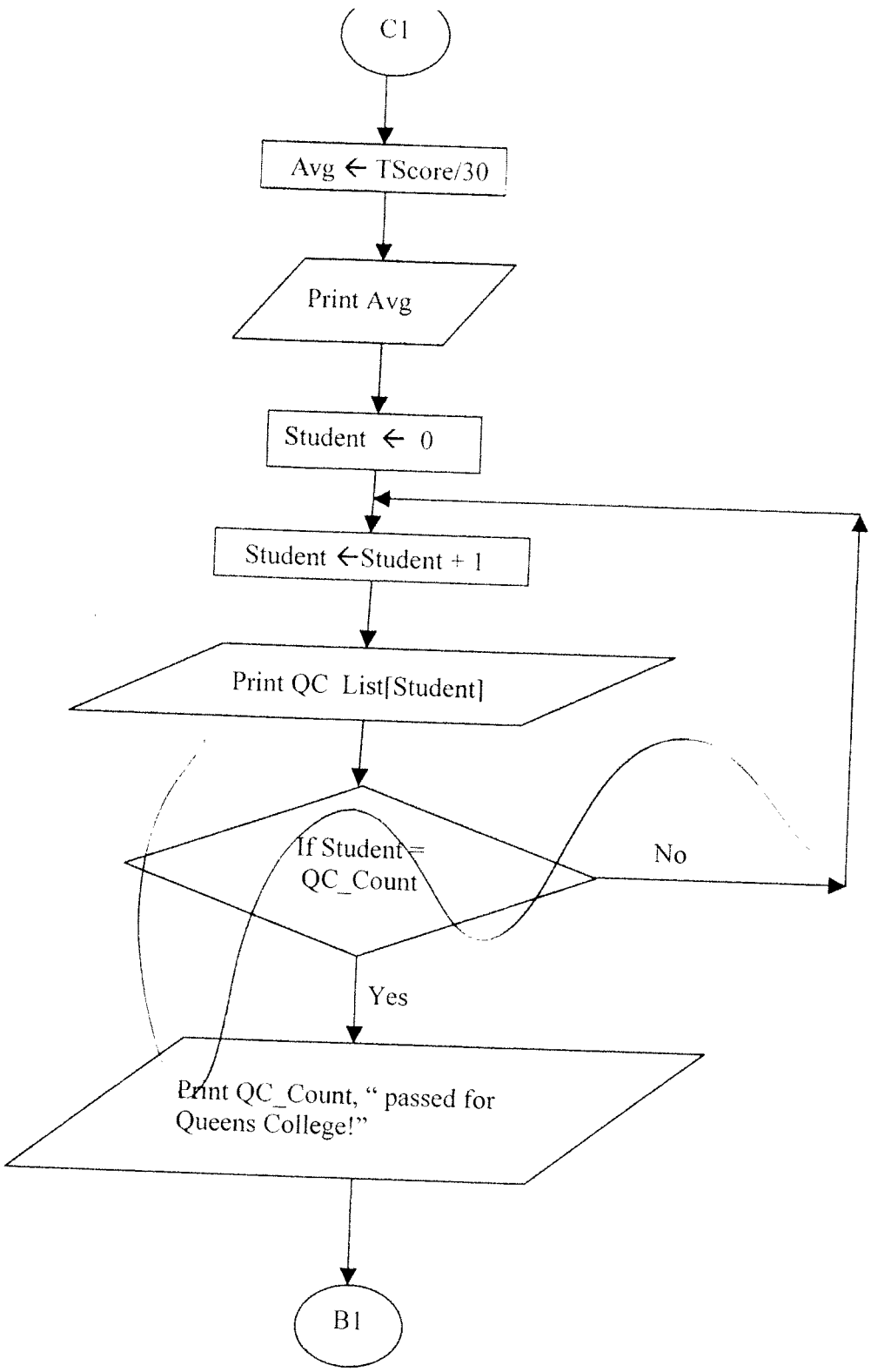
Flowchart

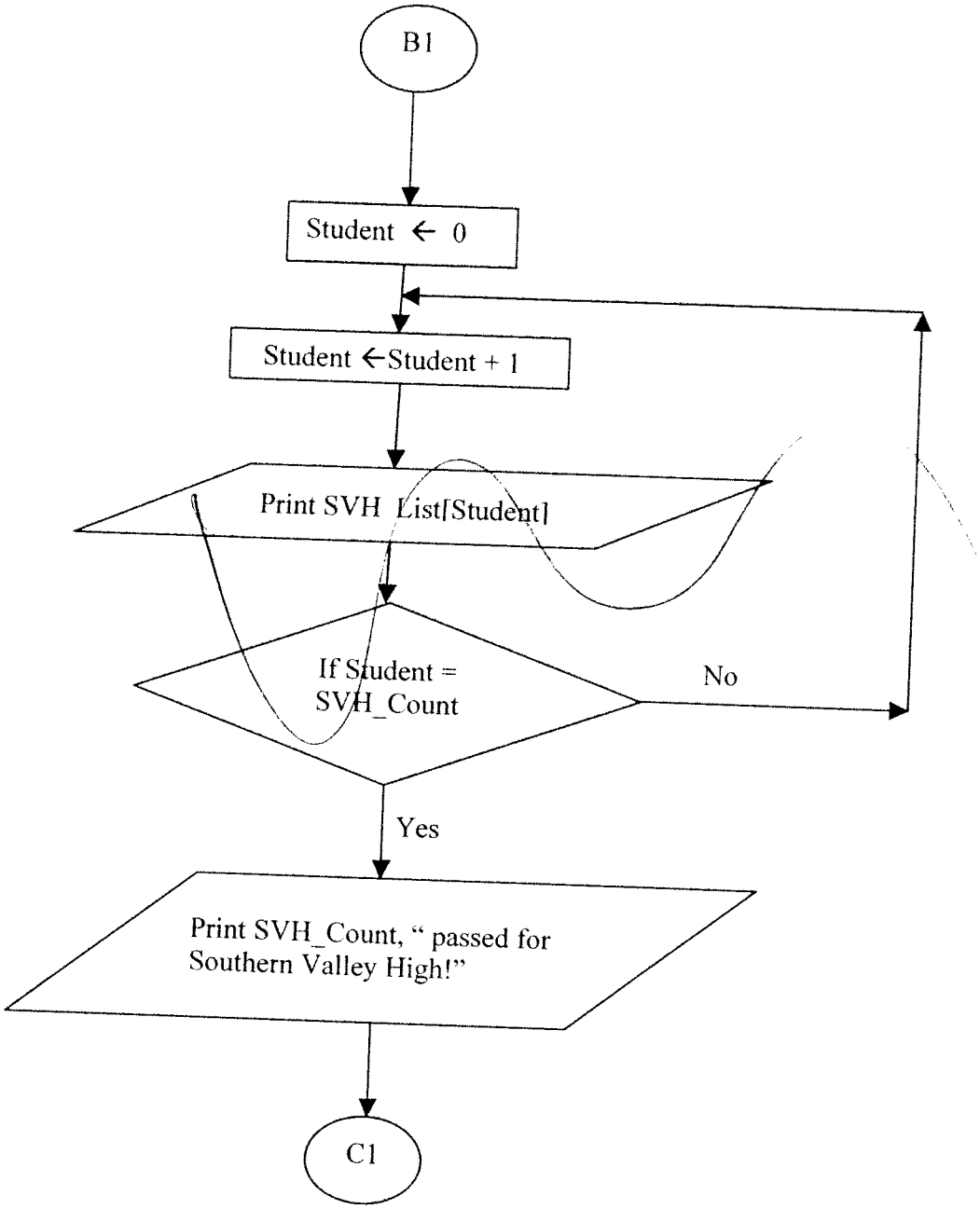

START
QC_Count ← 0
SVH_Count ← 0
HV_Count ← 0
TScore ← 0
Student ← 0
Initialisation
RE
Print "Enter First Name, Last Name, Primary School, IDNO"
Read Fname, Lname, Pschool, IDNO
Storing in variables
Request of data
Print "Enter Score"
Read Score
If Score <=0 or >=100
Yes
Selection
No
TScore ←TScore + Score
FC
RF
Print "Enter First Choice"
Read Fchoice
P1

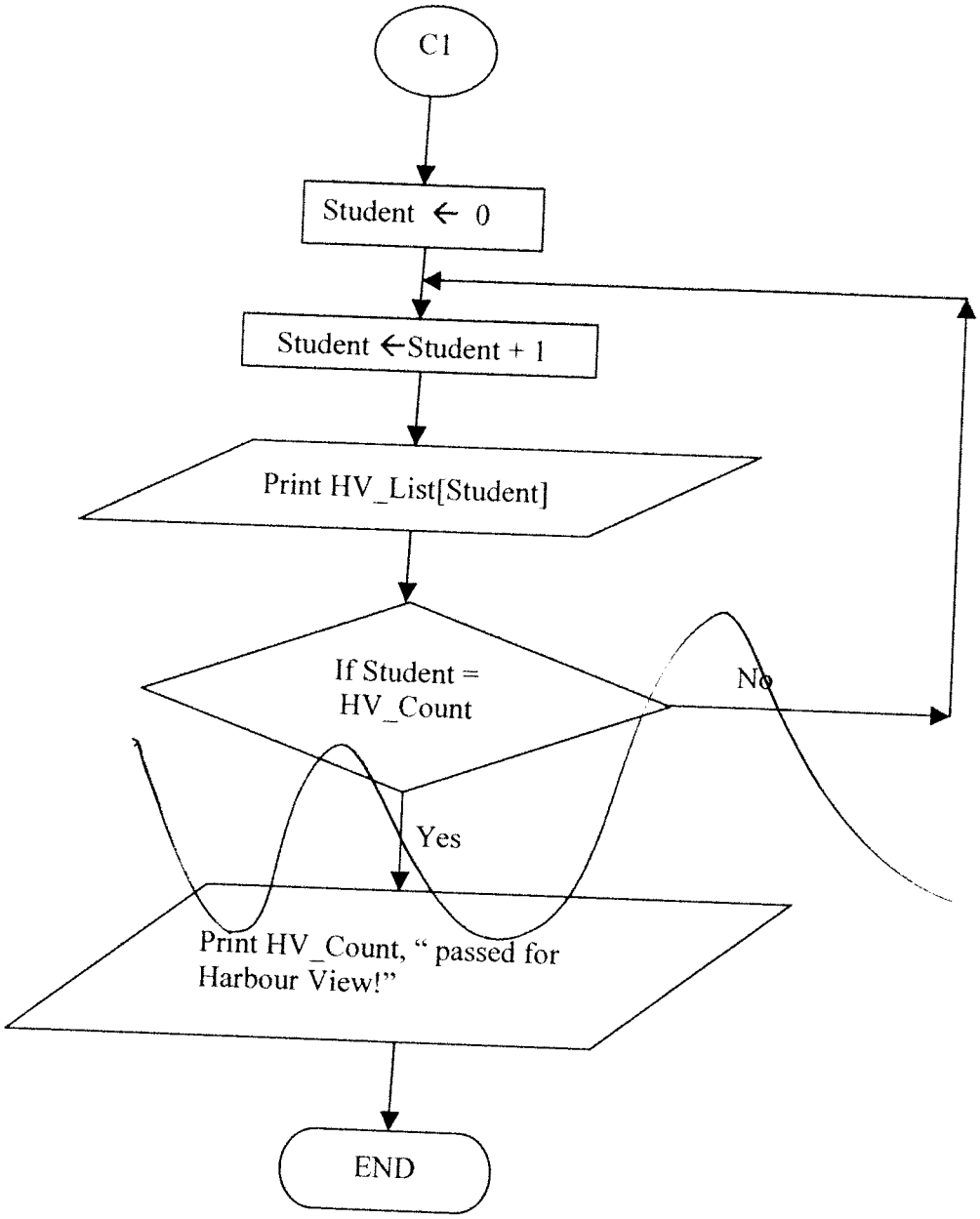

P1
If Fchoice = 'Q' or 'S'
No
FC
Selection
Yes
Print "Enter Second Choice"
Read Schoice
If Schoice = 'S' or 'H'
No
RF
Selection
Yes
Selection
If Score >= 75 and Fchoice = 'Q'
Yes
PQ
No
Selection
If Score >= 62 and (Fchoice or Schoice = 'S')
Yes
PS
No
P2


P2
If Score >= 55
and
Schoice = 'H'
Yes
PH
No
Output
Print Fname, Lname, Pschool, Score
Print 'Unassigned'
CG
PQ
Print Fname, Lname, Pschool, Score
Print 'Queens College'
QC Count ←QC Count + 1
QC_list[QC_Count] ←
IDNO + Fname + Lname
Storing in arrays
CG
PS
Print Fname, Lname, Pschool, Score
Print 'Southern Valley High'
SVH Count ←SVH Count + 1
SVH_list[SVH_Count] ←
IDNO + Fname + Lname
CG


PH
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
No
RE
Yes
C1


C1
Avg ← TScore/30
Print Avg
Student ← 0
Student ←Student + 1
Print QC List[Student]
If Student =
QC_Count
No
Yes
Print QC_Count, " passed for
Queens College!"
B1


B1
Student ← 0
Student ←Student + 1
Print SVH_List[Student]
If Student =
SVH_Count
No
Yes
Print SVH_Count, " passed for
Southern Valley High!"
C1


C1
Student ← 0
Student ←Student + 1
Print HV_List[Student]
If Student =
HV_Count
No
Yes
Print HV_Count, " passed for
Harbour View!"
END

Test Data

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

Appropriate test data.

Some variables identified.

Trace Table

Student	IDNO	Fname	Lname	Score	FC	SC	School Assigned	QC_Count	SVH_Count	HV_Count	TScore	Avg	Print
				0				0	0	0	0		
1	GJ01	John	Gabriel	77.3	Q	H	Queen's College	1	0	0	77.3		
2	GK02	Kain	Godspell	84.6	Q	S	Queen's College	2	0	0	161.9		
3	SL03	Laura	Smith	59.2	S	H	Harbour View	2	0	1	221.1		
4	WA04	Andrea	Williams	60.8	S	H	Harbour View	2	0	2	281.9		
5	PA05	Anita	Paul	89.2	Q	S	Queen's College	3	0	2	371.1		
6	AL06	Lucky	Acer	84.6	Q	S	Queen's College	4	0	2	455.7		
7	CE07	Eledoya	Civelle	52.3	Q	S	Unassigned	4	0	2	508.0		
												72.6	QC = 4 SVH = 0 HV = 2
													72.6

Some changes correctly demonstrated

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

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

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;

loop1 <- 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);

loop2 <- Repeat

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

4 variable types

initialisation of variables

Sequencing of program

```
If (Score <=0) or (Score >=100) Then
  Begin
     Validscore:= 'No';
     Writeln ('Invalid Score!');
  End
  Else
     Validscore:= 'Yes';
  Until Validscore = 'Yes';
  TScore:= TScore + Score;
  Repeat
     Writeln ('Enter The First Choice and Second Choice:');
     readln (Fchoice); readln (Schoice);
  Case Fchoice of
     'Q','S': ValidFc:= 'Yes';

  Else
  Begin
     ValidFc:='No';
     Writeln ('Invalid First Choice!');
  End
  End;

  Case Schoice of
     'S','H': ValidSc:= 'Yes';

  Else
  Begin
     ValidSc:='No';
     Writeln ('Invalid Second Choice!');

  End
  End;
  Until (ValidFc='Yes') and (ValidSc='Yes');


If (Score >= 75) and (Fchoice = 'Q') Then
```

```
Begin

   Writeln(Fname,' ',Lname,' ',Pschool,' ',Score);
   Writeln ('Congratulations, You Are Accepted To Queens College!');
    QC_Count:= QC_Count +1;
    QC_list[QC_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!');
```

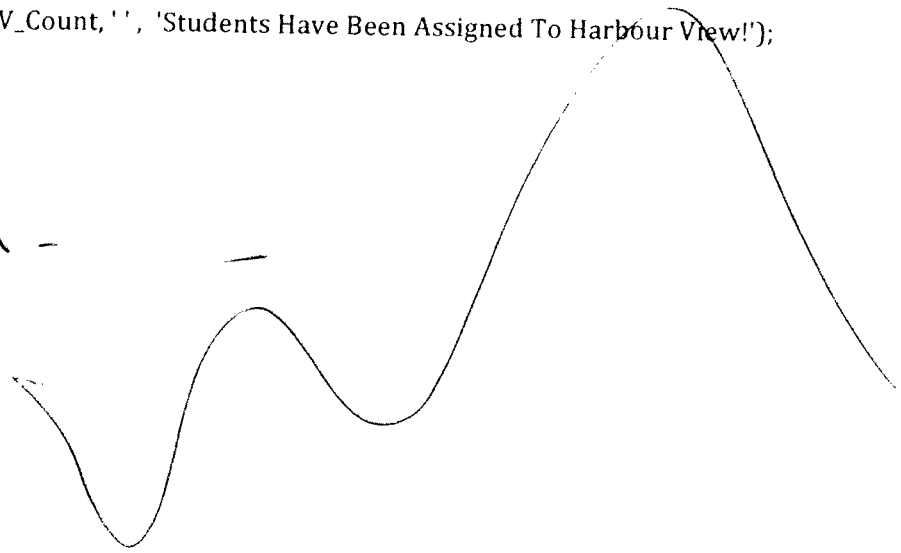
Nested IFs

loop entering data into an array

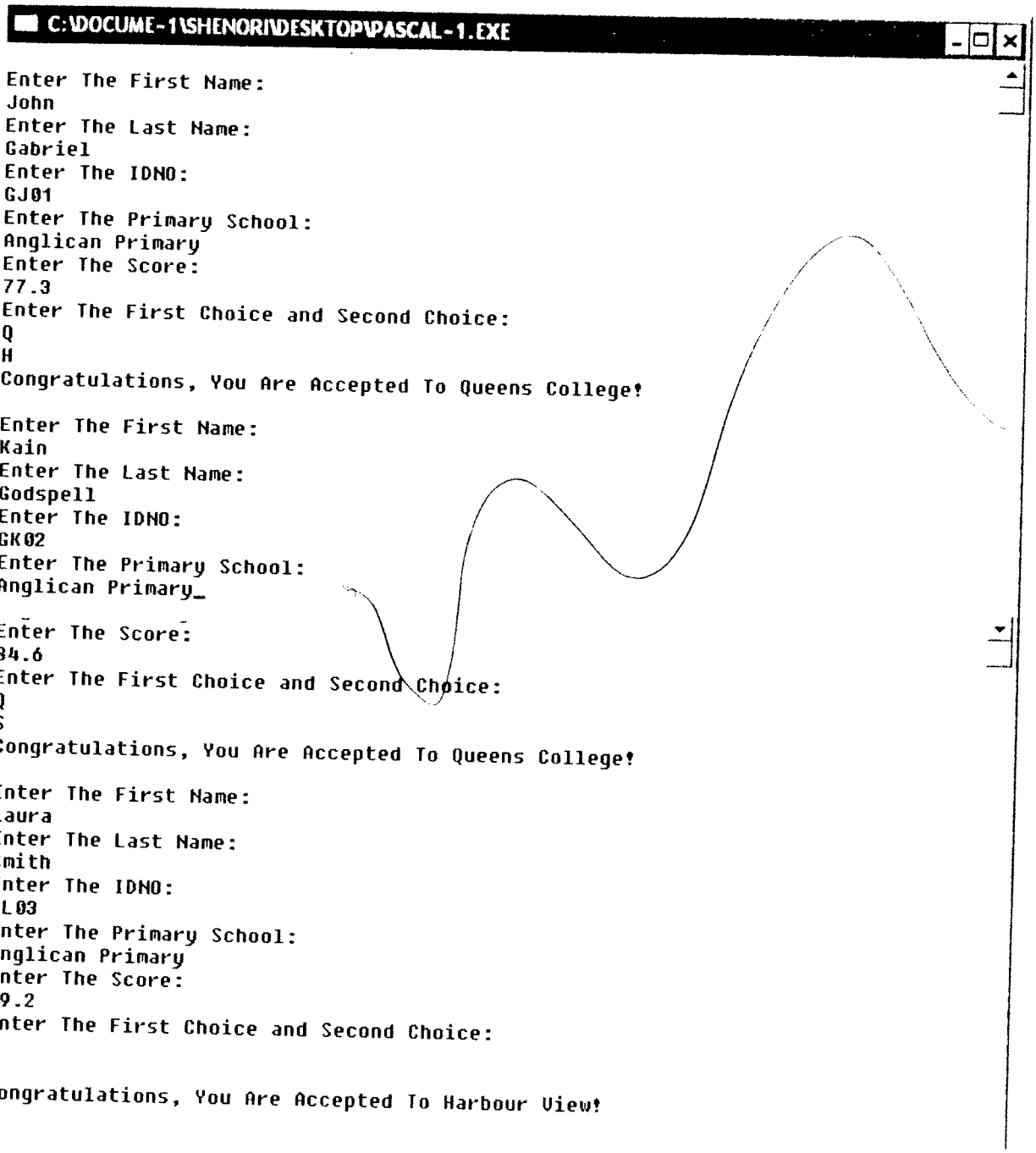
manipulating an array

```
    Writeln;
For Student:= 1 to SVH_Count do
    Writeln (SVH_list [Student]);

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

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

    Writeln;
    Writeln (HV_Count, ' ', 'Students Have Been Assigned To Harbour View!');

End.
```

Indentation –



Results



Program compiled —

Output correct for most values —

User-friendly —

```
C:\DOCUME~1\SHENORI\DESKTOP\PASCAL~1.EXE
Enter The First Name:
Andrea
Enter The Last Name:
Williams
Enter The IDNO:
WA04
Enter The Primary School:
Anglican Primary
Enter The Score:
60.8
Enter The First Choice and Second Choice:
S
H
Congratulations, You Are Accepted To Harbour View!

Enter The First Name:
Anitta
Enter The Last Name:
Paul
Enter The IDNO:
PA05
Enter The Primary School:
Anglican Primary
Enter The Score:
89.2
Enter The First Choice and Second Choice:
Q
S
Congratulations, You Are Accepted To Queens College!

Enter The First Name:
Lucky
Enter The Last Name:
Acer
Enter The IDNO:
AL06
Enter The Primary School:
Anglican Primary
Enter The Score:
101
Invalid Score!
Enter The Score:
84.6
Enter The First Choice and Second Choice:
Q
Q
Invalid Second Choice!
Enter The First Choice and Second Choice:
Q
S
Congratulations, You Are Accepted To Queens College!
```

```
Enter The First Name:
Eledoya
Enter The Last Name:
Civelle
Enter The IDNO:
CE07
Enter The Primary School:
Anglican Primary
Enter The Score:
52.3
Enter The First Choice and Second Choice:
H
O
Invalid First Choice!
Invalid Second Choice!
Enter The First Choice and 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!
```

CSEC Information Technology

SBA Exemplar – June 2011

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.

Algorithmn

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.

CSEC Information Technology

SBA Exemplar – June 2011

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.