6.33

```
int main()
{ double x[] = { 2.2, 3.3, 4.4, 5.5, 6.6, 7.7, 8.8, 9.9 };
  int n=8;
  print(x,n);
  double m = mean(x,n);
  double s = stdev(x,n);
  cout << "mean = " << m << ", std dev = " << s << endl;
  for (int i=0; i<n; i++)
    cout << "x[" << i << "] = " << x[i]
         << ", z[" << i << "] = " << (x[i] - m)/s << endl;
}
```

6.34

```
int main()
{ double x[] = { 2.5, 4.5, 6.3, 6.7, 7.2, 7.5, 7.8, 9.9 };
  int n=8;
  print(x,n);
  double m = mean(x,n);
  double s = stdev(x,n);
  cout << "mean = " << m << ", std dev = " << s << endl;
  for (int i=0; i<n; i++)
  { double z = (x[i] - m)/s;
    cout << "x[" << i << "] = " << x[i]
         << ", z[" << i << "] = " << z;
    if (z >= 1.5) cout << " = A" << endl;
    else if (z >= 0.5) cout << " = B" << endl;
    else if (z >= -0.5) cout << " = C" << endl;
    else if (z >= -1.5) cout << " = D" << endl;
    else  cout << " = F" << endl;
  }
}
```

6.35

```
void build_pascal(int p[][SIZE], int n)
{ assert(n > 0 && n < SIZE);
  for (int i=0; i<SIZE; i++)
    for (int j=0; j<SIZE; j++)
      if (i>n || j>i) p[i][j] = 0;
      else if (j==0 || j==i) p[i][j] = 1;
      else p[i][j] = p[i-1][j-1] + p[i-1][j];
}
```

6.36

```
double max_of_col(Matrix m, int n, int j)
{ double max=m[0][j];
  for (int i=1; i<n; i++)
    if (m[i][j]>max) max = m[i][j];
  return max;
}
double minimax(Matrix m, int n)
{ assert(n>0 && n < SIZE);
  double minimax=max_of_col(m,n,0);
  for (int j=1; j<n; j++)
  { double mm = max_of_col(m,n,j);
    if (mm<minimax) minimax = mm;
  }
  return minimax;
}
```