A Program:

एक कंप्यूटर प्रोग्राम:

निर्देशों का वह समूह जिनके द्वारा कंप्यूटर किसी कार्य को संपादित करता है, प्रोग्राम कहलाता है| अनेकों प्रोग्रामो के संकलन, जिनके माध्यम से किसी बड़े कार्य को निष्पादित किया जा सके, सॉफ्टवेयर कहलातें हैं| कंप्यूटर में किसी भी प्रोग्राम या निर्देश को ऐसी भाषा में लिखा जाता है जिसे कंप्यूटर आसानी से समझ सके| एक कंप्यूटर मशीनी भाषा या बाइनरी भाषा को आसानी से समझ लेता है| अतः वह पहली भाषा जिसे कंप्यूटर प्रोग्राम लिखने के लिए प्रयोग किया गया, मशीनी भाषा थी| मशीनी भाषा कि अपनी कुछ कमियाँ हैं जो निम्नलिखित हैं:

1- Very difficult to write programs because binary codes are not easy to remember.

ये भाषा कंप्यूटर प्रोग्राम लिखने के लिए बहुत कठिन है क्यूंकि बाइनरी कोड याद रखना काफी मुश्किल है|

2- A Program made on Machine Language is Machine Dependent and can not be run on other machines.

मशीनी भाषा में लिखे प्रोग्राम केवल उस मशीन पर चलते हैं जिनपर इन्हें लिखा गया है, इस प्रोग्राम को किसी अन्य मशीन पर नहीं चलाया जा सकता है| ये प्रोग्राम मशीन निर्भर प्रोग्राम कहलाते हैं|

चूँकि प्रत्येक इलेक्ट्रॉनिक मशीन की अपनी इलेक्ट्रॉनिक सर्किट संरचना एवं अपने रजिस्टर के सेट होतें हैं, अतः मशीन लैंग्वेज पर बने प्रोग्राम, मशीन निर्भर होने के कारण केवल उस मशीन पर चलते है जिनपर इन्हें लिखा गया है| इस कारण से एक नई प्रोग्रामिंग भाषा की आवश्यकता महसूस की गई जो कि मशीन निर्भर न हो | वह नई भाषा जो अस्तित्व में आई, उसे असेंबली लैंग्वेज कहा गया|

Assembly Language:

असेंबली लैंग्वेज:

प्रोग्रामिंग भाषा के विकास के क्रम में सर्वप्रथम विकसित होने वाली भाषा असेंबली लैंग्वेज थी| असेंबली लैंग्वेज में ऑपरेशन कोड को प्रस्तुत करने के लिए जो कोड प्रयोग किया जाता है उसे न्युमोनिक्स कहते हैं, तथा एड्रेस को प्रस्तुत करने के लिए करैक्टर स्ट्रिंग का प्रयोग किया जाता है |

इस प्रकार मशीन लैंग्वेज के कोड का असेंबली लैंग्वेज में पंक्तिवार (लाइन बाई लाइन) रूपांतरण कर दिया जाता है|

SEMESTER V

जैसा कि ज्ञात है कि असेंबली लैंग्वेज में न्युमोनिक्स तथा एड्रेस मशीन कोड को पंक्तिवार (लाइन बाई लाइन) रूपांतरित कर दिया जाता है, अतः असेंबली लैंग्वेज में भी बने प्रोग्राम मशीन विशेष के लिए ही होते है अर्थात असेंबली लैंग्वेज में बने प्रोग्राम भी मशीन निर्भर होतें हैं| साथ ही साथ चूँकि कंप्यूटर केवल बाइनरी भाषा ही समझता है इसलिए असेंबली लैंग्वेज में बने प्रोग्राम को भी रन करने के लिए मशीन कोड में परिवर्तित कराना होता है| यह कार्य एक विशेष प्रोग्राम द्वारा सम्पादित कराया जाता है जिसे असेम्बलर कहतें हैं| असेम्बलर वास्तव में एक अनुवादक होता है जो मशीन कोड को न्युमोनिक्स तथा एड्रेस में अनुवाद कर देता है |

असेम्बलर के एक्शन के द्वारा हमें असेंबली लैंग्वेज में बने प्रोग्राम के दो भाग प्राप्त होतें हैं|

सोर्स प्रोग्राम

Source Program

ऑब्जेक्ट प्रोग्राम

Object Program

सोर्स प्रोग्राम असेंबली लैंग्वेज में लिखे निर्देश होतें है जबकि ऑब्जेक्ट प्रोग्राम असेम्बलर द्वारा मशीन कोड में परिवर्तित प्रोग्राम होतें हैं| कंप्यूटर का प्रोसेसर मशीन भाषा के कोड को प्राप्त करता है, उसे प्रोसेस करता है तथा अंतिम आउटपुट प्रदान करता है|

Assembly language also has shortcomings like machine language:

असेंबली लैंग्वेज में भी मशीन लैंग्वेज के सामान कमियां हैं|

१- सभी न्युमोनिक्स तथा प्रत्येक एड्रेस को याद रखना होता है जोकि बहुत ही कठिन कार्य है|
२- प्रोग्राम अभी भी मशीन पर निर्भर है|

Classification of Programming Languages:

प्रोग्रामिंग लैंग्वेज के वर्गीकरण:

Low Level Languages

लो लेवल लैंग्वेज

Machine Language

Assembly Language

High Level Language

SEMESTER V

हाई लेवल लैंग्वेज

FORTRAN

PASCAL

COBOL

BASIC etc.

High Level Language:

हाई लेवल लैंग्वेज –

उच्च स्तरीय भाषा (High level language) सुविधाजनक होने के लक्षणों को ध्यान में रखकर बनाया गया है, इसका अर्थ यह कि ये भाषा मशीन पर निर्भर नहीं करती है | यह भाषा अंग्रेजी भाषा के कोड जैसी होती है, इसलिए इसे कोड करना या समझना सरल होता है | इसके लिए एक Translator की आवश्यकता होती है, जो उच्च स्तरीय भाषा के Program को मशीन कोड में translate करता है इसके उदाहरण है – फॉरटरैन (FORTRAN), बेसिक (BASIC), कोबोल (COBOL), पास्कल (PASCAL), सी (C), सी++ (C++), जावा (JAVA), VISUAL BASIC, Visual Basic.net HTML, Sun Studio आदि इसी श्रेणी (Category) की भाषा है इसको दो generation में बाँटा गया गई|

1. Third Generation Language
2. Fourth Generation Language

जिस प्रकार से असेंबली लैंग्वेज के लिए अनुवादक असेम्बलर होता है उसी प्रकार से हाई लेवल लैंग्वेज के अनुवादक को हम कम्पाइलर कहतें हैं| कम्पाइलर सम्पूर्ण प्रोग्राम को सर्वप्रथम चेक करता है तथा त्रुटियाँ होने कि दशा में उन्हें दर्शाता है| त्रुटियाँ दूर हो जाने के पश्चात् प्रोग्राम को प्रोसेस करता है तथा आउटपुट प्रदान करता है|कम्पाइलर तथा असेम्बलर में मुख्य अंतर यह होता है कि कम्पाइलर सम्पूर्ण प्रोग्राम को एक साथ कम्पाइल करता है, त्रुटियाँ दर्शाता है तथा त्रुटियाँ दूर हो जाने के पश्चात् प्रोग्राम को प्रोसेस करता है तथा आउटपुट प्रदान करता है जबकि असेम्बलर निर्देश कि प्रत्येक लाइन को चेक कर के त्रुटियाँ दर्शाता है तथा त्रुटियाँ दूर हो जाने के पश्चात् दूसरी लाइन को चेक करता है|

Low Level Language:

लो लेवल लैंग्वेज:

वह भाषाएँ (Languages) जो अपने संकेतो को मशीन संकेतो में बदलने के लिए किसी भी अनुवादक (Translator) को सम्मिलित नही करता, उसे निम्न स्तरीय भाषा कहते है अर्थात निम्न स्तरीय भाषा के कोड को किसी तरह से अनुवाद (Translate) करने की आवश्यकता नही होती है | मशीन भाषा (Machine Language) तथा असेम्बली भाषा (Assembly Language) इस भाषा के दो उदाहरण है| लेकिन इनका उपयोग प्रोग्राम (Program) में करना बहुत ही कठिन है | इसका उपयोग करने के लिए कम्प्यूटर के हार्डवेयर (Hardware) के विषय में गहरी जानकारी होना आवश्यक है | यह बहुत ही समय लेता है और त्रुटियों (Error) की सम्भावना अत्यधिक होती है | इनका संपादन (Execution) उच्च स्तरीय भाषा (High level language) से तेज होता है | ये दो प्रकार की होती है –

1. मशीन भाषा (Machine Language)
2. असेम्बली भाषा (Assembly Language)

Middle Level Language:

मिडिल लेवल लैंग्वेज:

जैसा कि ज्ञात है कि मशीन लैंग्वेज पर बने प्रोग्राम की मशीन पर निर्भरता होती है परन्तु प्रोग्राम का प्रोसेस तेज़ होता है, वहीं दूसरी ओर हाई लेवल लैंग्वेज पर बने प्रोग्राम की मशीन पर निर्भरता नहीं होती है परन्तु प्रोग्राम का प्रोसेस धीमा होता है, अतः एक ऐसी प्रोग्रामिंग लैंग्वेज कि अवधारणा अस्तित्व में आई जिसमे इन दोनों कमियों को दूर कर दिया गया|इस ल्प्रकर कि लैंग्वेज को मिडिल लेवल लैंग्वेज कहा गया| C तथा C++ मिडिल लेवल लैंग्वेज के उदहारण हैं|

A computer Program:

किसी भी प्रोग्रामिंग भाषा में चरणबद्ध तरीके से लिखे निर्देशों का वह समूह जिनके द्वारा कंप्यूटर किसी कार्य को संपादित करता है, प्रोग्राम कहलाता है| प्रोग्राम को समझने के लिए एक साधारण उदहारण लेते हैं-

यदि हमें चाय बनाना हो तो, इस कार्य को चरणबद्ध तरीके से सम्पादित करने की प्रक्रिया निम्न प्रकार से हो सकती है-

<u>चरण 1.</u>

युटेंसिल, छन्नी, दूध, पानी, चाय पत्ती, चीनी तथा स्टोव इकट्ठा करें|

<u>चरण 2.</u>

स्टोव जलाएं|

चरण 3:

दूध तथा पानी की आवश्यकता मात्रा के युटेंसिल्स में लें।

चरण 4:

चीनी तथा चाय की पत्ती की आवश्यक मात्रा को युटेंसिल्स में डालें।

चरण 5:

स्टोव को जलायें।

चरण 6:

मिश्रण को 5–10 मिनट तक उबालें

चरण 7:

चाय को प्याली में छानें।

चरण 8:

चाय को परोसें।

इसी प्रकार से हम किसी प्रोग्रामिंग भाषा का अपना (Syntax) व स्ट्रक्चर (Structure) होता है। जिसका प्रयोग लिखने में करते है। दो नंबरों को जोड़ने के लिए बेसिक (Basic) प्रोग्रामिंग भाषा में हम प्रोग्रामिंग निम्नलिखित तरह से करेंगे–

```
10 LET  A = 10
20 LET B = 20
30 PRINT C
```

SEMESTER V

बेसिक भाषा को भी उच्च भाषा की श्रेणी में रखा गया है जिसके द्वारा कंप्यूटर के लिए अथवा चित्रीय प्रोग्रामिंग किया जा सकता है। इस भाषा को लिखना कंपाइल क्रियान्वित होते है। इसका टेक्स्ट एडिटर हमें CUI प्रदान करता है तथा माउस प्वाइंटर हमें यूजर फ्रेंडली वातावरण प्रदान करता है।

प्रोग्रामिंग भाषा के कुछ अवयव होते है जो Variable, Constants, Reserve Keyword, Symbol आदि कहलाते है।

तो प्रोग्राम में गलती की संभावना बनी रहती है। इसी प्रकार रिजर्व प्रतीक के भी सुनिश्चित अर्थ होते हैं। इन रिजर्व प्रतीकों का उपयोग हम प्रोग्रामिंग भाषा में Logical या Arithemetical आपरेशन करने के लिए करते हैं।

उदाहरण –

(,), “,” +, -, %, *, /, <, >.

वैरियेबल –

वैरियेबल इस प्रकार के मान होते है जो प्रोग्राम के क्रियान्वित होने पर बदलते रहते हैं। वैरियेबल को Identifier के नाम से जाना जाता है तथा ये मेमोरी में ऐसे स्थान को दर्शाते है जहां डेटा स्टोर किया जाता है। वास्तव में वैरियेबल ऐसे भंडार होतें हैं जो अलग–अलग प्रकार के डेटा को संरक्षित करते है।

कॉन्सटेंट –

कान्सटेंट ऐसे मान होते हैं जो प्रोग्राम के क्रियान्वित होने पर कभी भी नहीं बदलते है।

SEMESTER V

1. न्यूमेरिक वैरियेबल
2. स्ट्रिंग वैरियेबल

नंबर को ही संरक्षित कर सकते है।

उदाहरण – A, B, A, A-B आदि।

स्ट्रिंग वेरियेबल के साथ डॉलर के चिन्ह होते हैं जो की अक्षर तथा अक्षर–अंक दोनों प्रकार के डेटा को संरक्षित करते है।

ऐरे –

ऐरे भी एक प्रकार वैरियेबल होता है जो मेमोरी की स्थिति को प्रदर्शित करता है और उसे कई छोटे–छोटे भागों में बॉट देता है और प्रत्येक प्रत्येक भाग डेटा को संचित कर सकता है।

बेसिक भाषा के कुछ महत्वपूर्ण निर्देश (Command) –

CLS :- स्क्रीन को साफ करता है।

Print :- स्क्रीन पर सूचनाऍं प्रिंट करता है।

End :- प्रोग्राम को समाप्त करता है।

SEMESTER V

<u>GOTO :-</u> किसी प्रोग्राम में कंट्रोल को एक बिंदु में दूसरी बिंदु पर भेजने के लिए। इसके द्वारा कंट्रोल को अपनी तत्कालिक स्थिति से आगे या पीछे भेज सकते है।

<u>LOCATE :-</u> स्क्रीन पर किसी स्थान को निर्धारित करने के लिए।

<u>FOR NEXT :-</u> किसी निर्दश के समूह को अनेको बार क्रियानवित करने के लिए। यदि किसी FOR-NEXT लूप में कोई भी निर्देश नहीं है।

तो इसे विलम्बित लूप कहते है।

<u>RND :-</u> इस निर्देश द्वारा अनिश्चित अंक उत्पन्न कराये जाते है। RND द्वारा 0 और 9 के बीच के अंक उत्पन्न कराये जाते हैं। 1 और 100 तक के अनिश्चित अंकों को उत्पन्न करने के लिए INT(RND * 100) निर्देश का प्रयोग करेंगे।

<u>LEN :-</u> किसी शब्दों के समूह में अक्षरों की संख्या देता है।

<u>INKEYS $:-</u> यह निर्देश अक्षरों की – बोर्ड से ग्रहण कराता है लेकिन INPUT निर्देश की तरह एन्टर की को दबाने की प्रतीक्षा नहीं करता है।

<u>उदाहरण 1 –</u>

किसी ऐरे का प्रयोग करके पॉच फलों के नाम को इनपुट कराने तथा उसी समय स्क्रीन पर दर्शाने के लिए BASIC प्रोग्राम –

REM “Display name of fruit”

SEMESTER V

```
FOR count = 1 to 5
INPUT "Type the name of seasonal fruit " Fruit$(count)
NEXT count
CLS
For  n = 1 to 5
LOCATE n + 2.20
PRINT Fruit$(n)
NEXT n
END
```

दूसरे DIM निर्देश कप्यूटर को यह बताता है कि एक ऐरे के लिए कितनी मेमोरी की आवश्यकता है। यहॉ DIM Fruit$(5) यह इंगित करता है कि ऐसे में पॉच अवयव हैं। यदि हम सात नागों को इनपुट करना चाहते हैं तो DIM निर्देश क्या होगा?

For-Next लूप में count को वैरियेबल की तरह प्रयोग किया गया है। Fruit(count) का मतलब है Fruit(1), Fruit(2)......हम जानते है कि RND निर्देश अनिश्चित संख्या उत्पन्न करना चाहते हैं तो RND निर्देश का प्रयोग करते हैं। परंतु यदि हम अनिश्चित नंबर उत्पन्न करना चाहते हैं तो Randomize Timer का प्रयोग करते है। Randomize Timer प्रत्येक बार पुनः व्यवस्थित होता है जिससे आप एक जैसी अनिश्चित संख्याओं की श्रृंखला बार बार नहीं प्राप्त करेंगे।

SEMESTER V

Fuit $ (5), Count वैरियेबल के मान के अनुसार –

प्रिटिंग के लिए एक दूसरे वैरियेबल n का प्रयोग किया गया है। n = 2.20 का LOCATE निर्देश में समान्यतः खाली स्थान के लिए प्रयोग किया गया है। हम एक वैरियेबल को Locate तथा FOR-NEXT दोनों के लिए उपयोग कर सकते हैं।

उदाहरण–2:

इंटिजर 1 या 2 को 50 बार रैंडम्ली उत्पन्न करने के लिए एक प्रोग्राम बनायें तथा यह देखें कि अंक 1 और 2 कितनी बार प्राप्त हुए हैं। उपरोक्त के लिए प्रोग्राम इस प्रकार हैं–:

The program for this is as follows:

Label for the program

Clear the screen

Initialize random number generation

Define the array size

Generate random numbers

And store in num(1) and num(2)

Display count

End the program

REM"Find the number of 1s and 2s"

SEMESTER V

```
Cls
RANDOMIZE TIMER
DIM Num(2)
FOR count 1 TO 50
LET n = INT(RND * 2+1)
LET num(n) = num(n)+1
NEXT count
PRINT "Number of 1s were", num(1)
PRINT "Number of 2s were", num(2)
END
```

उपरोक्त प्रोग्राम के ऐरे केवल दो अवयव Num(1) तथा Num(2) है।

चलिए इस निर्देश की उपयोगिता समझते है। हम जानते हैं कि RND निर्देश अनिश्चित संख्या उत्पन्न करता है। यदि आप एक जैसे 50 बार अनिश्चित संख्या उत्पन्न करना चाहते हैं तो RND निर्देश का प्रयोग करते हैं। परंतु यदि हम अनिश्चित नंबर उत्पन्न करना चाहते है तो Randomize Timer का प्रयोग करते है। Randomize Timer प्रत्येक बार पुनः व्यवस्थित होता है जिसमें आप एक जैसी अनिश्चित संख्याओं की श्रृंखला नहीं प्राप्त करेंगे।

SEMESTER V

उदाहरण – 3:

एक प्रोग्राम लिखिए जिसमें एक छात्र द्वारा पॉच विषयों के प्राप्तांको के योग की गणना प्राप्त हो तथा यदि इसका प्रतिशत 90 से ज्यादा हो तो संदेश “Good Performance” प्रिंट हो।

```
REM "Calculate total score"
CLSe
DIM score(5), Subject$(5)
Subject$(1) = "English"
Subject$(2) = "Mathmetics"
Subject$(3) = "Hindi"
Subject$(4) = "Science"
Subject$(5) = "Social Studies"
INPUT "Name of students", Name$
FOR n = 1 to 5
PRINT Subject$(n);";"
INPUT "Enter marks out of 100",score(n)
Total = total + score(n)
PRINT Name$;"Has score a total of",total,"marks"
IF total/5>90 THEN PRINT "Good Performance"
```

SEMESTER V

END

Go Sub. निर्देश की तरह ही On-Go Sub निर्देश होता है। Go Sub निर्देश के द्वारा हम अपने प्रोग्राम में एक निश्चित लेबल पर पहुँच जाते है। No-Go Sub निर्देश भी निर्देश की तरह कार्य करता है अंतर केवल यह है कि Return निर्देश प्रोग्राम के कंट्रोल को स्वतः ही मुख्य प्रोग्राम को दे देता है। इसलिए यहाँ पर "Go to" निर्देश की आश्यकता नहीं होती है।

उदाहरण :–

नीचे दिया गया प्रोग्रामएक मेन्यू दिखाता है जिसमें विभिन्न प्रकार के भोजन के वर्गों के नाम दिखेंगे तथा ग्राहक के चुने गये वर्ग के अंर्तगत आने वाले भोजन के नामों की सूची प्राप्त होती है।

Given below is the main menu program and two subroutines and snakes.

REM "Menu program"

Main:CLS

PRINT "1. Beverage"

PRINT "2. Snakes"

PRINT "3. Indian Food Vegetarian"

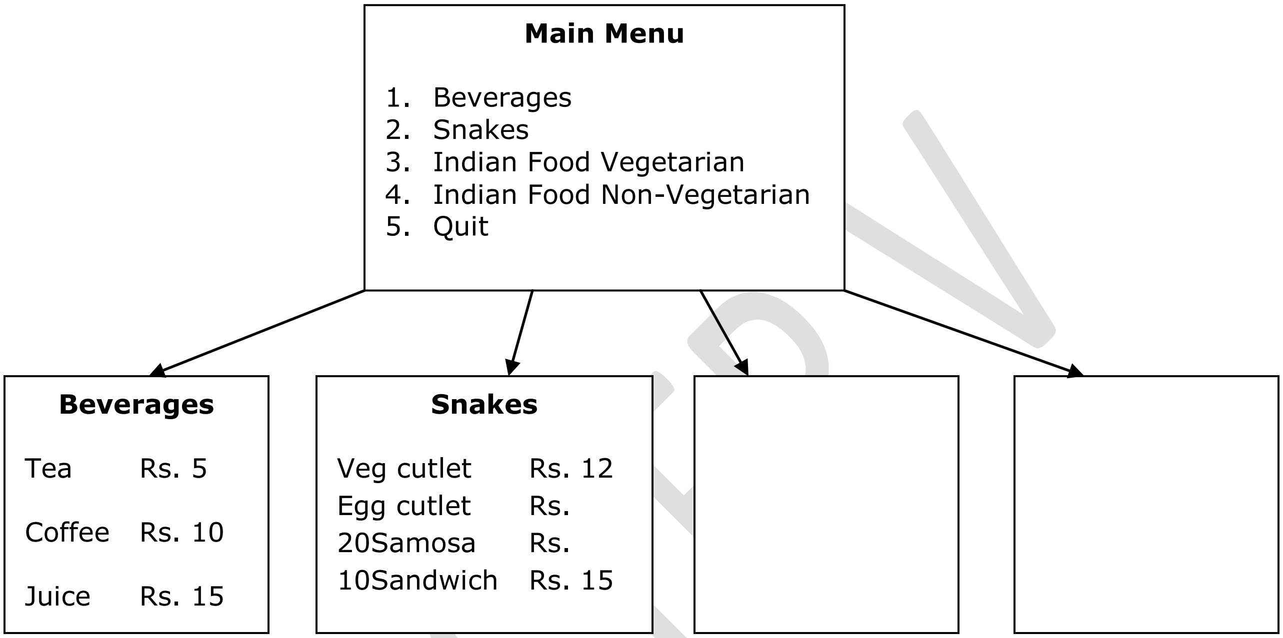
PRINT "4. Indian Food Vegetarian"

PRINT "5. Quit"

```
INPUT "Type the number of your choice",choice
On choice GOTO Beverages, Snakes, Indian, Food Vegetarian, Indian Non-Vegetarian
Beverage: CLS
PRINT "Tea     Rs.5"
PRINT "Coffee        Rs.10"
PRINT "Juice   Rs.15"
PRINT "Cola    Rs.12"
PRINT "Press any key to go back to main menu"
Waiting: A$ = INKEY$
IF A$ = "" THN GOTO waiting ELSE GOTO main
Snacks: CLS
PRINT  " Veg Cutlet              Rs.12"
PRINT "Egg Cutlet          Rs.20"
PRINT "Samosa              Rs.10"
PRINT "Sandwich            Rs.15"
PRINT "Press any key to go back to main menu"
Waiting: A$ = ""THEN GOTO waiting ELSE GOTO main
Quit: CLS
```

SEMESTER V

END



Subroutine में INKEYS$ कमांड कंप्यूटर को तब तक प्रतीक्षारत रखता है जब तक कि प्रयोगकर्ता द्वारा कोई न दबाई जाए। जैसे ही किसी की को दबाया जाता है, कंट्रोल मेन मेन्यू पर चला जाता है।

Q4- QBASIC की सहायता से आप चित्रीय प्रोग्राम किस प्रकार बना सकते हैं। पॉच चित्रीय कमांड की व्याख्या करें।

1. ग्राफिक कमांड

   SCREENn, PSET, LINE, CIRCLE, और PAINT इत्यादि कमांड है।

2. CIRCLE(X1, Y1), r

केंद्र X1, Y1 तथा त्रिज्या का वृत्त बनाने के लिये।

CIRCLE(X1, Y1), r,c

के लिए इसी वृत्त को किसी रंग से बनाने के लिये।

वृत्त खण्ड बनानाः

वृत्त का एक भाग वृत्तखंड कहलाता है। एक वृत्तखंड बनाने के लिए इसके प्रारंभिक तथा कोंण का मान देना होगा। ये मान डिग्री में न हो कर रेडियन में मापे जाते है।

डिग्री तथा रेडियन में संबंध निम्न प्रकार हैं

90 = 3.14/2 radians and 360=2x3.14 radians = 6.28

अतः डिग्री को रेडियन में बदलने का सामान्य फार्मूला है।

degree * 3.14/180

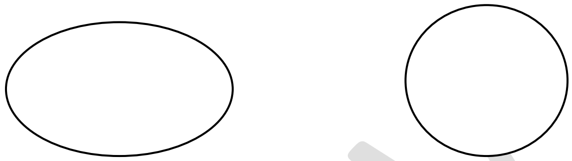
Circle(100,200),50,4,0,1.57

उरोक्त कमांड के द्वारा एक लाल रंग का वृत्तखंड बनाया जाता है जिसका केन्द्र (100,200) होगा त्रिज्या 50 होगी इसका प्रारंभिक कोण 0^0 होगा एवम अंतिम कोंण 90^0 होगा।

SEMESTER V

अंडाकार बनाया:

वास्तव में अंडाकार एक वृत्त ही होता है जिसे क्षैतिज अथवा उध्वार्धर दिशा में खींचा गया हो।

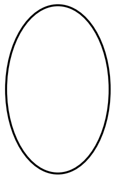


Circle stretched horizontally to make ellipse

Circle कमांड के द्वारा अंडाकार भी बनाया जा सकता है। इसके लिए y-axis and x-axis के अनुपात (आस्पेक्ट रेशियो) में परिवर्तन जैसे 0.5 करना होता है।

3. Circle(100,200),20,3,,,,0.5
   उपरोक्त कमांड क्षैतिज दिशा में खींचा गया एक अंडाकार देता है।

4. Circle(100,200),20,3,,,1.5
   उपरोक्त कमांड उर्ध्वाधर दिशा में खींचा गया एक अंडाकार देता है तथा इसका आस्पेक्ट रेशियो 1 से अधिक है।



त्री विमीय प्रभाव:

कमांड के कुशलतापूर्वक प्रयोग द्वारा त्री विमीय चित्रण(3-D) किया जा सकता है। निम्न प्रोग्राम को टाइप करके क्रियानवित करें प्रभाव देखें।

REM “3D Effect”

SEMESTER V

CLS

SCREEN 12

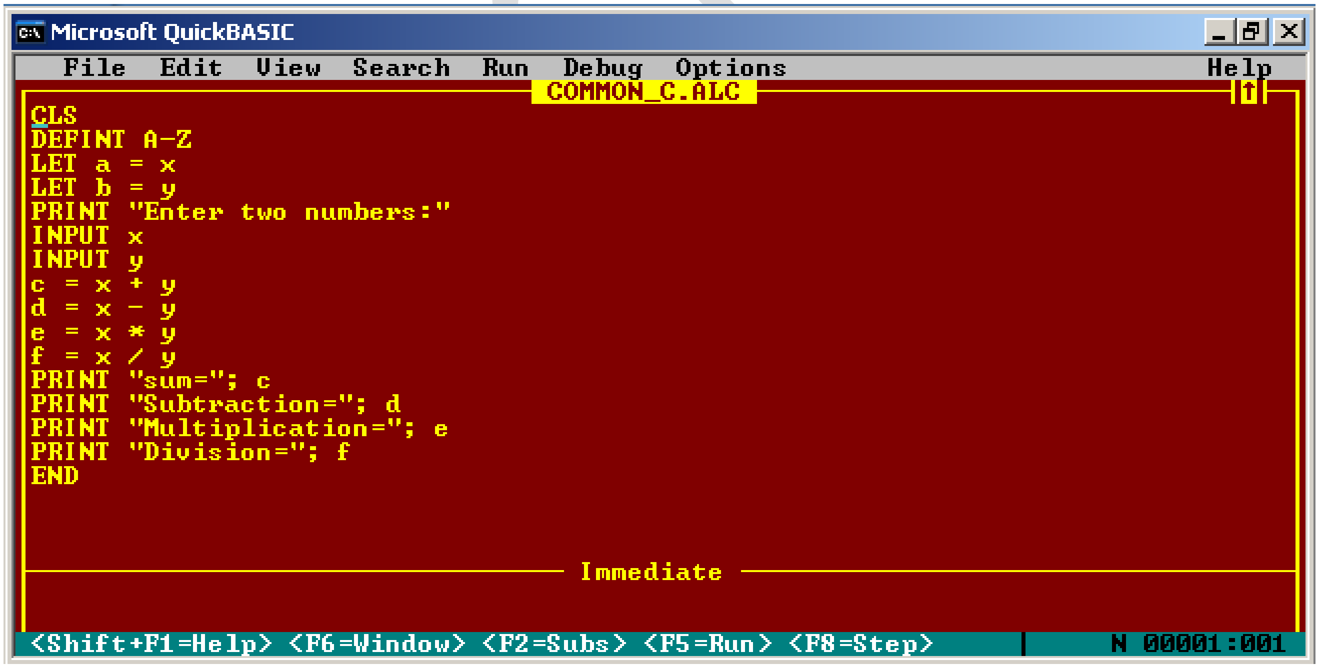
FOR Y=300 to 240 STEP-4

CIRCLE(240,y), y-200,3,,,0.5

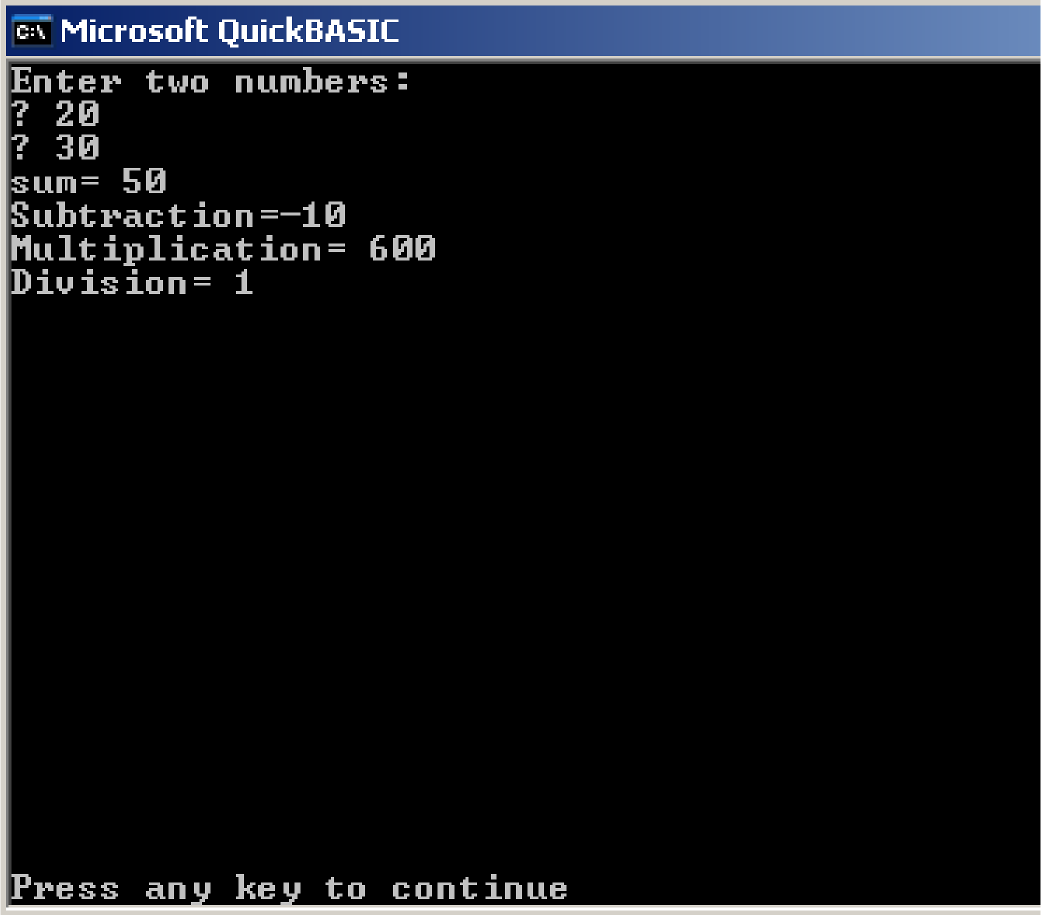
NEXT y

Some Simple QBASIC Programs :

1- Program for Common Calculation, to input two digits and to get their Sum, Subtraction, Multiplication & Division.

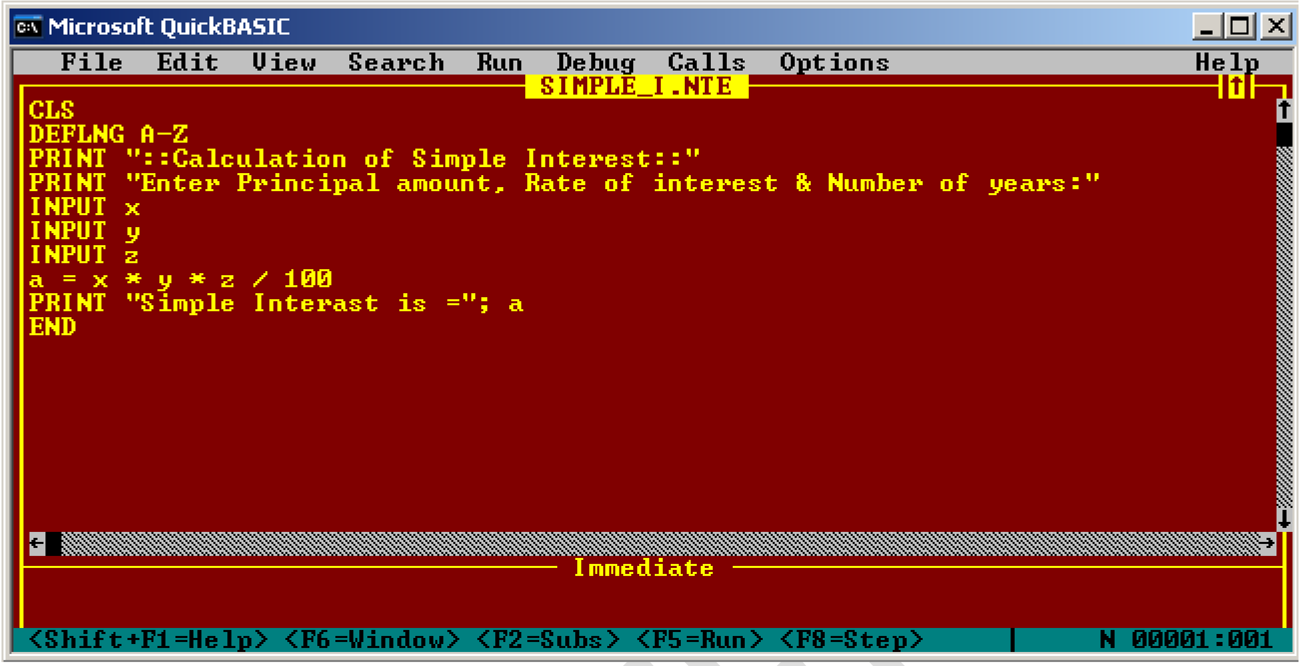


Code Window

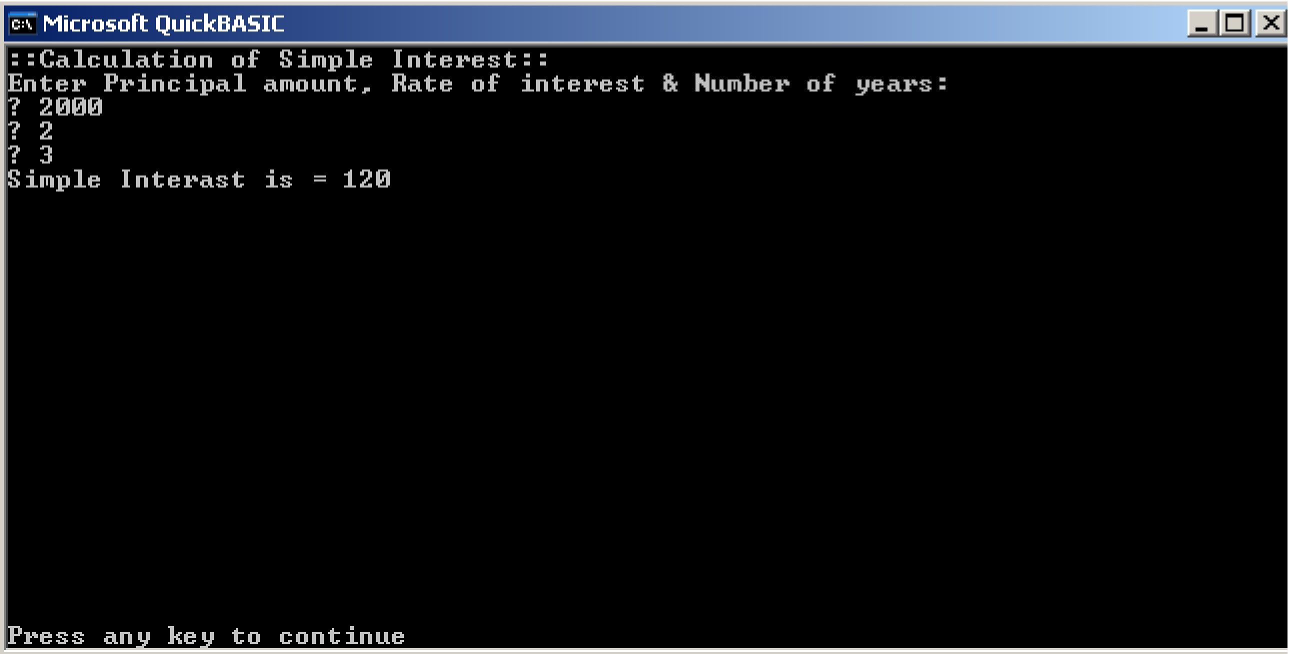


Run Time Window

2- Program to obtain simple interest while entering “Principle amount”, Rate of interest” and Number of years”.

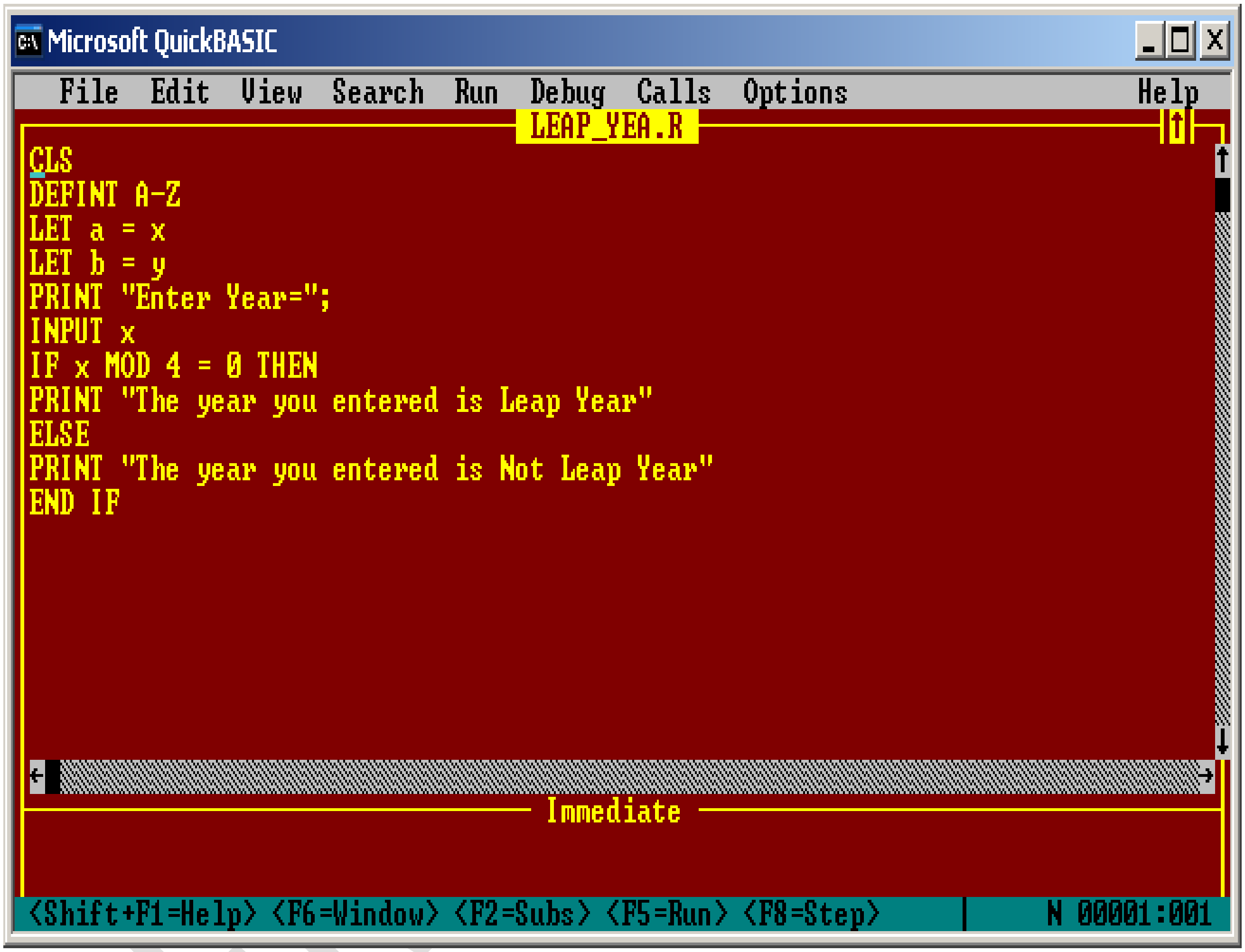


Code Window

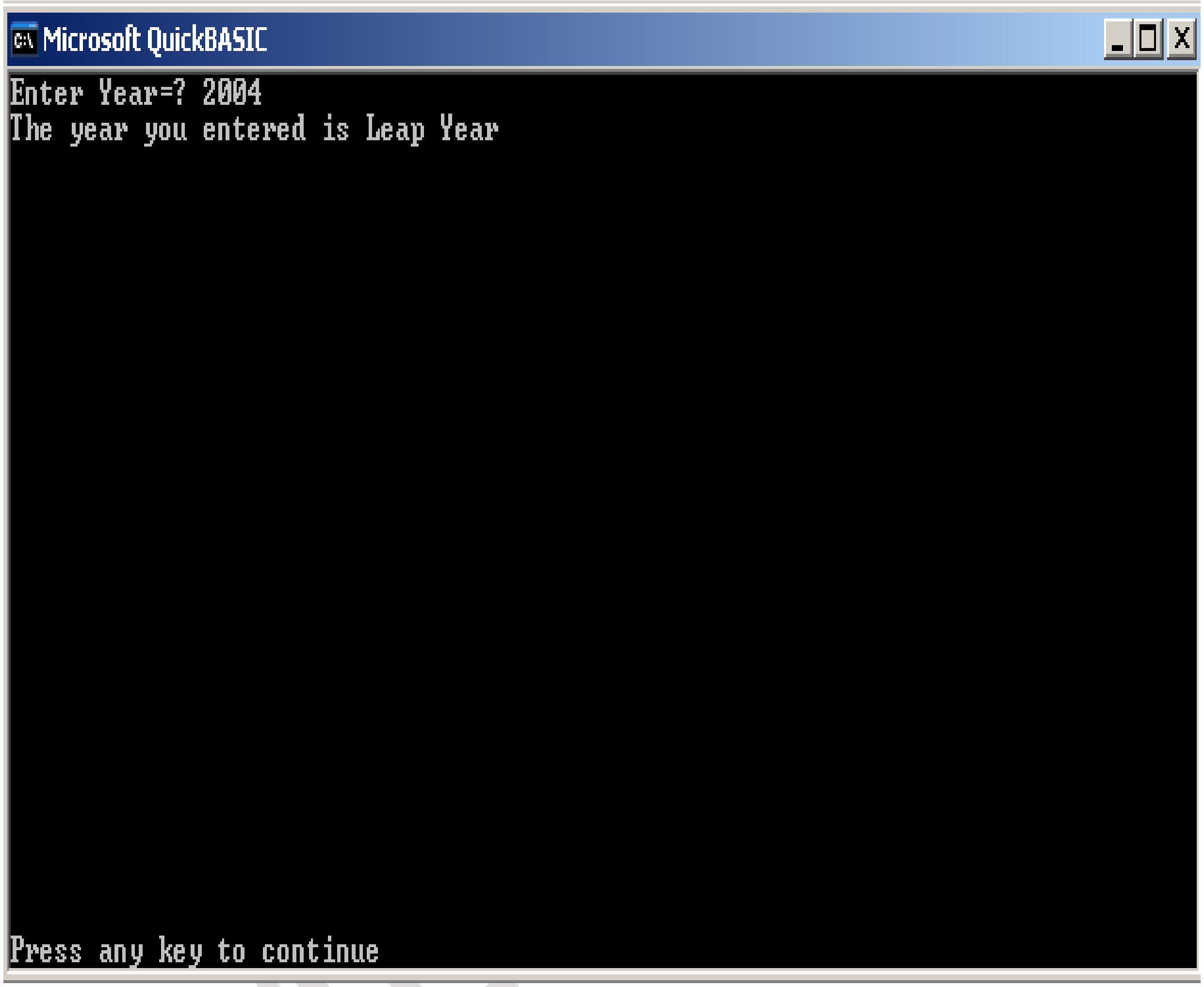


Run Time Window

3- Program to enter “Year” and to obtain whether this year is Leap Year or not?

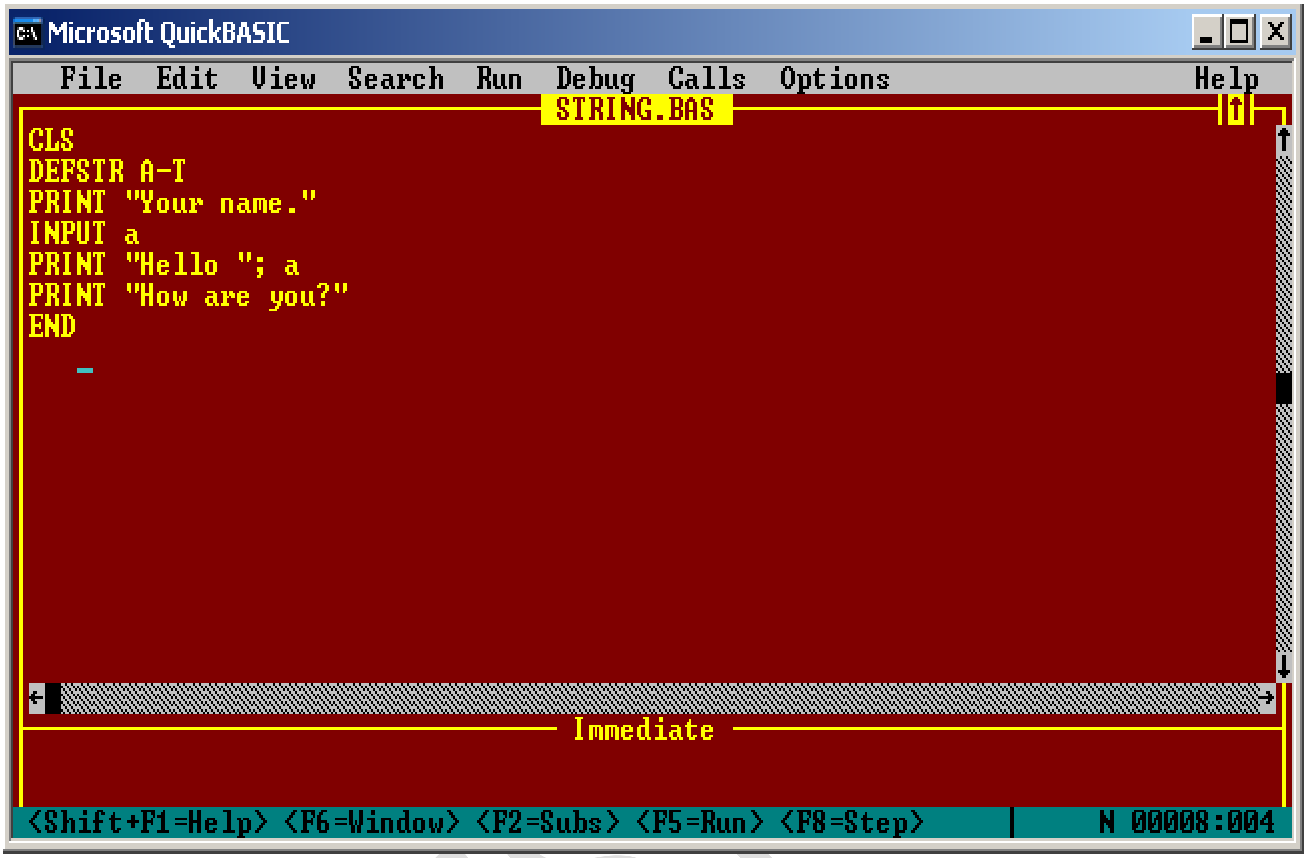


Code Window

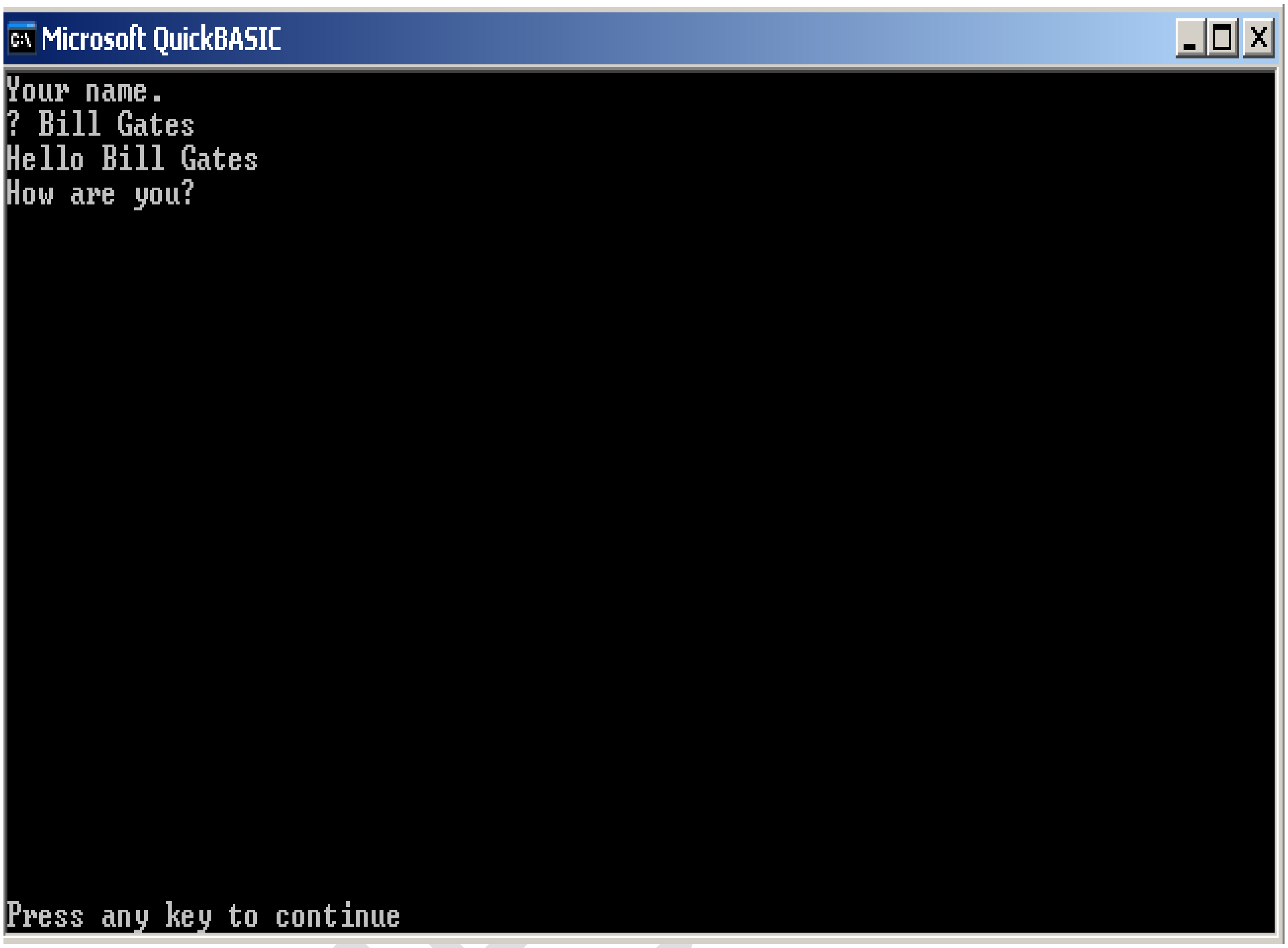


Run Time Window

4- Program to enter any string and obtain any message including that string.


Microsoft QuickBASIC
File Edit View Search Run Debug Calls Options Help
STRING.BAS
CLS
DEFSTR A-T
PRINT "Your name."
INPUT a
PRINT "Hello "; a
PRINT "How are you?"
END
Immediate
<Shift+F1=Help> <F6=Window> <F2=Subs> <F5=Run> <F8=Step>
N 00008:004

Code Window



Run Time Window

SEMES