11.

```
int main()
{
   int a[5], i;

   for (i = 0; i < 5; i++)
      a[i] = i * i;

   for (i = 0; i < 5; i++)
      printf("a[%d] = %d\n", i, a[i]);

   for (i = 1; i < 5; i++)
      a[i] += i;

   for (i=0; i<5; i++)
      printf("a[%d] = %d\n", i, a[i]);
   }
}
```

w/ pragams:

```
int main()
{
   int a[5], i;
      #pragma omp parallel
      {
          #pragma omp for
          for (i = 0; i < 5; i++)
             a[i] = i * i;

             #pragma omp master
             for (i = 0; i < 5; i++)
                printf("a[%d] = %d\n", i, a[i]);

             #pragma omp for
             for (i = 1; i < 5; i++)
                a[i] += i;

             #pragma omp master
             for (i=0; i<5; i++)
```

```
                printf("a[%d] = %d\n", i, a[i]);
        }
}
```

Gives the following output:

```
a[0] = 0
a[1] = 1
a[2] = 6
a[3] = 12
a[4] = 20
a[0] = 0
a[1] = 2
a[2] = 6
a[3] = 12
a[4] = 20
```

12.

```
#include <stdio.h>
#include <omp.h>

int main()
{
   int i, maxv, minv, sum;
   int a[10];

   for(i=0; i<10; i++)
   {
      a[i] = rand()%1000;
      printf("%d\n", a[i]);
   }

   maxv = a[0]; minv = a[0]; sum = 0;

   for(i=0; i<10; i++)
   {
      if (maxv<a[i])
         maxv = a[i];
      if (minv>a[i])
         minv = a[i];

      sum += a[i];
   }

   printf("max:%d\n", maxv);
   printf("min:%d\n", minv);
   printf("sum:%d\n", sum);

   return 0;
}
```

13.

```
#include <stdio.h>
#include <omp.h>

int main()
{
   omp_set_num_threads(4);

   int i = 0;
   #pragma omp parallel
   {
      #pragma omp single
      {
         #pragma omp task
         {
               printf("Am I tasking?...\n");
         }
      }
   }
}
```

14.

```
#include <stdio.h>
#include <omp.h>
#include <math.h>

int main()
{
   int n=0, nMax = 1000;
   float x=0, y, tempY;

   for(n=0; n<nMax; n++)
   {
      tempY =  sin(x)/(x+1);
      y = y + tempY*tempY;
      x += .01;
   }

   printf("Final Y:%f\n", y);

}
```