Aga Khan University Examination Board

Notes from E-Marking Centre on SSC-II Computer Science Examination May 2017

Introduction

This document has been produced for the teachers and candidates of Secondary School Certificate (SSC-II) Computer Science. It contains comments on candidates' responses to the 2017 SSC-II Examination, indicating the quality of the responses and highlighting their relative strengths and weaknesses.

E-Marking Notes

This includes overall comments on candidates' performance on every question and *some* specific examples of candidates' responses which support the mentioned comments. Please note that the descriptive comments represent an overall perception of the better and weaker responses as gathered from the e-marking session. However, the candidates' responses shared in this document represent some specific example(s) of the mentioned comments.

Teachers and candidates should be aware that examiners may ask questions that address the Student Learning Outcomes (SLOs) in a manner that requires candidates to respond by integrating knowledge, understanding and application skills they have developed during the course of study. Candidates are advised to read and comprehend each question carefully before writing the response to fulfill the demand of the question.

Candidates need to be aware that the marks allocated to the questions are related to the answer space provided on the examination paper as a guide to the length of the required response. A longer response will not in itself lead to higher marks. Candidates need to be familiar with the command words in the SLOs which contain terms commonly used in examination questions. However, candidates should also be aware that not all questions will start with or contain one of the command words. Words such as 'how', 'why' or 'what' may also be used.

General Comments

In general, questions related to flowchart, operators in GW-BASIC, rewriting given code using WHILE…WEND loop, one-dimensional array, writing output for given code and completion of truth table were well attempted. However, questions based on two dimensional array measures to avoid hacking, calculating maximum value in GW-BASIC and decision making on the basis of given criteria in GW-BASIC were generally not well attempted.

Detailed Comments:

Constructed Response Questions (CRQs)

Question 1:

An algorithm is a step by step solution to a given problem.

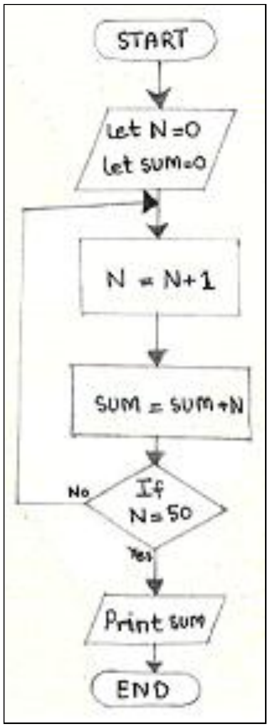
Read the given algorithm and draw a flowchart for it in the given space.

1. Assign the value 0 to the variables SUM and N respectively.
2. Apply an increment on N by 1 and store the new value back in variable N.
3. Add the values of SUM and N. Store the result back in SUM variable.
4. If the value of N is equal to 50 then print the value of SUM otherwise repeat the steps 2, 3 and 4 until the value of N is equal to 50.

(**Note:** Use the same variable names in flowchart that are given in the algorithm.)

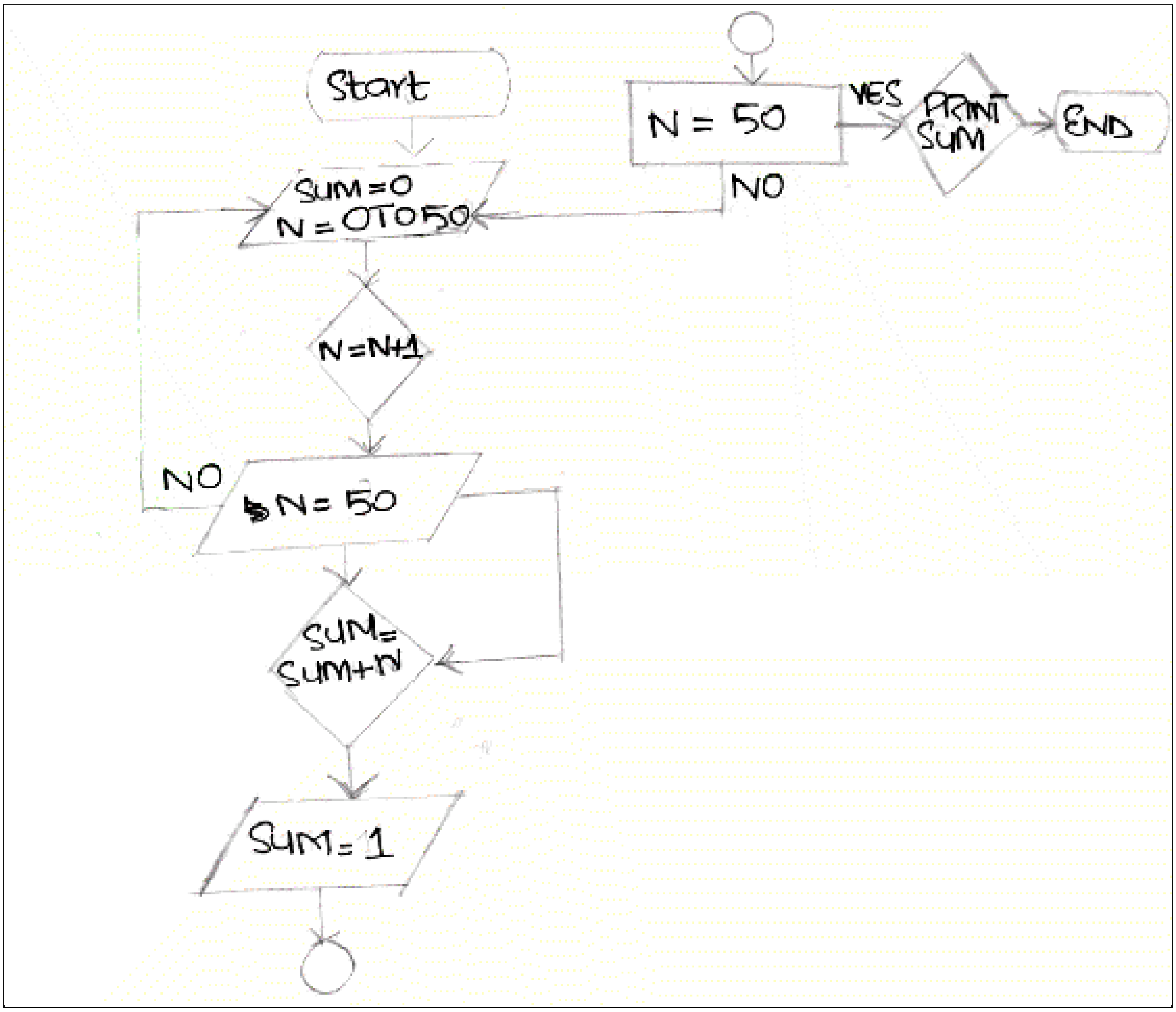
Better responses demonstrated that candidates had clear understanding of purpose of each flowchart symbol and used correct flowchart symbols along with statements for each step of the algorithm given in question. Moreover, these responses connected the flowchart symbols in the correct order.

Example:



Weaker responses depicted lack of understanding of flowchart symbols and did not know clearly that which flowchart symbol should be used to perform the given task. Moreover, these responses were not able to connect the flowchart symbols in the correct sequence. Moreover, such responses did not write appropriate statements inside the flowchart symbols, e.g. they assigned values to variables inside input/ output flowchart symbol rather than processing symbol. Most of these responses were not able to show looping in the flowchart using decision box.

Example:



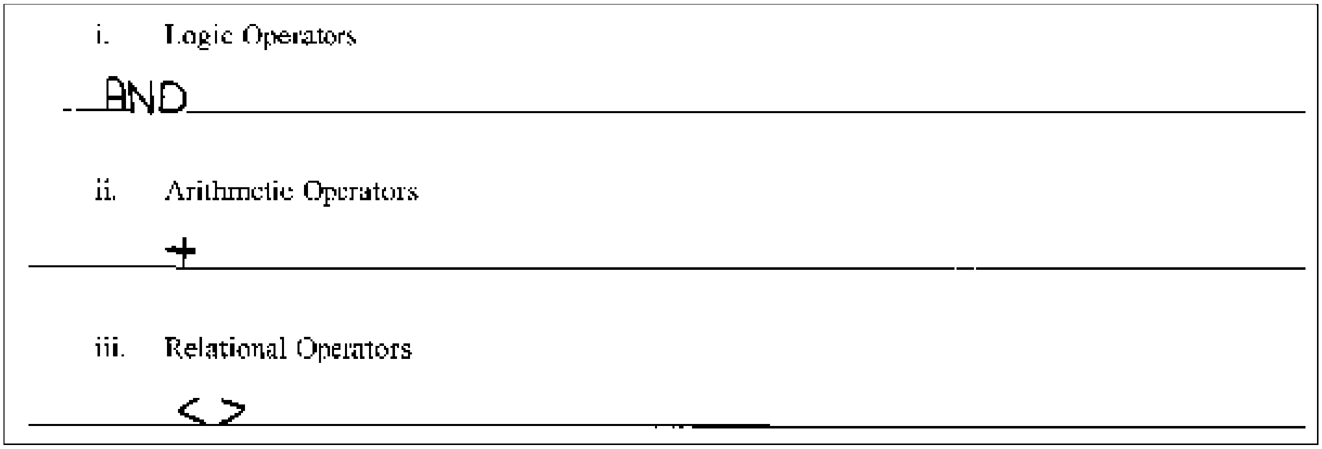
Question 2a:

Write any ONE valid symbol in the given space for each of the stated operators.

i. Logic Operators:
ii. Arithmetic Operators:
iii. Relational Operators:

Better responses showed good knowledge of logic, arithmetic and relational operators and wrote the correct symbols for each type of operator. However, most of the responses wrote two or three correct symbols for each type of operators even though the question asked to write ONE symbol for each type of operators only. This was not required.

Example:



i. Logic Operators

AND

ii. Arithmetic Operators

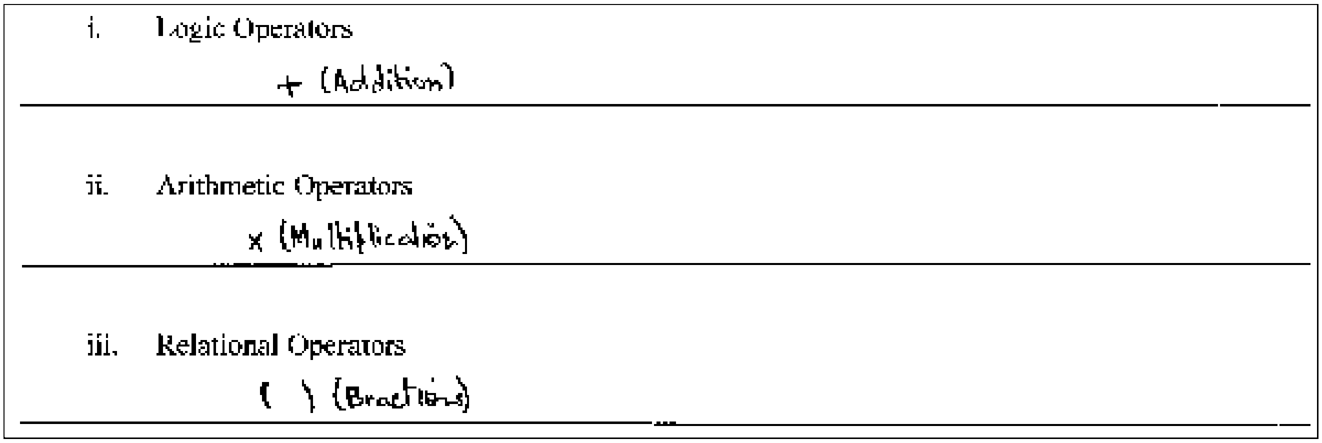
+

iii. Relational Operators

< >

Weaker responses depicted poor understanding of the all three types of operators and wrote purpose of each type of operators/ mathematical operator instead of GW-BASIC operator for example in arithmetic operator, they wrote × instead of *.

Example:



i. Logic Operators

+ (Addition)

ii. Arithmetic Operators

x (Multiplication)

iii. Relational Operators

( ) (Bractions)

Question 2b:

Three GW-BASIC variables are listed below.

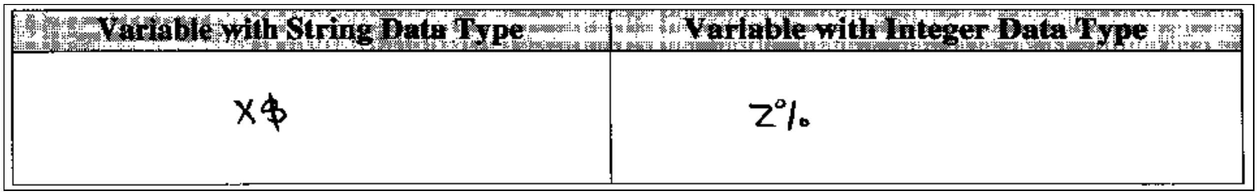
- Z%
- Y#
- X$

Categorise them in the given table.

Variable with String Data Type	Variable with Integer Data Type

Better responses identified X$ as string variable and Z% as integer variable which shows that candidates know the purpose of data type symbols with variable names.

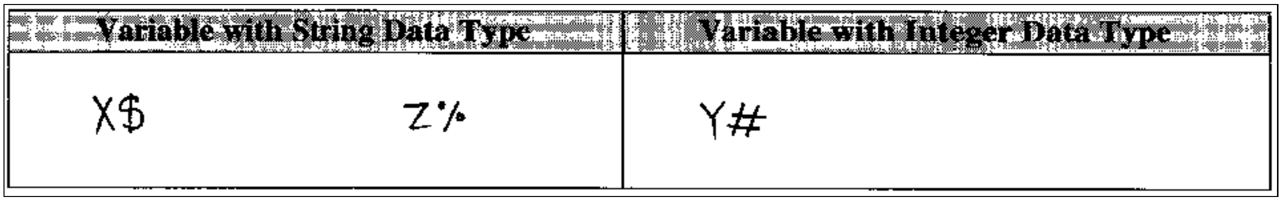
Example:



Variable with String Data Type	Variable with Integer Data Type
X$	Z%

Weaker responses mostly identified string variable correctly but could not identify the integer variable correctly. These responses identified both Y# and Z% as integer variables rather than identifying only Z% as integer variable because Y# is a precision variable.

Example:



Variable with String Data Type	Variable with Integer Data Type
X$ Z%	Y#

Question 3:

Read the GW-BASIC code given below.

```
10      CLS
20      FOR Count = 0 TO 4
30      PRINT "*"
40      NEXT
50      END
```

a. Which of the following outputs, **A** or **B**, is correct for the given GW-BASIC code?

```
*
*
*
*
*
```

A

```
*
*
*
*
```

B

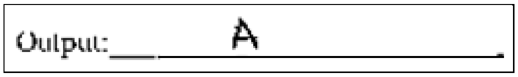
Output: ____________________

b. REWRITE the GW-BASIC code given in this question using WHILE…WEND loop.

Better responses selected A as output of the given program which means candidates had clear understanding that loop will iterate 5 times (starting from 0 and ending at 4). Moreover, these responses converted the given FOR loop program correctly into WHILE…WEND program and placed the programming statements in correct order.

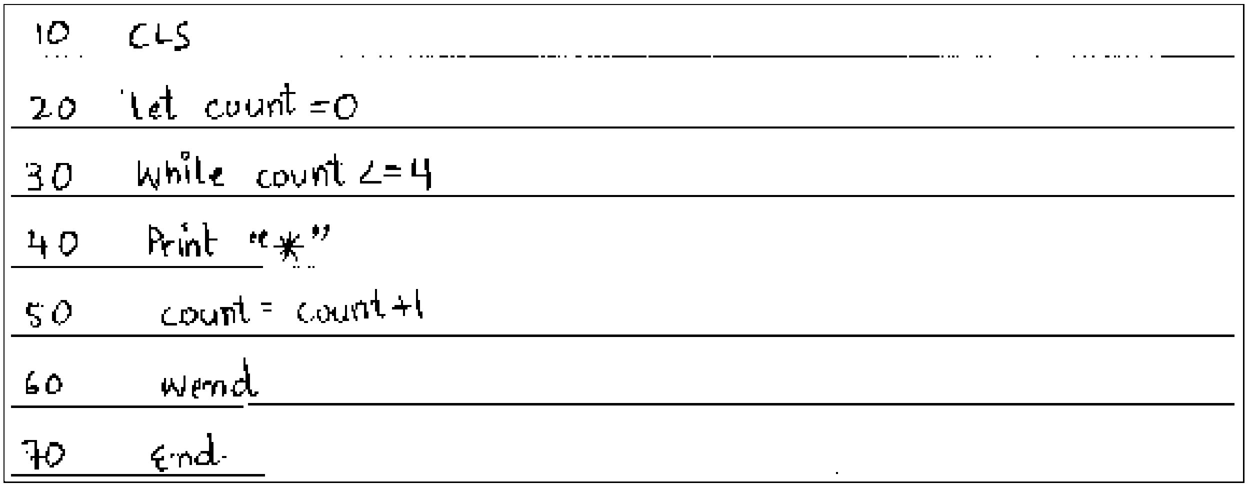
Example:

a.



Output:___A___________________

b.

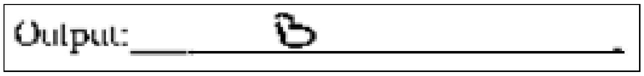


```
10  CLS
20  'Let count =0
30  While count <=4
40  Print "*"
50  count = count+1
60  wend
70  End.
```

Weaker responses selected B as output of the given program which means candidates only focused on the ending value of loop which is 4 and on the basis of this value, the output was chosen. Moreover, these responses could not convert the given FOR loop program correctly into WHILE…WEND program and wrote incorrect loop range/ wrote the same given program/ wrote programming statements in wrong sequence or wrote both WHILE…WEND and FOR loop statements.

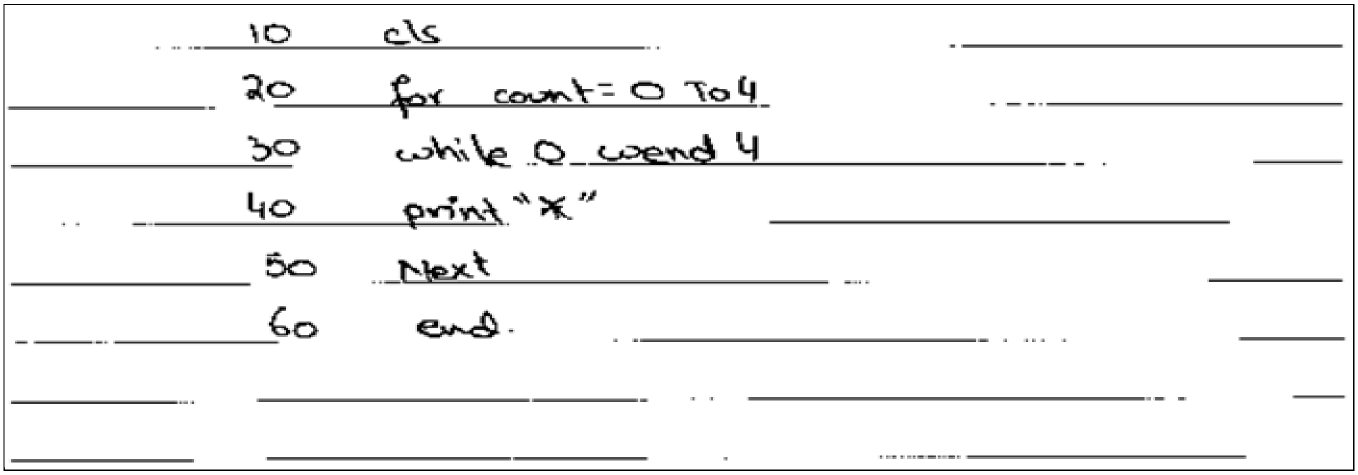
Example:

a.



Output: B

b.



```
10  cls
20  for count= 0 To4
30  while 0 wend 4
40  print "*"
50  Next
60  end.
```

Question 4:

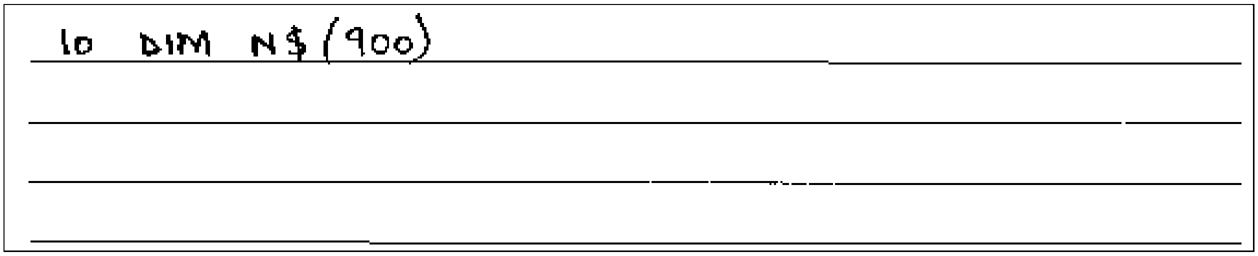
There are 900 students studying in a school. The school wants to record the names and ages of these students. Write GW-BASIC syntax to declare a

a. one-dimensional array with appropriate subscript to store the name of students.
b. two-dimensional array with appropriate subscripts to store the age of students.

Better responses declared arrays with proper name, correct data type and size of array, i.e. array size is 900/ 899 for one-dimensional array and array size is (30, 30)/ (150, 6)/ (15, 60)/ (90, 10)/ (36, 25) in case of two-dimensional array.

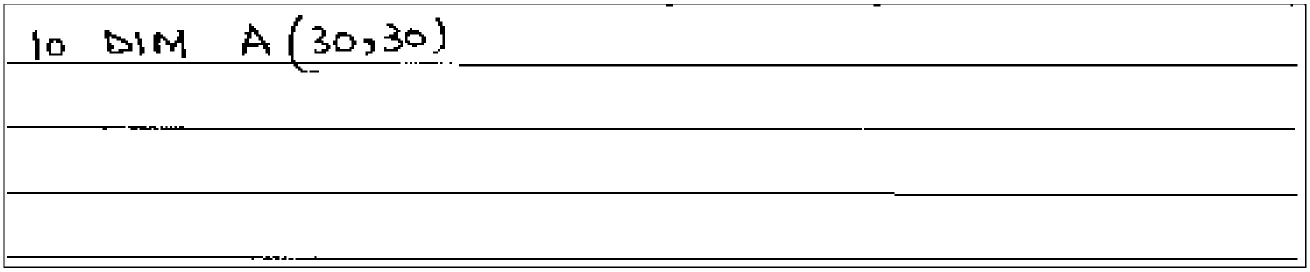
Example:

a.



```
10 DIM N$(900)
```

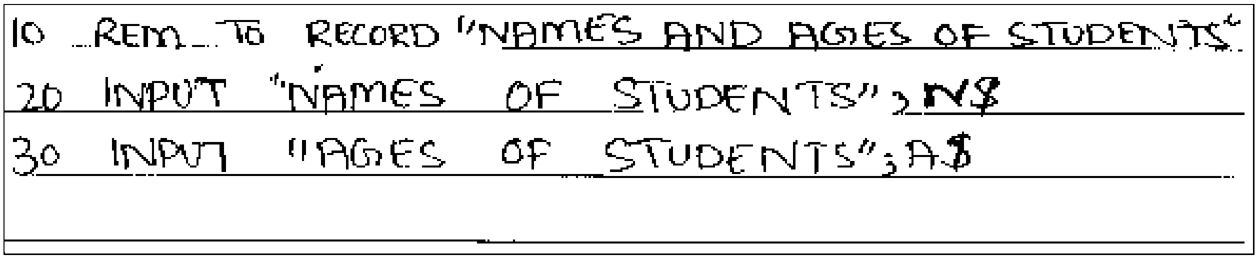
b.



```
10 DIM A(30,30)
```

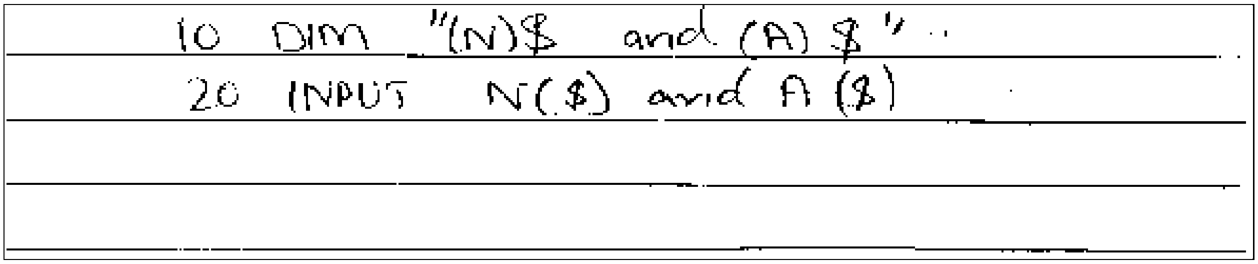
Weaker responses did not specify the correct data type, name and size of array. In most of these responses, name of array was not mentioned and, instead of defining size of an array by using numeric value (subscript), candidates wrote the textual data which is not correct.

Example 4a:



```
10 REM TO RECORD "NAMES AND AGES OF STUDENTS"
20 INPUT "NAMES OF STUDENTS";N$
30 INPUT "AGES OF STUDENTS";A$
```

b.



10 DIM "(N)$ and (A) $"
20 INPUT N($) and A ($)

Question 5:

Read the GW-BASIC codes given in boxes and write the output of each.

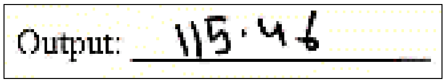
```
10      CLS
20      LET H = − 115.46
30      PRINT ABS(H)
40      END
```

```
10      CLS
20      LET G = 25
30      PRINT SQR(G)
40      END
```

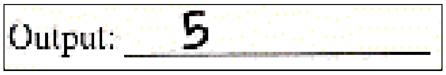
```
10      CLS
20      LET N$ = “Kentucky”
30      PRINT  LEFT$(N$,4)
40      END
```

Better responses showed the good understanding of the built-in functions ‘ABS’, ‘SQR’ and ‘Left$’ in GW-BASIC by writing the correct output.

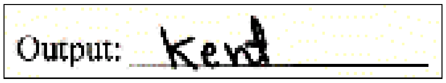
Example



Output: 115·46



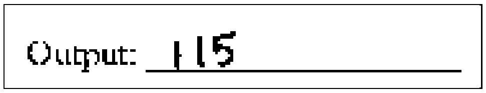
Output: 5



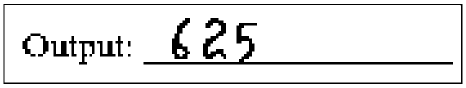
Output: Kent

Weaker responses demonstrated lack of understanding of the built-in function in GW-BASIC and wrote the incorrect output. Most of these responses mentioned that ABS function is used for rounding off a number or removing the part after the decimal point. Moreover, these responses depicted the confusion regarding ‘SQR’ function and wrote square of 25 instead of writing the value of square root. Most of these responses knew the purpose of ‘Left$’ function, however, there were few which interpreted this function in the opposite way and wrote the four right characters of word Kentucky in output rather than writing four left characters.

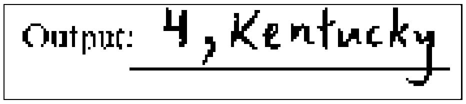
Example:



Output: 115



Output: 625



Output: 4, Kentucky

Question 6:

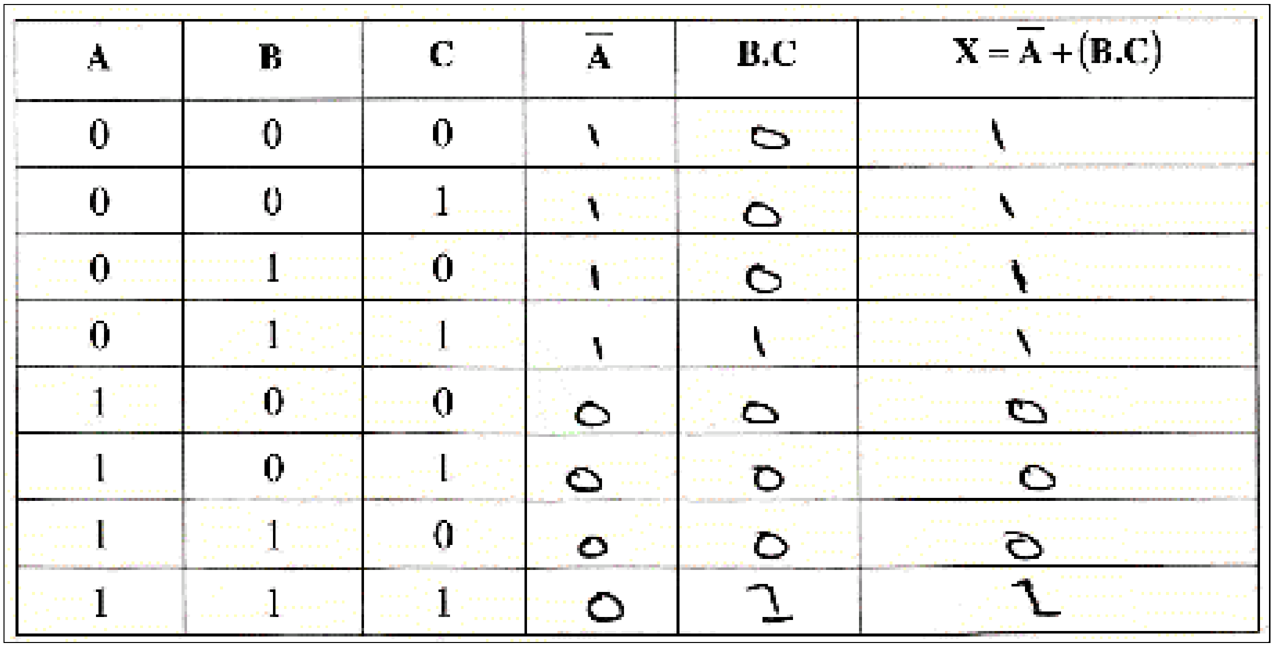
Complete the truth table for the given Boolean expression.

$$\mathbf{X} = \overline{\mathbf{A}} + (\mathbf{B.C})$$

A	**B**	**C**	$\overline{\mathbf{A}}$	**B.C**	$\mathbf{X} = \overline{\mathbf{A}} + (\mathbf{B.C})$
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Better responses demonstrated very good understanding of the Boolean operations (complement, AND, OR), performed them and completed the table correctly.

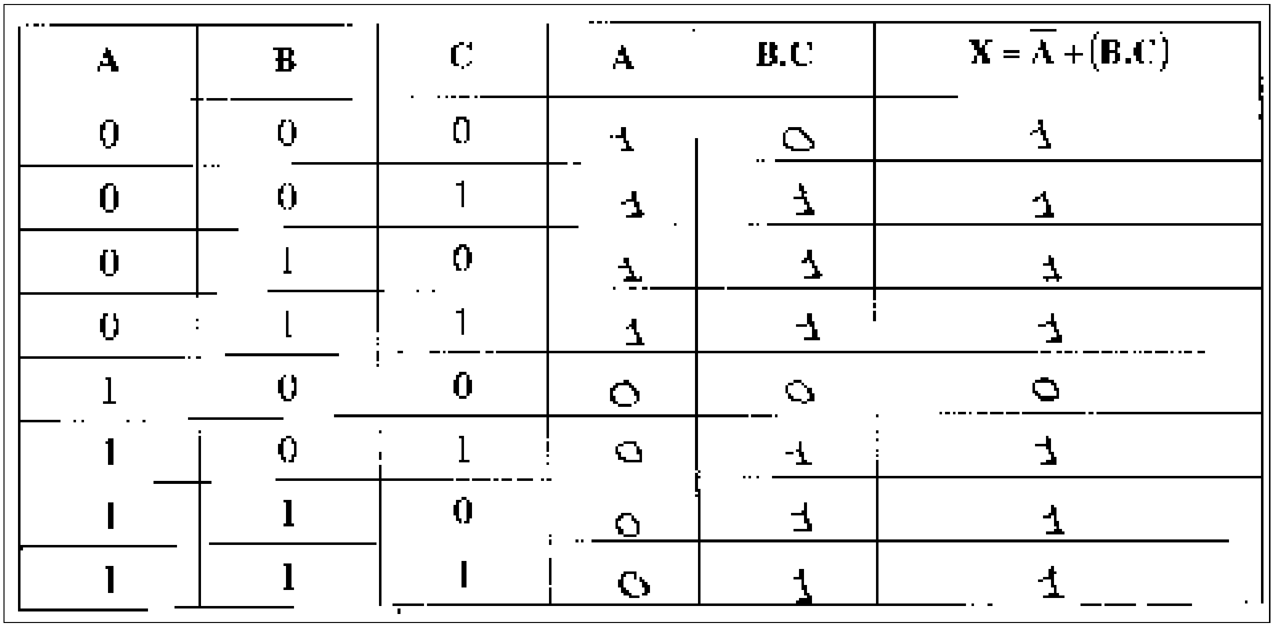
Example:



A	B	C	$\overline{A}$	B.C	$X = \overline{A} + (B.C)$
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	1	1

Weaker responses were mostly able to calculate the complement of **A** but such responses could not perform the AND/ OR operations.

Example:



A	B	C	A	B.C	$X = \overline{A} + (B.C)$
0	0	0	1	0	1
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	0	1	1

Extended Response Questions (ERQs)

The following questions offered a choice between part **a** and **b**.

Question 7a:

i. Define the term virus with reference to computers.

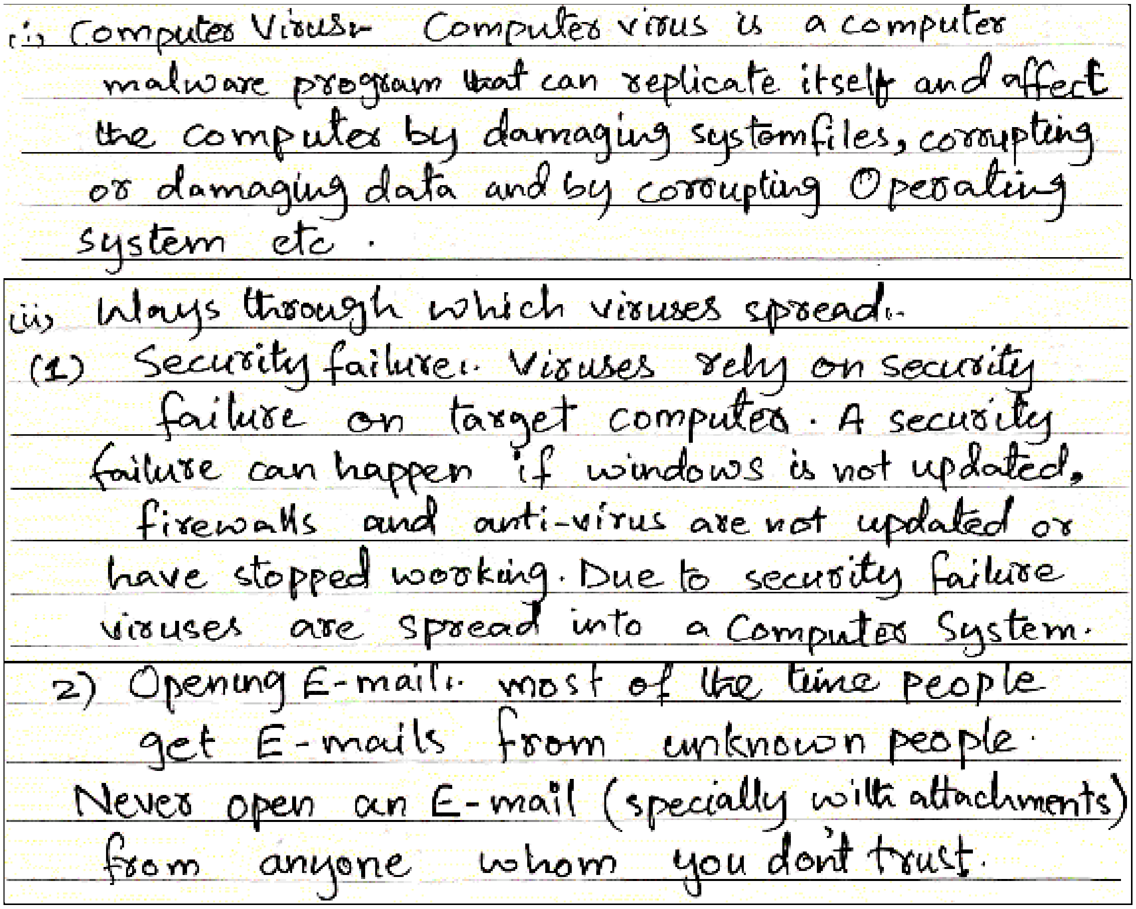
ii. Describe any THREE ways through which viruses can spread in a computer system.

iii. Write any FOUR appropriate steps that can be taken to protect a computer system against viruses.

(Note: Most of the candidates opted part **a** of this question and it was clear after assessment that most of the candidates performed well in this part.)

Better responses defined the computer virus correctly and described the ways to spread viruses such as downloading and opening infected attachment in an email will cause virus to spread in computer system; downloading an infected file from the internet and running it will spread virus in computer system; viruses can be spread from infected floppy disk/ CD/ DVD inserted in a computer system; connecting a flash memory stick and opening infected files stored in flash memory stick will cause virus to spread in computer system; viruses can also spread in a computer system via computer networks. Also, these responses wrote correct steps to be taken to protect a computer system such as keep your anti-virus software up to date; don't click on links within emails; use a firewall; explore USB drive via new windows option; do not double click and explore USB drive directly; scan a USB drive before exploring it in your computer system.

Example:

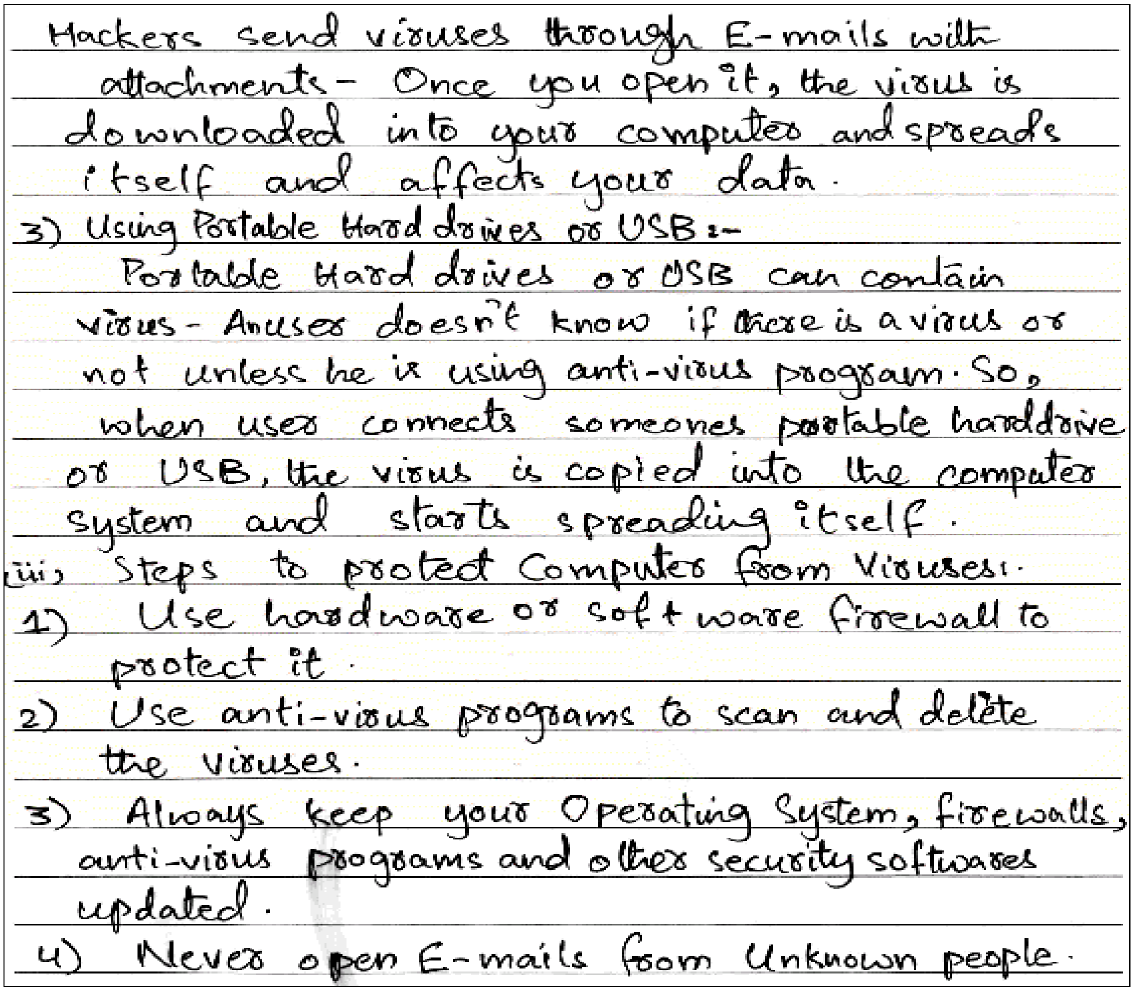


(i) Computer Virus:- Computer virus is a computer malware program that can replicate itself and affect the computer by damaging systemfiles, corrupting or damaging data and by corroupting Operating system etc.

(ii) Ways through which viruses spread:-

(1) Security failure:- Viruses rely on security failure on target computer. A security failure can happen if windows is not updated, firewalls and anti-virus are not updated or have stopped working. Due to security failure viruses are spread into a Computer System.

2) Opening E-mail:- most of the time people get E-mails from unknown people. Never open an E-mail (specially with attachments) from anyone whom you don't trust.



Hackers send viruses through E-mails with attachments- Once you open it, the virus is downloaded into your computer and spreads itself and affects your data.

3) Using Portable Hard drives or USB :-
Portable Hard drives or USB can contain virus- Anuser doesn't know if there is a virus or not unless he is using anti-virus program. So, when user connects someones portable harddrive or USB, the virus is copied into the computer system and starts spreading itself.

(iii) Steps to protect Computer from Viruses:-

1) Use hardware or software firewall to protect it.

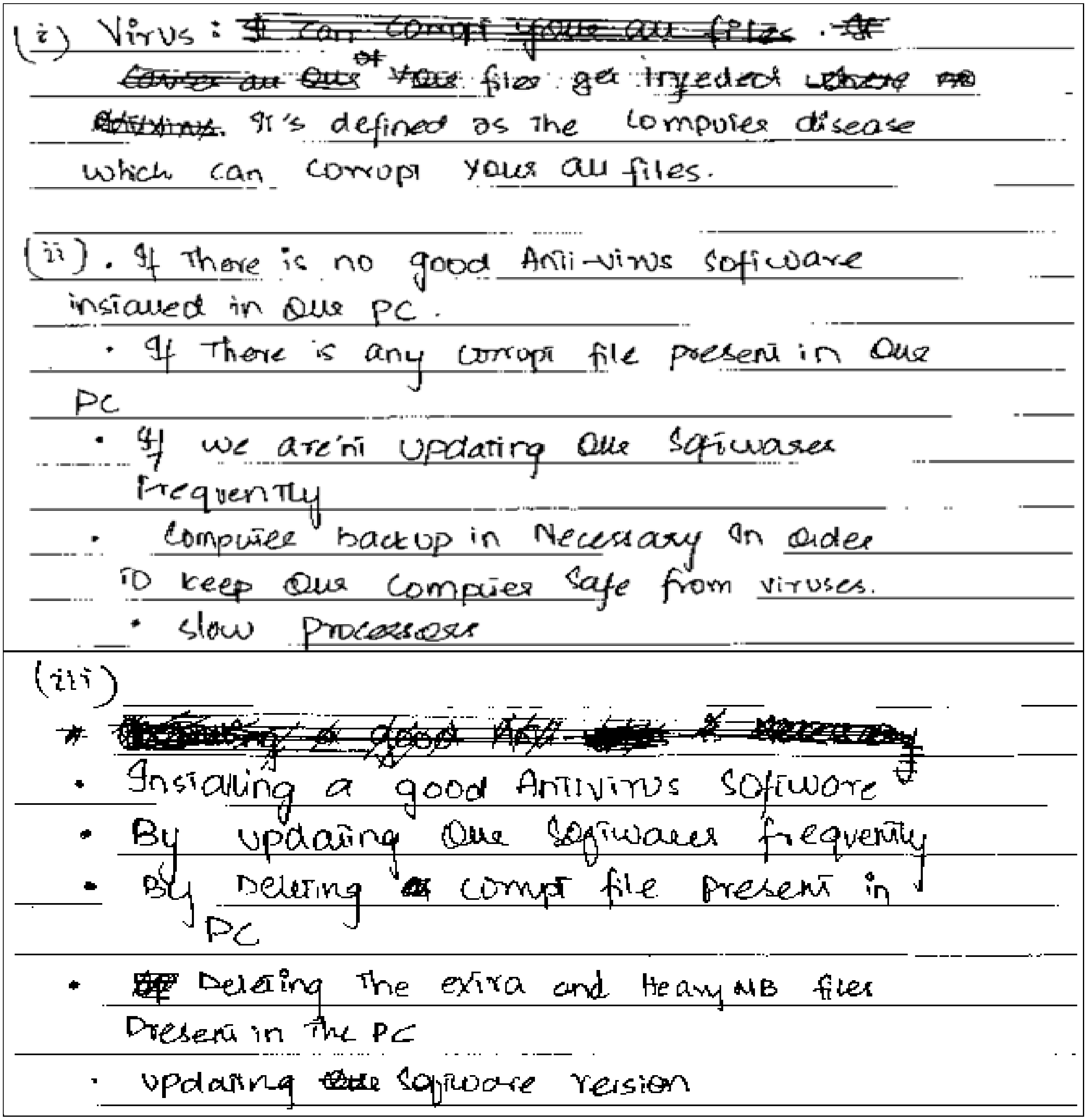
2) Use anti-virus programs to scan and delete the viruses.

3) Always keep your Operating System, firewalls, anti-virus programs and other security softwares updated.

4) Never open E-mails from Unknown people.

Weaker responses were not able to define and explain the ways through which viruses spread and, instead of that, types of viruses are explained. Moreover, these responses were not able to write unique steps to protect computer system against viruses and wrote duplicate steps such as install anti-virus software; download and install virus free programs, etc.

Example:



(i) Virus : ~~It can corropt your all files~~ . ~~If~~ ~~corrupt all our~~ of ~~your~~ files get infected ~~where no~~ ~~Antivirus.~~ It's defined as the computer disease which can corropt your all files.

(ii) . If there is no good Anti-virus software installed in Our PC .

- If there is any corropt file present in Our PC
- If we aren't updating Our Softwarer frequently
- Computer backup in Necessary In Order To keep Our Computer Safe from viruses.
- slow Processor

(iii)

* ~~Installing a good Anti-virus is necessary~~

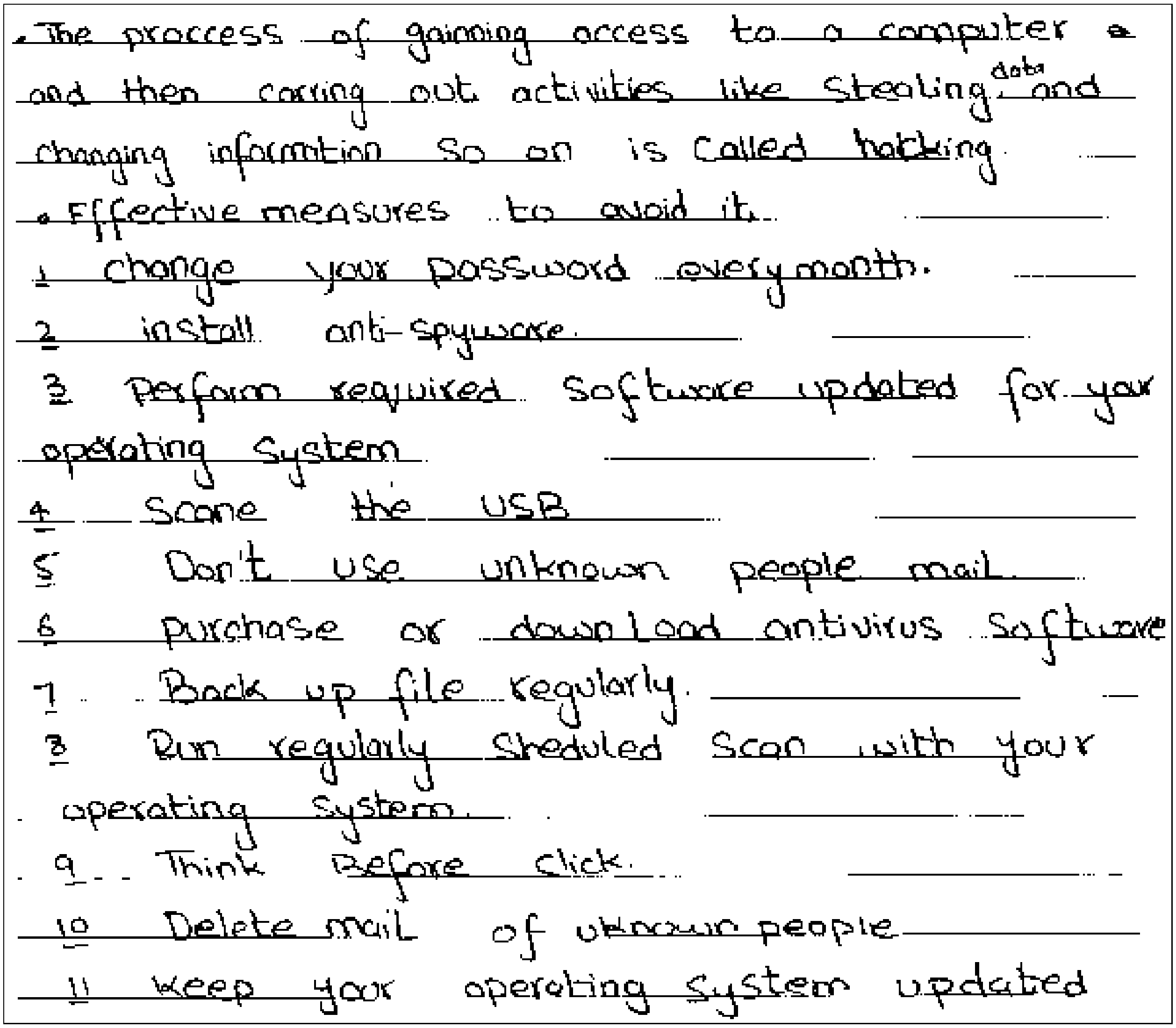
- Installing a good Antivirus Software
- By updating Our Softwarer frequently
- By Deleting corrupt file present in PC
- ~~By~~ Deleting the extra and Heavy MB files Present in the PC
- updating ~~our~~ Software version

Question 7b:

Define the term hacking and suggest any SEVEN effective measures to avoid it.

Better responses showed good understanding about hacking and measures to avoid it. These responses suggested such as change your password periodically; install firewall on computer system; update operating system frequently; install anti-spyware; adware program; do not login your social media or bank accounts from public computers, etc.

Example:



. The proccess of gainning access to a computer ~~a~~ and then carring out activities like Stealing data and changing information So on is Called hacking.

. Effective measures to avoid it.

1 change your password every month.

2 install anti-spyware.

3 Perform required Software updated for your operating System

4 Scane the USB

5 Don't use unknown people mail.

6 purchase or down Load antivirus Software

7 Back up file regularly.

8 Run regularly Sheduled Scan with your operating System.

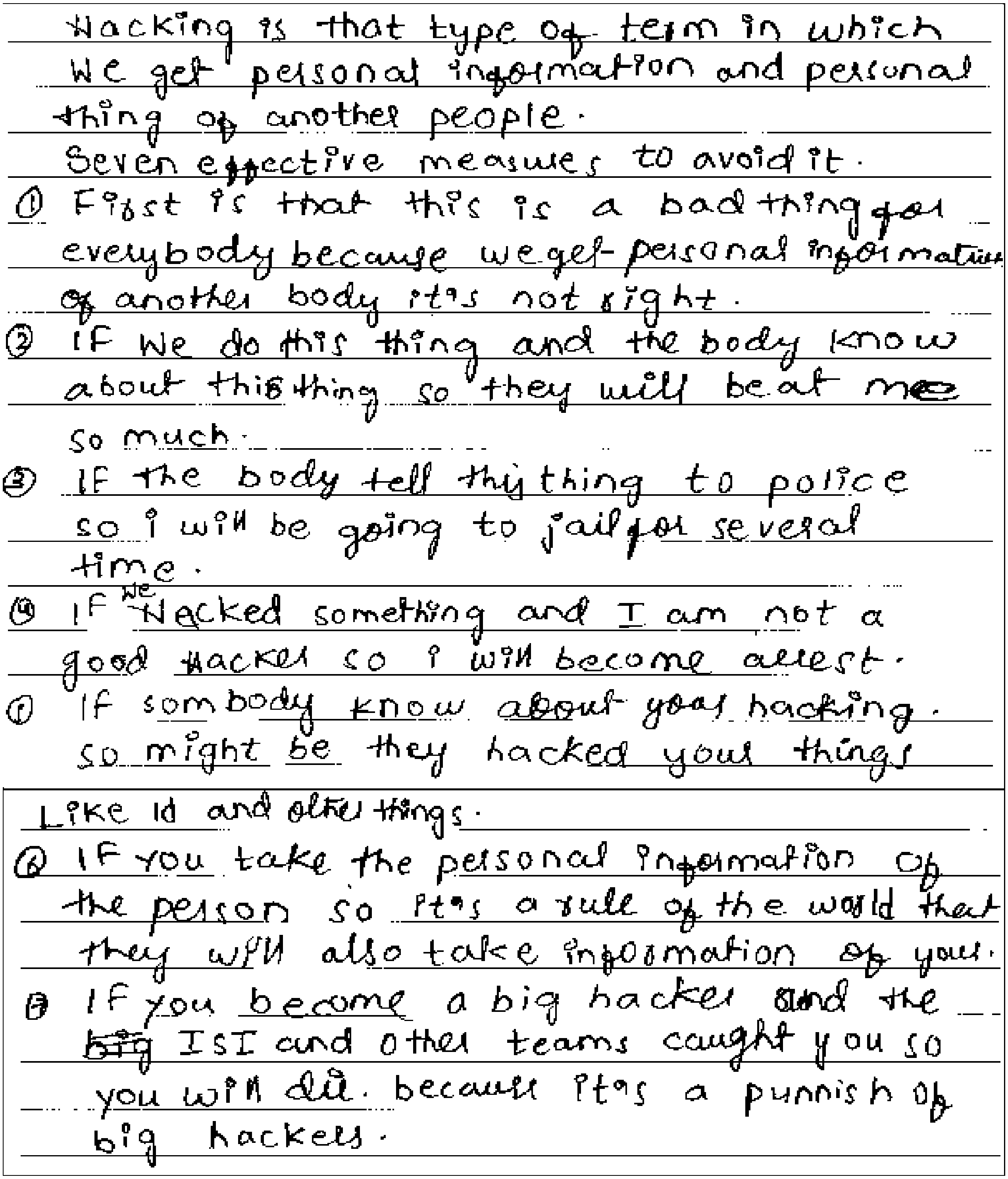
9 Think Before click.

10 Delete mail of uknown people

11 keep your operating System updated

Weaker responses depicted that candidates understood the question but were not able to write to the point answer in technical language. They mostly wrote general things about hacking such as hacking is bad thing; hacking may cause one to go police station; by blocking internet websites; by restricting the internet access; scheduled use of internet etc.

Example:



Hacking is that type of term in which We get personal information and personal thing of another people.
Seven effective measures to avoid it.
① First is that this is a bad thing for everybody because we get personal information of another body it's not right.
② If We do this thing and the body know about this thing so they will beat me so much.
③ If the body tell thiy thing to police so i will be going to jail for several time.
④ If We Hacked something and I am not a good Hacker so i will become arrest.
⑤ If sombody know about your hacking. so might be they hacked your things Like Id and other things.
⑥ If you take the personal information of the person so it's a rule of the world that they will also take information of you.
⑦ If you become a big hacker and the ~~big~~ ISI and other teams caught you so you will die. because it's a punnish of big hackers.

Question 8a:

A cricket club trainer wants to calculate the average weight of club members.
Write GW-BASIC code that will

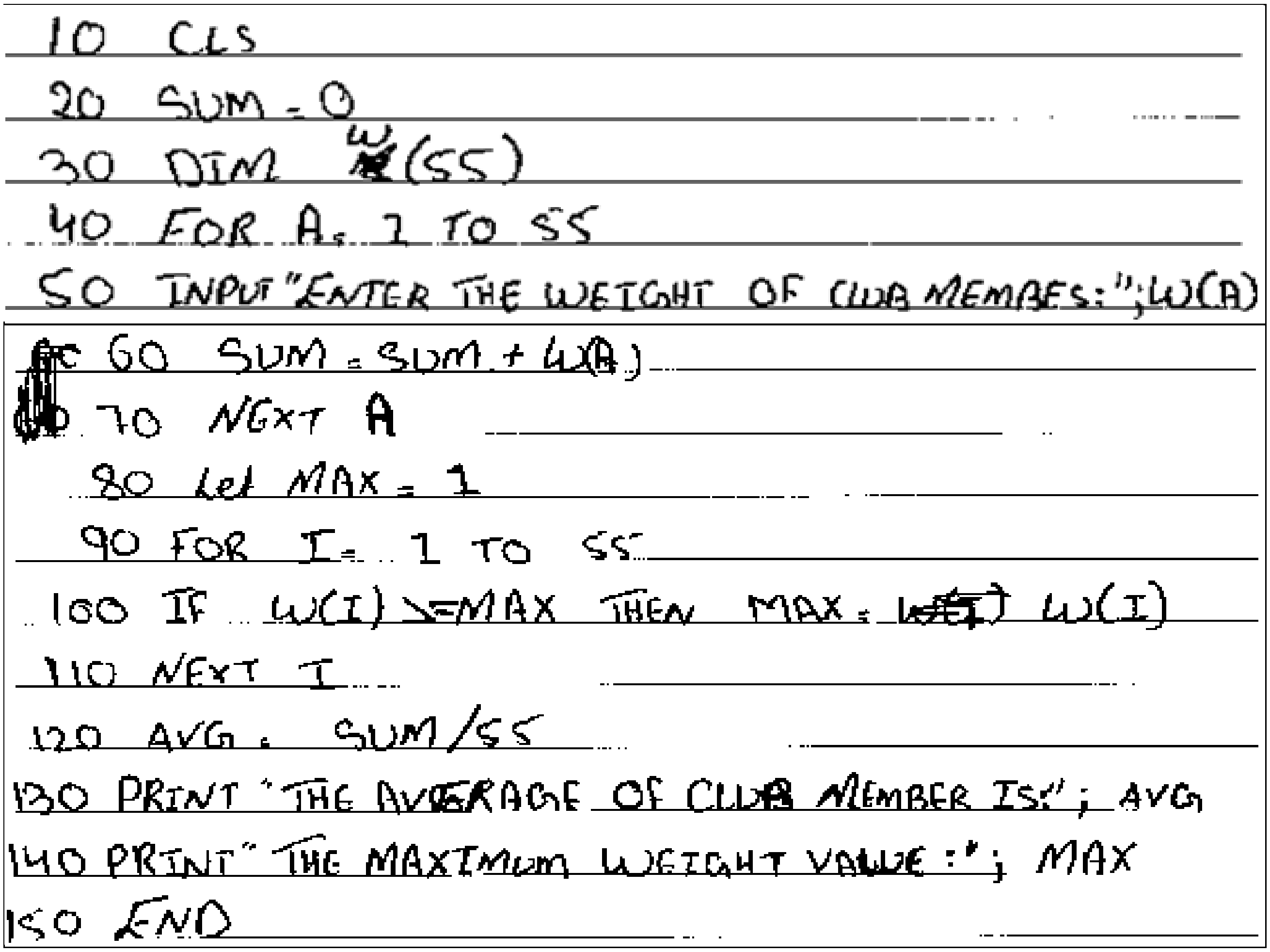
- input the weight of 55 club members.
- calculate the average and maximum weight values.
- display the average and maximum weight values.

(**Note:** Use the For loop for repeating the code.)

(**Note:** Most of the candidates opted part **b** of this question and they performed well.)

Better responses demonstrated the good application of For loop and conditional statement for the given problem. These responses used correct range to start and end the loop. Although this question can easily be solved without using Arrays which needs more expertise but few responses had written the solution of given problem using Arrays.

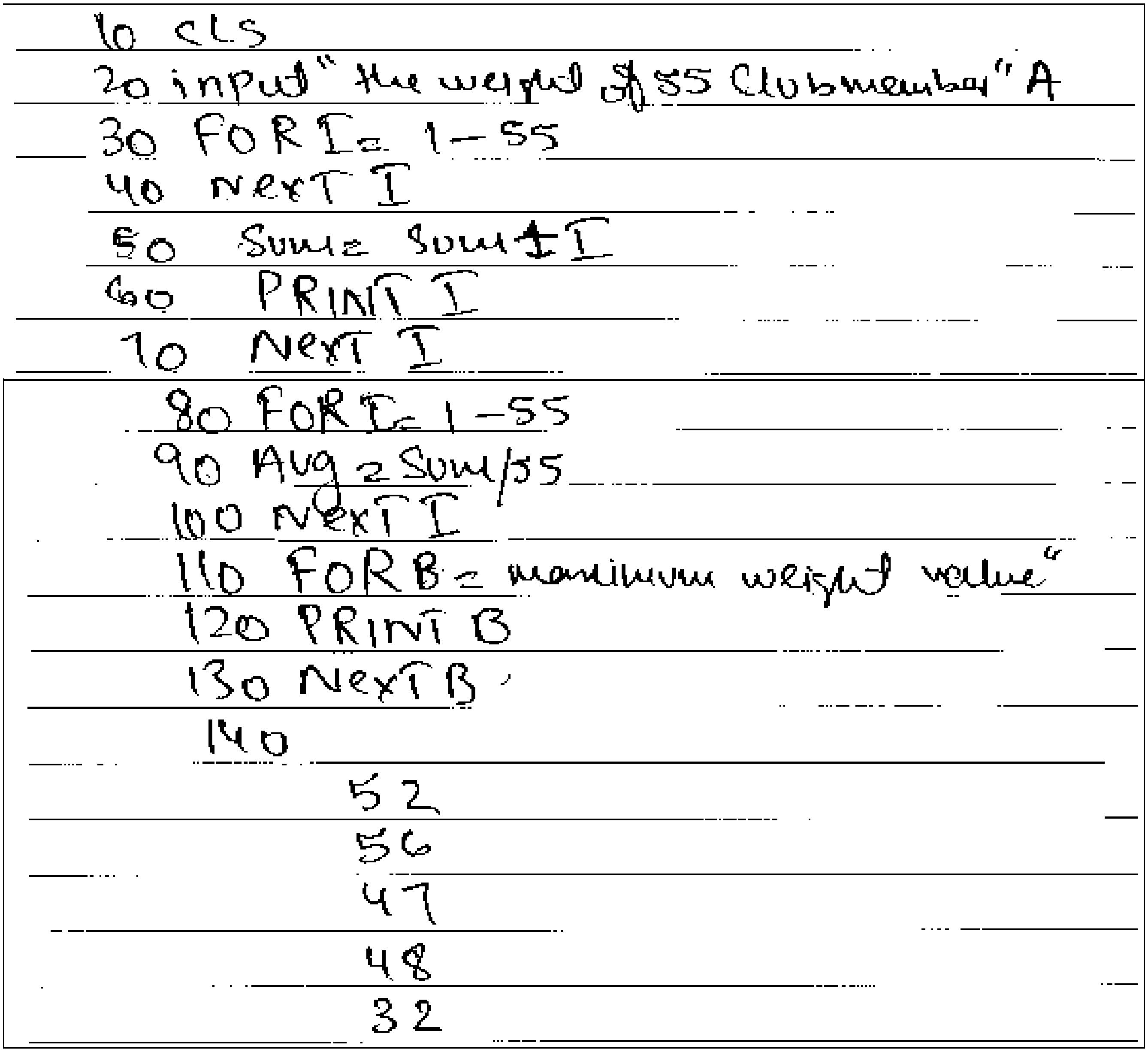
Example:



```
10 CLS
20 SUM = 0
30 DIM W(55)
40 FOR A = 1 TO 55
50 INPUT "ENTER THE WEIGHT OF CLUB MEMBES:";W(A)
60 SUM = SUM + W(A)
70 NEXT A
80 let MAX = 1
90 FOR I = 1 TO 55
100 IF W(I) >=MAX THEN MAX = W(I)
110 NEXT I
120 AVG = SUM/55
130 PRINT "THE AVERAGE OF CLUB MEMBER IS:"; AVG
140 PRINT "THE MAXIMUM WEIGHT VALUE :"; MAX
150 END
```

Weaker responses did not state proper GW-BASIC statements and could not apply the For loop on the given problem. Also, most of these responses depicted lack of ability to build the logic to calculate maximum value from input values in a loop statement.

Example:



```
10 cls
20 input" the weight of 55 Clubmember" A
30 FOR I= 1-55
40 NexT I
50 Sum= Sum+I
60 PRINT I
70 NexT I
80 FOR I= 1-55
90 Avg = Sum/55
100 NexT I
110 FOR B= maximum weight value"
120 PRINT B
130 NexT B
140
52
56
47
48
32
```

Question 8b:

The Body Mass Index (BMI) is a simple weight-for-height (kg/m²) index. It is an important screening tool that tells whether you are obese, overweight, underweight or healthy.

Write a GW-BASIC program that will

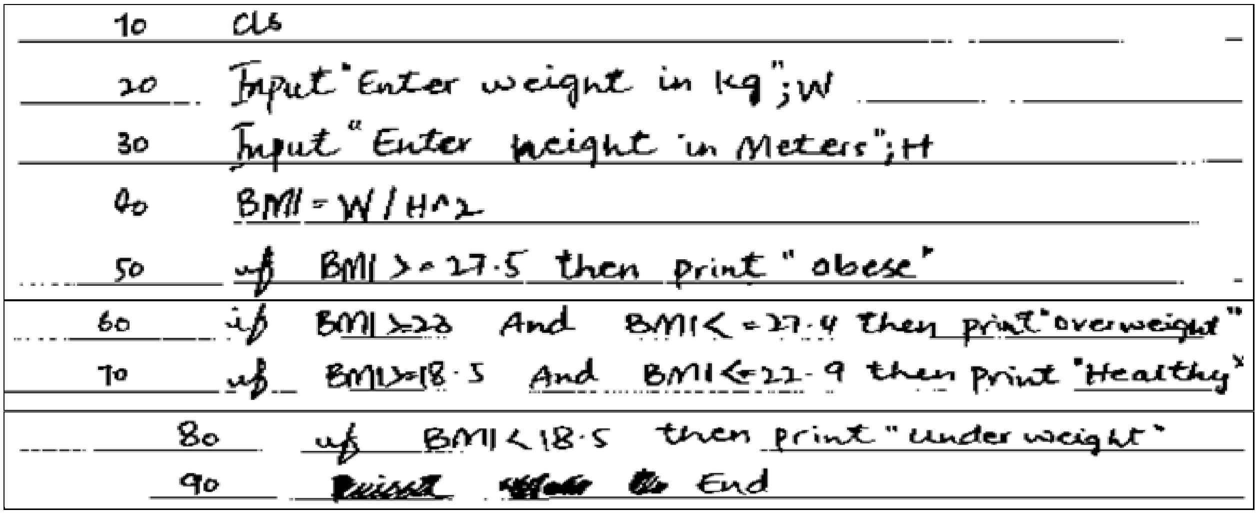
- input the weight in kg (W) and height in metres (H).
- calculate the BMI using the given formula.
- print appropriate weight category according to the criteria given in the following table.

$$BMI = \frac{W}{H^2}$$

BMI (kg/m^2)	Weight Category
>= 27.5	Obese
>= 23 AND < = 27.4	Overweight
>= 18.5 AND < = 22.9	Healthy
< 18.5	Underweight

Better responses used proper GW-BASIC statements to take input weight and height values. Moreover, these responses correctly converted the given formula into GW-BASIC expression. Likewise, these responses used conditional statements with correct range of BMI to display the weight category in output using appropriate GW-BASIC instructions.

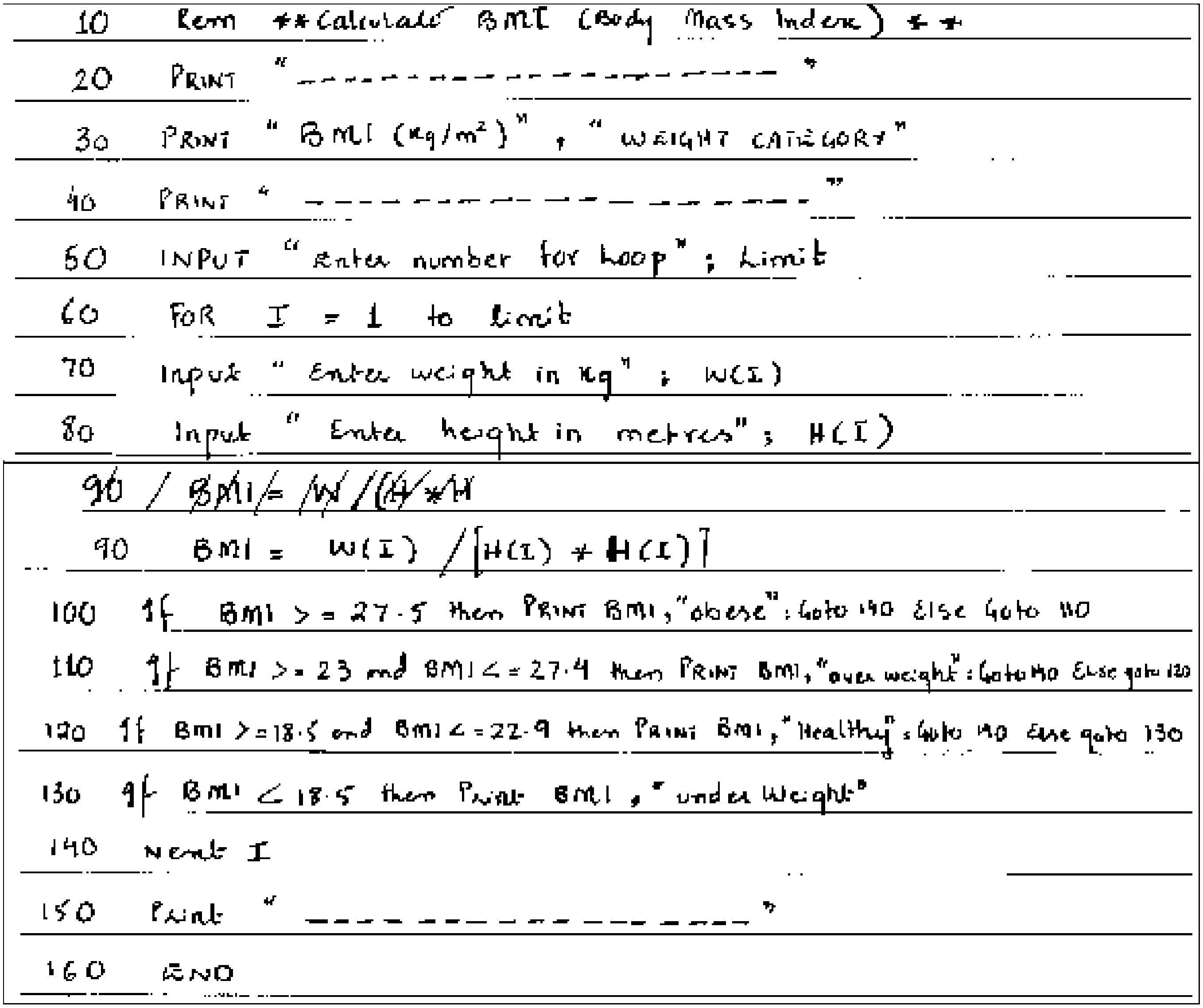
Example:



```
10 cls
20 Input "Enter weight in kg";W
30 Input "Enter height in Meters";H
40 BMI=W/H^2
50 if BMI>=27.5 then print "obese"
60 if BMI>=23 And BMI<=27.4 then print"overweight"
70 if BMI>=18.5 And BMI<=22.9 then print "Healthy"
80 if BMI<18.5 then print "underweight"
90 End
```

Weaker responses showed that candidates understood the question but were not able to write proper syntax of GW-BASIC statements for input and output purposes. Also, these responses were not able to convert the given BMI formula correctly into GW-BASIC expression. Moreover, these responses could not use correct conditional statements with appropriate range of BMI.

Example:



```
10  Rem **Calculate BMI (Body Mass Index) **
20  PRINT "-------------------------"
30  PRINT "BMI (kg/m²)", "WEIGHT CATEGORY"
40  PRINT "-------------------------"
50  INPUT "Enter number for Loop"; Limit
60  FOR I = 1 to Limit
70  Input " Enter weight in kg"; W(I)
80  Input " Enter height in metres"; H(I)
~~90 / BMI = W / (H * H~~
90  BMI = W(I) / [H(I) * H(I)]
100 If BMI >= 27.5 then PRINT BMI, "obese" : Goto 140 Else Goto 110
110 If BMI >= 23 and BMI <= 27.4 then PRINT BMI, "over weight" : Goto 140 Else goto 120
120 If BMI >= 18.5 and BMI <= 22.9 then PRINT BMI, "Healthy" : Goto 140 Else goto 130
130 If BMI < 18.5 then Print BMI, "under Weight"
140 Next I
150 Print "----------------------"
160 END
```