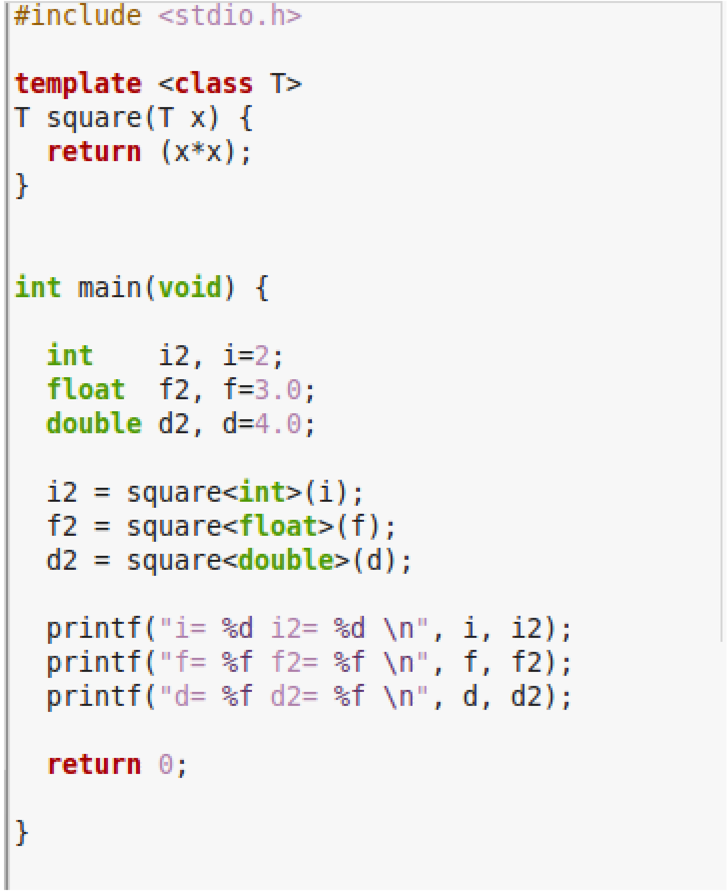
Sorting (4A)

Young Won Lim
5/20/15

Please send corrections (or suggestions) to youngwlim@hotmail.com.

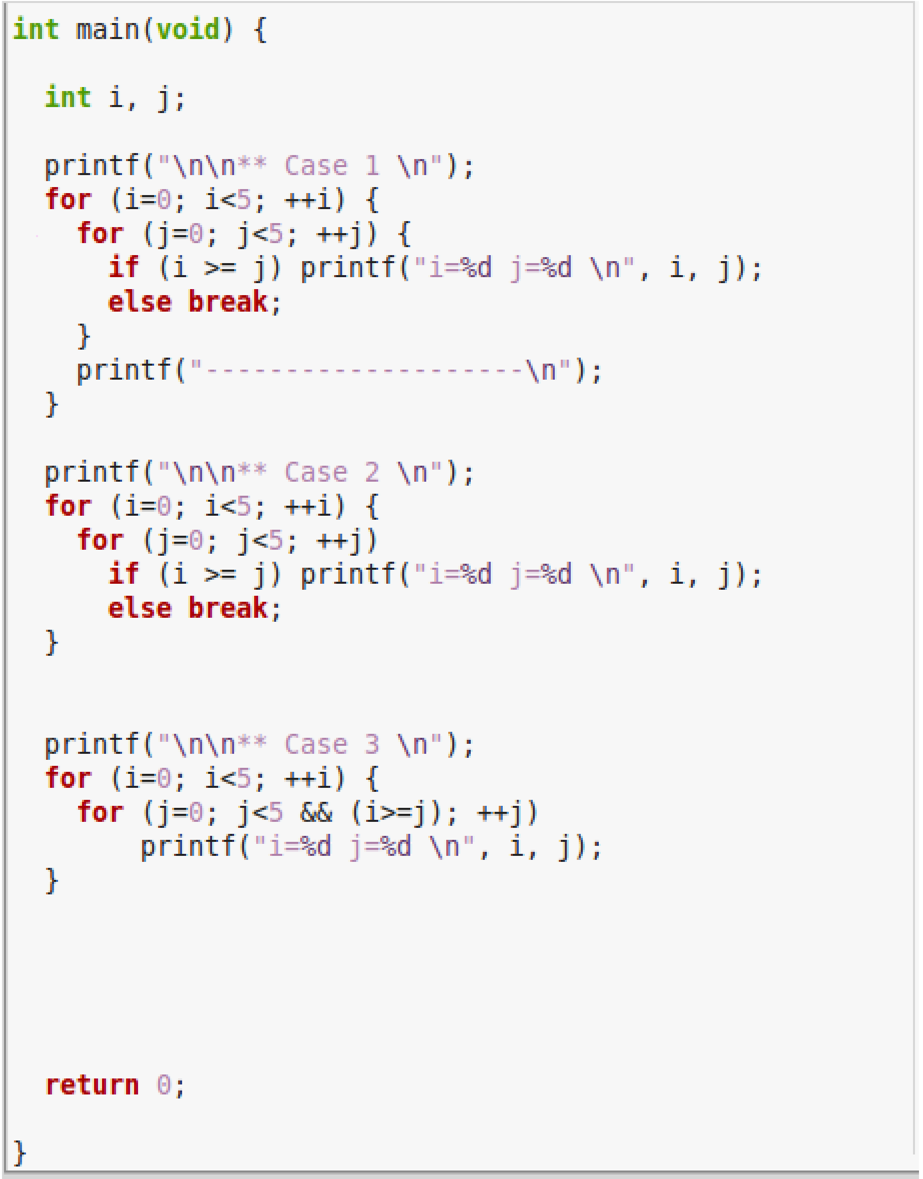
This document was produced by using OpenOffice and Octave.

Template



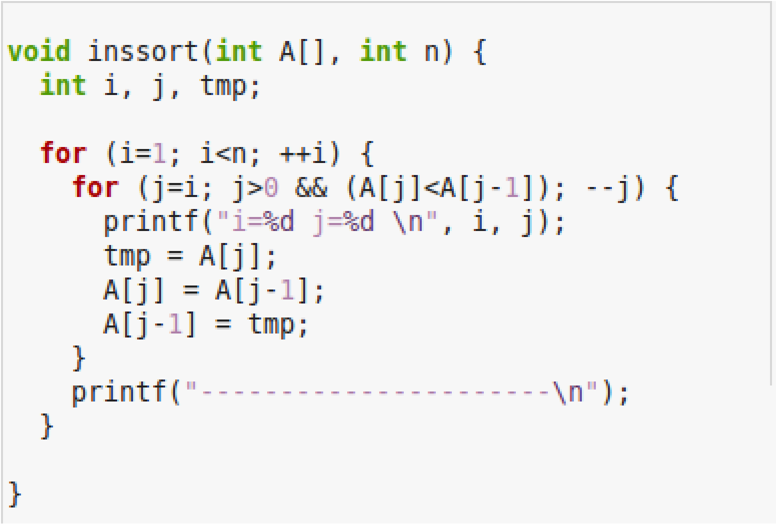
```
#include <stdio.h>

template <class T>
T square(T x) {
  return (x*x);
}


int main(void) {

  int    i2, i=2;
  float  f2, f=3.0;
  double d2, d=4.0;

  i2 = square<int>(i);
  f2 = square<float>(f);
  d2 = square<double>(d);

  printf("i= %d i2= %d \n", i, i2);
  printf("f= %f f2= %f \n", f, f2);
  printf("d= %f d2= %f \n", d, d2);

  return 0;

}
```

Loop Conditions & Break



```
int main(void) {

  int i, j;

  printf("\n\n** Case 1 \n");
  for (i=0; i<5; ++i) {
    for (j=0; j<5; ++j) {
      if (i >= j) printf("i=%d j=%d \n", i, j);
      else break;
    }
    printf("--------------------\n");
  }

  printf("\n\n** Case 2 \n");
  for (i=0; i<5; ++i) {
    for (j=0; j<5; ++j)
      if (i >= j) printf("i=%d j=%d \n", i, j);
      else break;
  }


  printf("\n\n** Case 3 \n");
  for (i=0; i<5; ++i) {
    for (j=0; j<5 && (i>=j); ++j)
        printf("i=%d j=%d \n", i, j);
  }




  return 0;

}
```

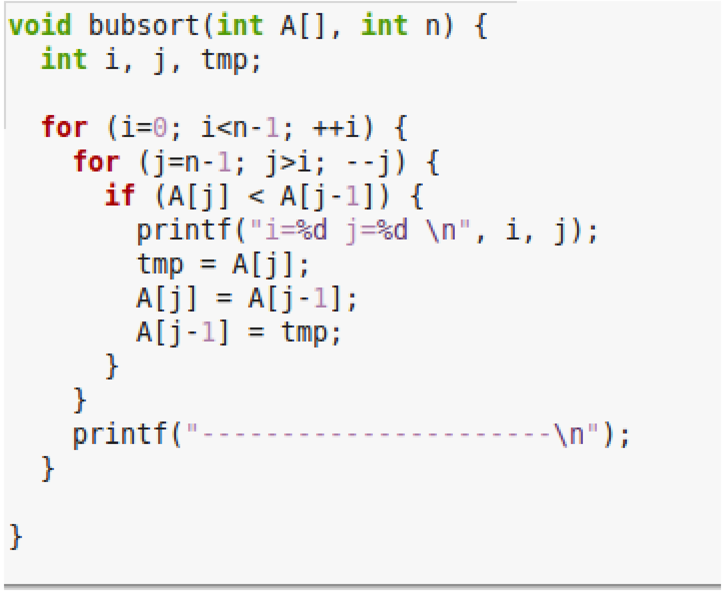
Insertion Sort



```
void inssort(int A[], int n) {
  int i, j, tmp;

  for (i=1; i<n; ++i) {
    for (j=i; j>0 && (A[j]<A[j-1]); --j) {
      printf("i=%d j=%d \n", i, j);
      tmp = A[j];
      A[j] = A[j-1];
      A[j-1] = tmp;
    }
    printf("----------------------\n");
  }

}
```

http://people.cs.vt.edu/shaffer/Book/C++3elatest.pdf

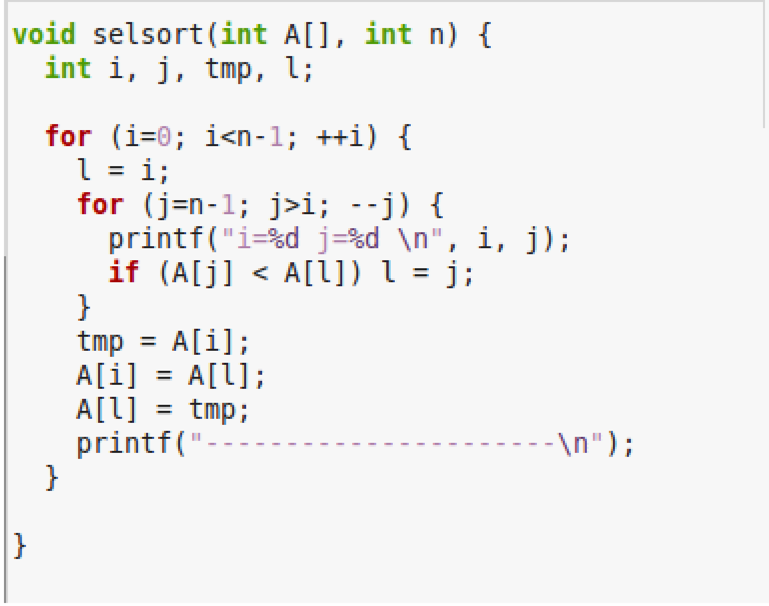
Bubble Sort



```c
void bubsort(int A[], int n) {
  int i, j, tmp;

  for (i=0; i<n-1; ++i) {
    for (j=n-1; j>i; --j) {
      if (A[j] < A[j-1]) {
        printf("i=%d j=%d \n", i, j);
        tmp = A[j];
        A[j] = A[j-1];
        A[j-1] = tmp;
      }
    }
    printf("----------------------\n");
  }

}
```

http://people.cs.vt.edu/shaffer/Book/C++3elatest.pdf

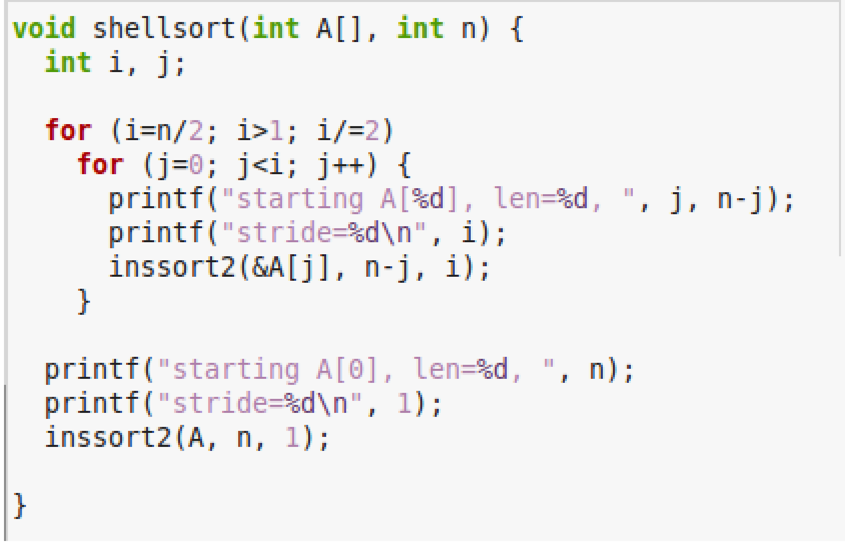
Selection Sort



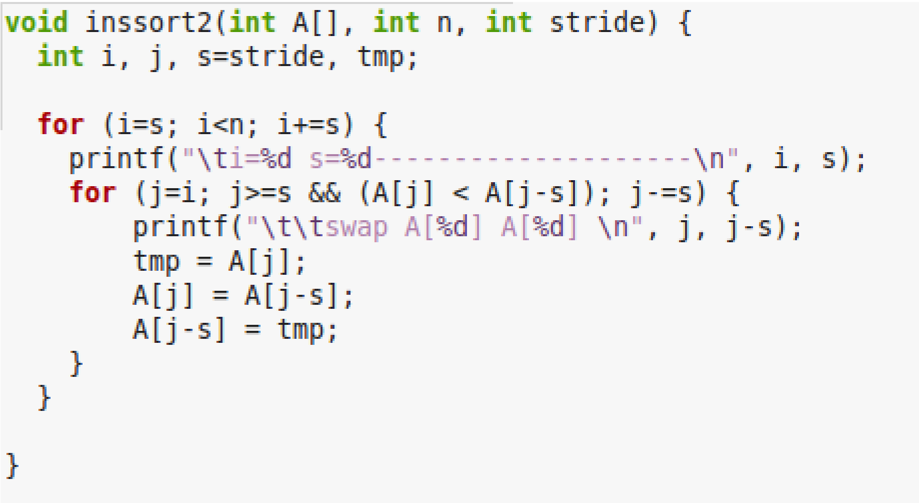
```
void selsort(int A[], int n) {
  int i, j, tmp, l;

  for (i=0; i<n-1; ++i) {
    l = i;
    for (j=n-1; j>i; --j) {
      printf("i=%d j=%d \n", i, j);
      if (A[j] < A[l]) l = j;
    }
    tmp = A[i];
    A[i] = A[l];
    A[l] = tmp;
    printf("----------------------\n");
  }

}
```

http://people.cs.vt.edu/shaffer/Book/C++3elatest.pdf

Shell Sort



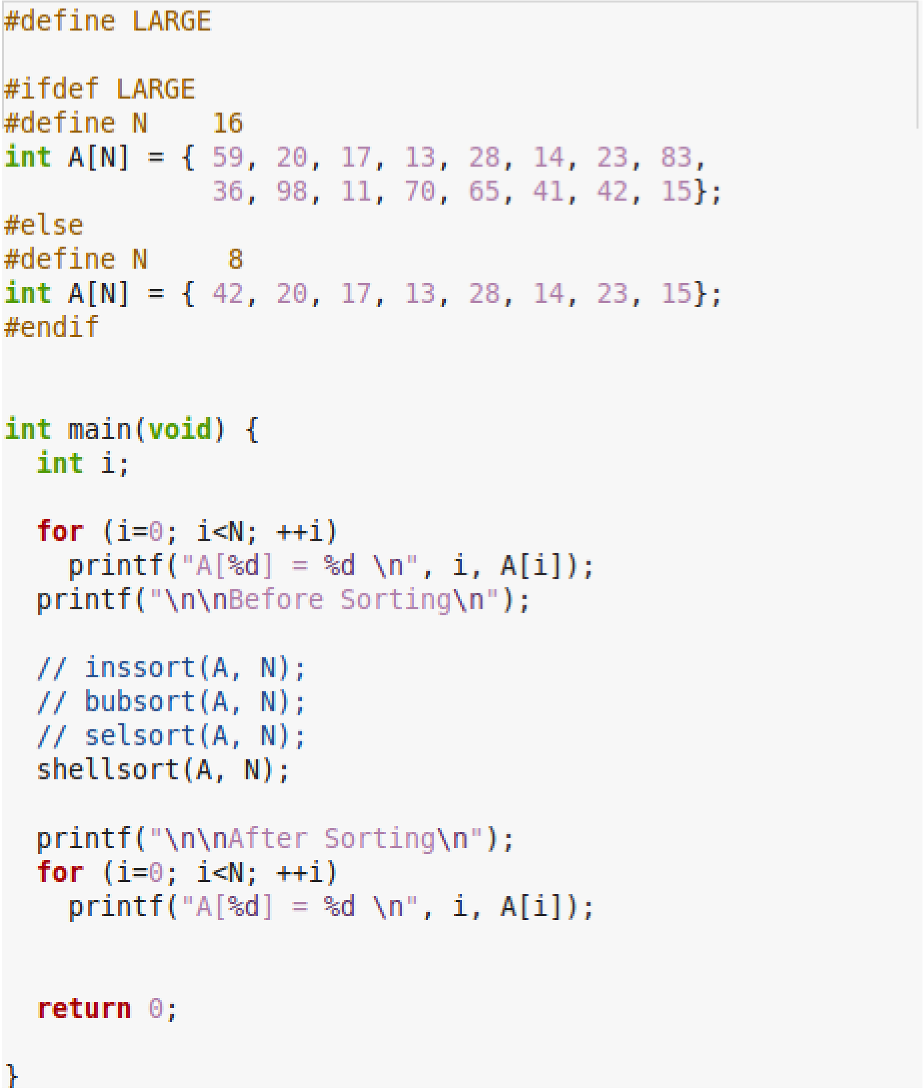
```
void shellsort(int A[], int n) {
  int i, j;

  for (i=n/2; i>1; i/=2)
    for (j=0; j<i; j++) {
      printf("starting A[%d], len=%d, ", j, n-j);
      printf("stride=%d\n", i);
      inssort2(&A[j], n-j, i);
    }

  printf("starting A[0], len=%d, ", n);
  printf("stride=%d\n", 1);
  inssort2(A, n, 1);

}
```



```
void inssort2(int A[], int n, int stride) {
  int i, j, s=stride, tmp;

  for (i=s; i<n; i+=s) {
    printf("\ti=%d s=%d--------------------\n", i, s);
    for (j=i; j>=s && (A[j] < A[j-s]); j-=s) {
        printf("\t\tswap A[%d] A[%d] \n", j, j-s);
        tmp = A[j];
        A[j] = A[j-s];
        A[j-s] = tmp;
    }
  }

}
```

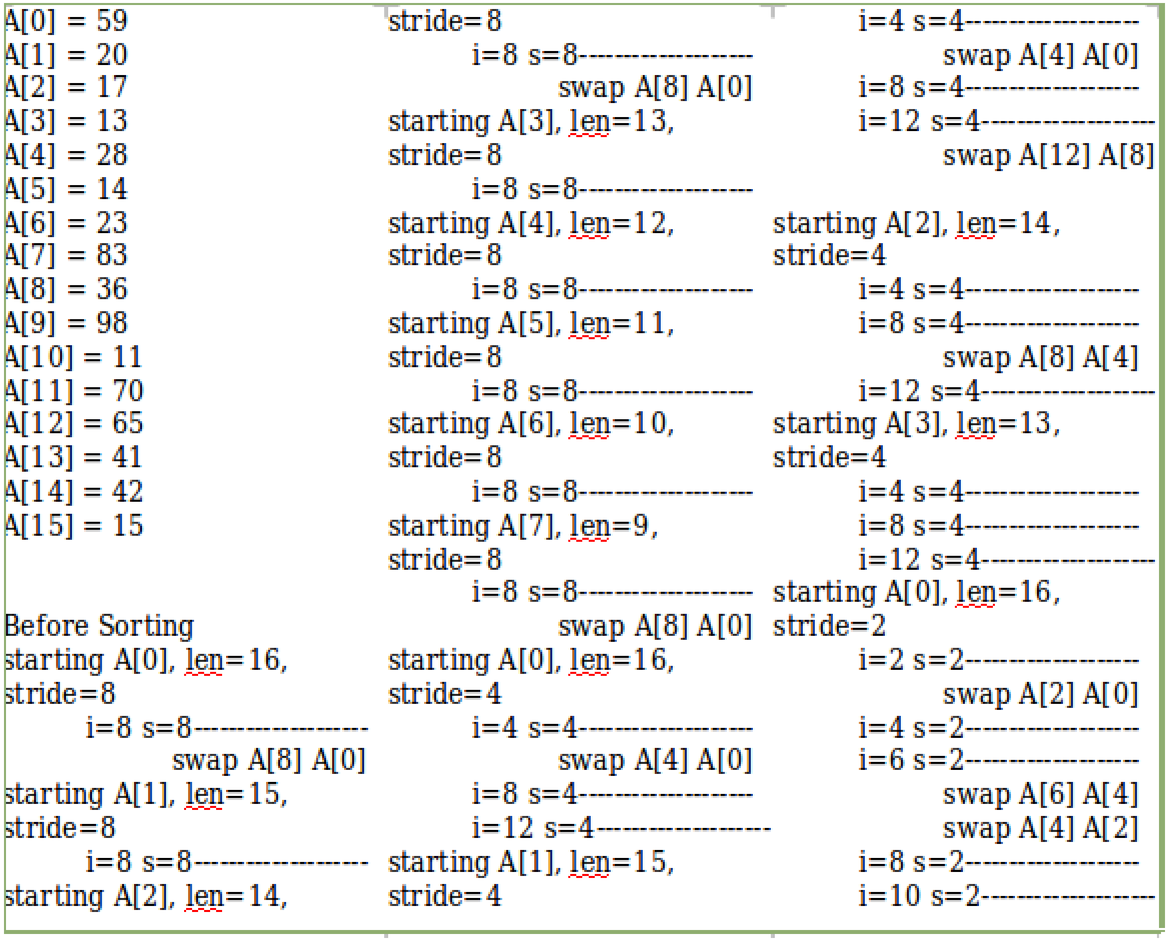
http://people.cs.vt.edu/shaffer/Book/C++3elatest.pdf

main function



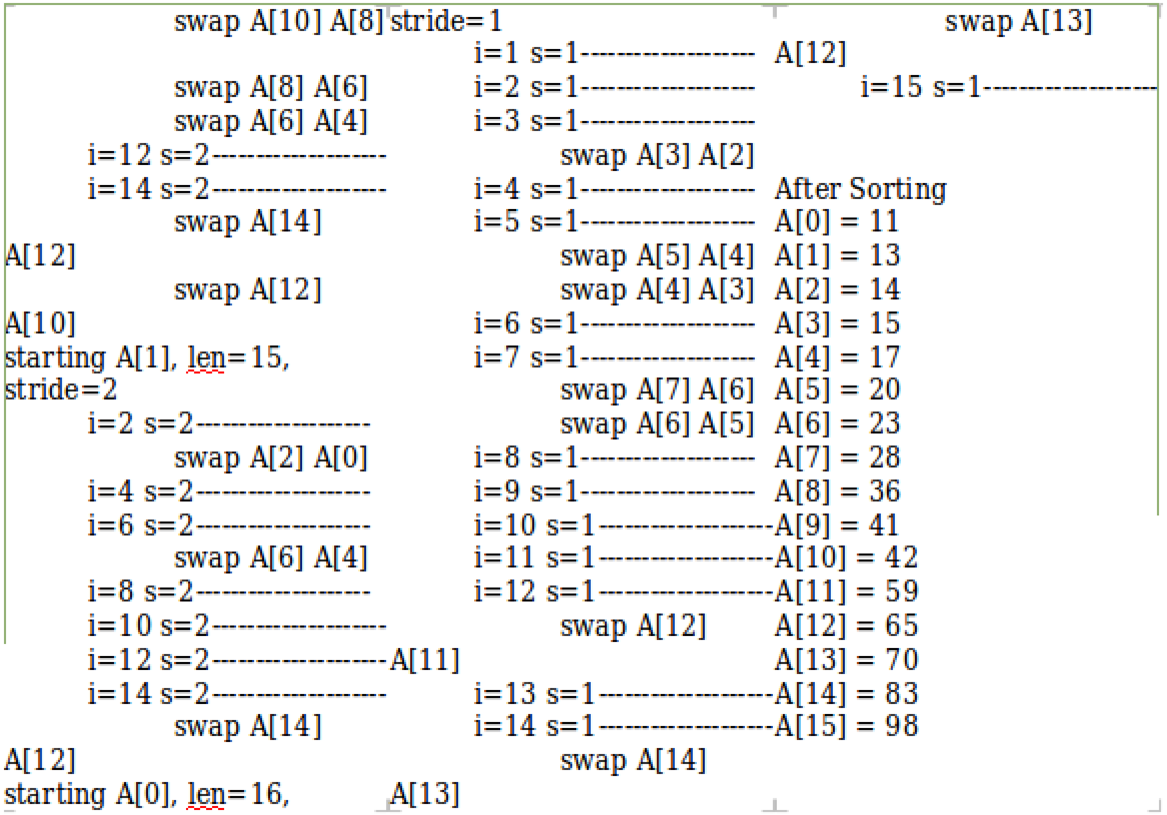
```
#define LARGE

#ifdef LARGE
#define N    16
int A[N] = { 59, 20, 17, 13, 28, 14, 23, 83,
             36, 98, 11, 70, 65, 41, 42, 15};
#else
#define N     8
int A[N] = { 42, 20, 17, 13, 28, 14, 23, 15};
#endif


int main(void) {
  int i;

  for (i=0; i<N; ++i)
    printf("A[%d] = %d \n", i, A[i]);
  printf("\n\nBefore Sorting\n");

  // inssort(A, N);
  // bubsort(A, N);
  // selsort(A, N);
  shellsort(A, N);

  printf("\n\nAfter Sorting\n");
  for (i=0; i<N; ++i)
    printf("A[%d] = %d \n", i, A[i]);


  return 0;

}
```

Shell Sort Result



```
A[0] = 59
A[1] = 20
A[2] = 17
A[3] = 13
A[4] = 28
A[5] = 14
A[6] = 23
A[7] = 83
A[8] = 36
A[9] = 98
A[10] = 11
A[11] = 70
A[12] = 65
A[13] = 41
A[14] = 42
A[15] = 15


Before Sorting
starting A[0], len=16,
stride=8
        i=8 s=8------------------
                swap A[8] A[0]
starting A[1], len=15,
stride=8
        i=8 s=8------------------
starting A[2], len=14,
stride=8
        i=8 s=8------------------
                swap A[8] A[0]
starting A[3], len=13,
stride=8
        i=8 s=8------------------
starting A[4], len=12,
stride=8
        i=8 s=8------------------
starting A[5], len=11,
stride=8
        i=8 s=8------------------
starting A[6], len=10,
stride=8
        i=8 s=8------------------
starting A[7], len=9,
stride=8
        i=8 s=8------------------
                swap A[8] A[0]
starting A[0], len=16,
stride=4
        i=4 s=4------------------
                swap A[4] A[0]
        i=8 s=4------------------
        i=12 s=4------------------
starting A[1], len=15,
stride=4
        i=4 s=4------------------
                swap A[4] A[0]
        i=8 s=4------------------
        i=12 s=4------------------
                swap A[12] A[8]

starting A[2], len=14,
stride=4
        i=4 s=4------------------
        i=8 s=4------------------
                swap A[8] A[4]
        i=12 s=4------------------
starting A[3], len=13,
stride=4
        i=4 s=4------------------
        i=8 s=4------------------
        i=12 s=4------------------
starting A[0], len=16,
stride=2
        i=2 s=2------------------
                swap A[2] A[0]
        i=4 s=2------------------
        i=6 s=2------------------
                swap A[6] A[4]
                swap A[4] A[2]
        i=8 s=2------------------
        i=10 s=2------------------
```

Shell Sort Result



```
          swap A[10] A[8]
          swap A[8] A[6]
          swap A[6] A[4]
     i=12 s=2-----------------
     i=14 s=2-----------------
          swap A[14]
A[12]
          swap A[12]
A[10]
starting A[1], len=15,
stride=2
     i=2 s=2-----------------
          swap A[2] A[0]
     i=4 s=2-----------------
     i=6 s=2-----------------
          swap A[6] A[4]
     i=8 s=2-----------------
     i=10 s=2-----------------
     i=12 s=2-----------------
     i=14 s=2-----------------
          swap A[14]
A[12]
starting A[0], len=16,
stride=1
     i=1 s=1-----------------
     i=2 s=1-----------------
     i=3 s=1-----------------
          swap A[3] A[2]
     i=4 s=1-----------------
     i=5 s=1-----------------
          swap A[5] A[4]
          swap A[4] A[3]
     i=6 s=1-----------------
     i=7 s=1-----------------
          swap A[7] A[6]
          swap A[6] A[5]
     i=8 s=1-----------------
     i=9 s=1-----------------
     i=10 s=1-----------------
     i=11 s=1-----------------
     i=12 s=1-----------------
          swap A[12]
A[11]
     i=13 s=1-----------------
     i=14 s=1-----------------
          swap A[14]
A[13]
          swap A[13]
A[12]
     i=15 s=1-----------------

After Sorting
A[0] = 11
A[1] = 13
A[2] = 14
A[3] = 15
A[4] = 17
A[5] = 20
A[6] = 23
A[7] = 28
A[8] = 36
A[9] = 41
A[10] = 42
A[11] = 59
A[12] = 65
A[13] = 70
A[14] = 83
A[15] = 98
```

main function

References

[1] http://en.wikipedia.org/
[2] http://people.cs.vt.edu/shaffer/Book/C++3elatest.pdf