

8. The program below provides an encoding procedure Encode for an English word and a corresponding decoding procedure Decode to decode English word. Some program statements of the Decode procedure are missing.

Line Number	Program statement
100	Program CODE(input, output);
110	Const M = 5;
120	D = 21;
130	N = 26;
140	A = 65;
150	Var ENG, SEC : string;
160	Procedure Encode(C:string; var P:string);
170	Var Ch, Tmp : char;
180	Len, Count, Asc : integer;
190	Begin
200	P := '';
210	Len := length(c);
220	For Count := 1 to Len do
230	Begin
240	Ch := C[Count];
250	Asc := ord(Ch) - A;
260	Asc := (M*Asc) mod N;
270	Tmp := chr(Asc + A);
280	P := P + Tmp;
290	End;
300	End;
310	Procedure Decode (Missing Statement);
320	Var Ch, Tmp : char;
330	Len, Count, Asc : integer;
340	Begin
350	C := '';
	Missing Statements
800	End;
810	Begin
820	Writeln('Please type an English word: ');
830	Readln(Eng);
840	Encode(ENG, SEC);
850	Writeln('The resulting secret word is: ', SEC);
860	Writeln('Please type a secret word: ');
870	Readln(SEC);
880	Decode(SEC, ENG);
890	Writeln('The original English word is:', ENG):
900	End.

- (a) (i) If the program segment from lines 820 to 850 is executed and the string variable ENG stores 'ABLE' as input, complete the following:

After the executing the FOR looping when Count = 1, Ch = __, Tmp = __, P = __

After the executing the FOR looping when Count = 2, Ch = __, Tmp = __, P = __

After the executing the FOR looping when Count = 3, Ch = __, Tmp = __, P = __

After the executing the FOR looping when Count = 4, Ch = __, Tmp = __, P = __

In line 850, SEC = _____

- (ii) The encoding and the decoding formulae in the program are $P = (5 \times C) \bmod 26$ and $C = (21 \times p) \bmod 26$ respectively where P and C are integers from 0 to 25. By using these two formulae, complete the procedure Decode to decode the string stored in the variable SEC generated by the procedure Encode.

```
procedure Decode ( _____ );  
Var Ch, Tmp : char;  
    Len, Count, Asc : integer;  
begin  
    C := '';
```

```
end;
```

- (iii) Write down the output of the variable ENG if the above procedure Decode is applied to the string stored in the variable SEC in (a) (i) and the decoding formula is changed to $C = (20 \times P) \bmod 26$.

(8 marks)

- (b) If this program is going to handle only small case letters (a.. z), one statement in the program should be changed. Write down the line number and the amended statement.

(2 marks)

9. Tom writes a Pascal program to check whether an English word is anagram of another English word.

One English word is anagram of another English word if the second word is made by changing the order of the letters of the first word. For example,

- (1) ITEM is an anagram of TIME.
- (2) EXAM is not anagram of TIME.

Assume that all inputs are correct English words composed of letters of the English alphabet only. Tom is going to write a Pascal program to perform the following tasks:

Task I. Initialize two arrays to store the frequencies of the letters in each of the two words.

Task II. Input two English words.

Task III. Check whether the number of letters in the two words are equal.

Task IV. Change all the letters in the words into upper case.

Task V. Calculate the frequencies of the letters in each word.

Task VI. Check whether the second word is an anagram of the first word.

Task VII. Display the result of this checking.

The following shows sample outputs of the program. (In these outputs, all the data following a colon is entered by the user via the keyboard. All other items are output from the program.)

Sample Output 1

```
Input the first English word:TIME
Input the second English word:ITEM
ITEM is an anagram of TIME
```

Sample Output 2

```
Input the first English word:time
Input the second English word:ITEM
ITEM is an anagram of TIME
```

Sample Output 3

```
Input the first English word:TIME
Input the second English word:EXAM
EXAM is not an anagram of TIME
```

Tom has already written the declaration and the main body of the program. Some procedures are missing. Part of the program is as follows:

```
Program Q9(input, output);
```

```
var
```

```
    freq1, freq2 : array['A'..'Z'] of integer;
    word1, word2 : string;
    c : char;
    i : integer;
    match : boolean;
```

```
procedure Init;
```

```
begin
```

```
    for c:= 'A' to 'Z' do
    begin
        freq1[c]:=0;
        freq2[c]:=0;
    end;
```

Missing procedures

```
begin
```

```
    match := true;
    Init;
    Inputwords;
    CheckLength;
    If match then begin
        ChangeUpper(word1);
        ChangeUpper(word2);
        CalFreq;
        CheckMatching
    end;
    if match then
        writeln(word2, 'is an anagram of ', word1)
    else
        writeln(word2, 'is not an anagram of ', word1);
end.
```

The following indicate the description of each identifier used in this program.

Identifier	Description
Freq1	An array used to store the frequency of letters in the first English word
Freq2	An array used to store the frequency of letters in the second English word
Word1	A variable used to store the first English word
Word2	A variable used to store the second English word
C	A character variable
I	An integer variable used as a counter
Match	A Boolean variable
st	A string variable parameter in procedure ChangeUpper

You are not allowed to add any new variable in answering the question.

- (a) Write the procedure InputWords that will perform Task II. (1 mark)
- (b) Write procedure CheckLength that will perform Task III. (2 marks)
- (c) Complete the produce parameter and write the procedure ChangeUpper that will perform Task IV.
- procedure ChangeUpper (_____); (3 marks)
- (d) Write the procedure CalFreq that will perform Task V. (2 marks)
- (e) Write the procedure CheckMatching that will perform Task VI. (2 marks)