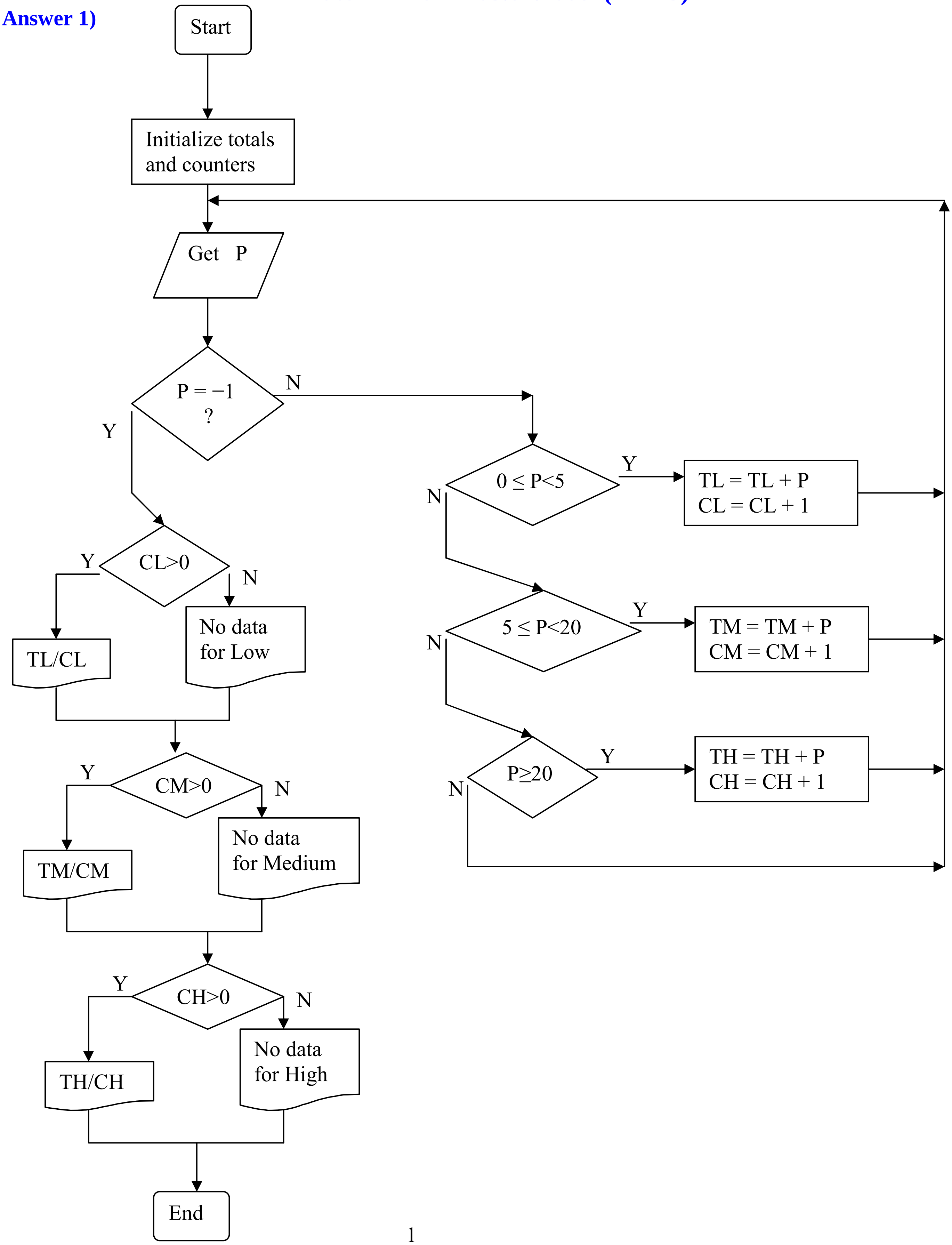
BIL105E - Introduction to Scientific and Engineering Computing
Midterm Exam - 03.04.2008 (KEYS)

Answer 1)



```
// SOLUTION OF POLLUTION QUESTION

#include <iostream.h>
#include <stdlib.h>

int main()
{
   int P; // Pollution variable
   int TL=0, TM=0, TH=0; // Totals for Low, Medium, and High
   int CL=0, CM=0, CH=0; // Counters for Low, Medium, and High

   cout.setf(ios::showpoint); // Allow decimal points to display

   // Get air pollution values from user:
   while (1) // Infinite loop
   {
    cout << "Enter P (-1 to stop) : ";
    cin >> P;

    if (P == -1) // Terminate infinite loop
      break;

    if (P >= 0 && P < 5)  { TL += P; CL++;}
    if (P >= 5 && P < 20) { TM += P; CM++;}
    if (P >= 20)          { TH += P; CH++;}
   }

   // Calculate averages and display messages:
   cout << "\n\n";
   cout << "Avg. of Low Levels= ";
   if (CL > 0) cout << (float) TL / CL;
   else cout << "No data";
   cout << endl;

   cout << "Avg. of Medium Levels= ";
   if (CM > 0) cout << (float) TM / CM;
   else cout << "No data";
   cout << endl;

   cout << "Avg. of High Levels= ";
   if (CH > 0) cout << (float) TH / CH;
   else cout << "No data";
   cout << endl;

   cout << "\n";
   system("PAUSE");
   return 0;
}
```

Answer 2)

```
// SOLUTION OF POLYGON QUESTION
/* TEST DATA FOR POLYGON PROBLEM:
   Inputs:                  N = 5 ,  Point coordinates = (3 7) (6 4) (3 -2) (-6 1) (-6 7)
   Expected outputs:   A = 81 , Cx = 0  , Cy = 3
*/
#include <iostream.h>
#include <stdlib.h>
int main()
{
      int X[100],Y[100]; // Arrays to keep points coordinates in the polygon
      int Cx, Cy;        // Centroid coordinates to be computed.
      double A;          // Area of polygon to be computed.
      double TotX, TotY, term;
      int i, j, N;
      cout << "Enter number of edges : ";  cin >> N;

      A = 0; TotX = 0; TotY = 0;

      // Get all point coordinates of polygon:
      for (i=0; i <= N-1; i++)
      {
          cout << "Enter X and Y for " << i << ". point : ";
          cin >> X[i] >> Y[i];
      }

      // Calculate area:
      for (i=0; i <= N-1; i++)
      {
          if (i == N-1)
             j = 0;
          else
             j = i+1;

          term = (X[i]*Y[j]) - (X[j]*Y[i]);
          A = A + term;
          TotX += (X[i]+X[j]) * term;
          TotY += (Y[i]+Y[j]) * term;
      }
      A = A / 2;

      // Calculate centroid coordinates:
      Cx = TotX /(6*A);
      Cy = TotY /(6*A);

      cout << "AREA= " << fabs(A) << "  Cx= " << Cx << "  Cy= " << Cy << "\n\n";
      system("PAUSE");
      return 0;
}
```