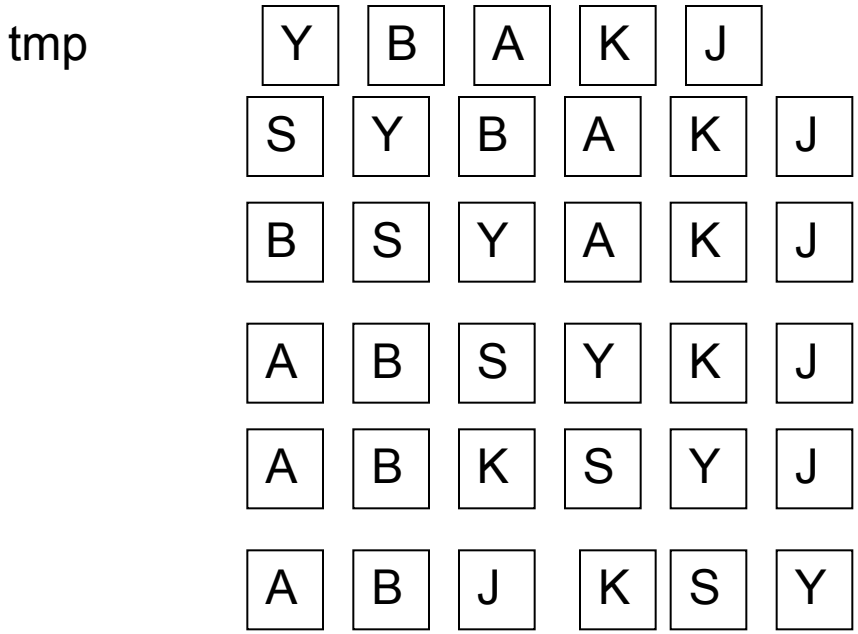
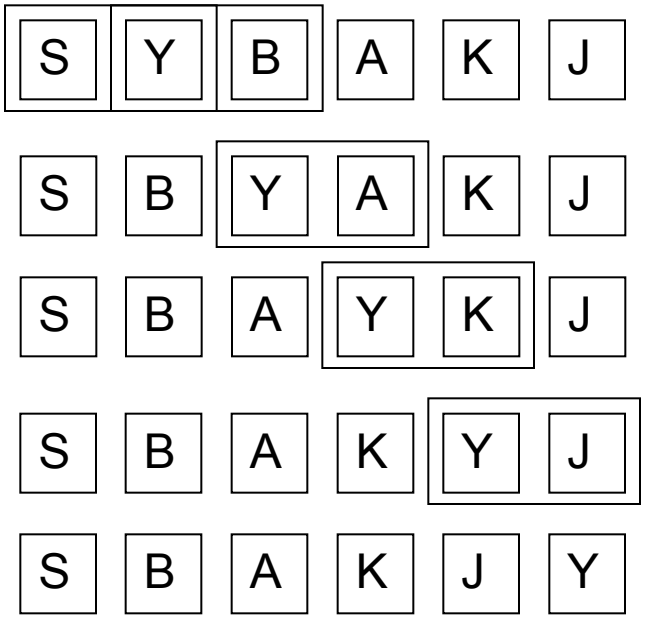
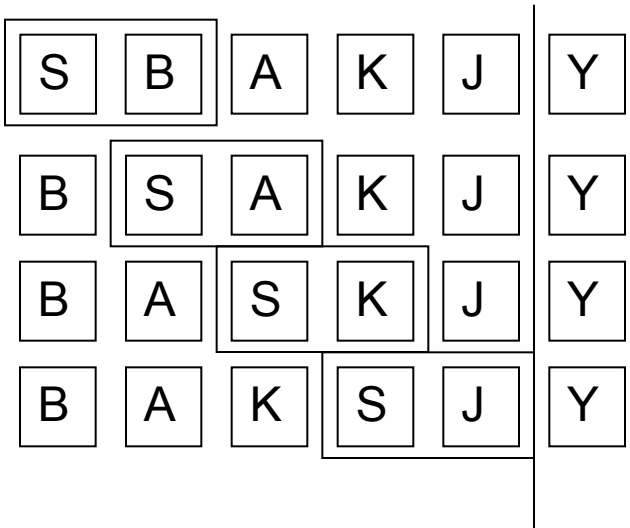
Сортировка Вставками

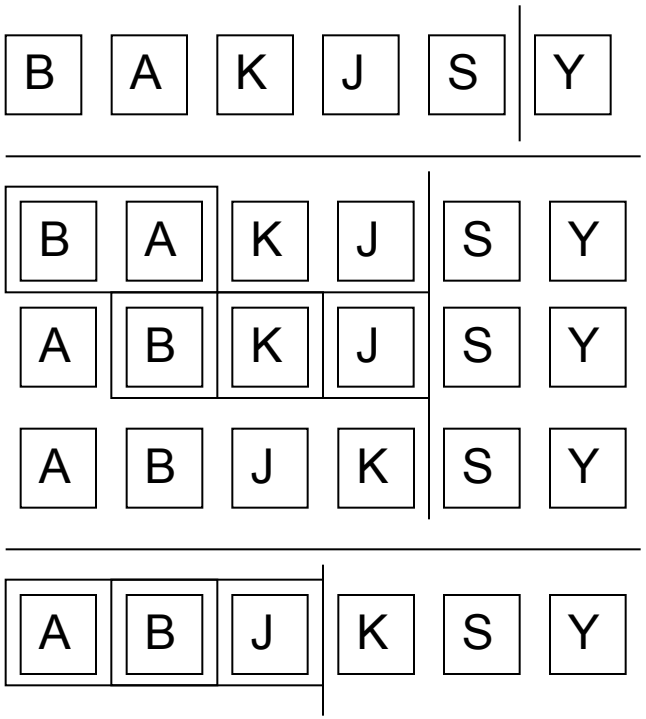


```
public void InsertSort()
      {  int i, j;
         int tmp;
         bool eoc;
         for (i=1;i<N;i++)
         {  tmp=A[i];
            j=i-1;
            eoc=false;
            while (!eoc)
            {  if (j < 0)
                  eoc = true;
               else
                  if (A[j] < tmp)
                     eoc = true;
                  else
                  {  A[j+1]=A[j];
                     j=j-1;
                  }
            }
            A[j+1]=tmp;
         }
      }
```

Сортировка Пузырек







```
public void BubbleSort(int h)
{  int i, j, Margin;
   int tmp;
   bool Finish;
   Margin = N - h-1;
   j = Margin;
   do
   {  Finish=true;
      for (i = 0; i <= Margin; i++)
      {   if (A[i] > A[i + h])
          {   tmp = A[i];
              A[i] = A[i + h];
              A[i + h] = tmp;
              j = i;
              Finish = false;
          }
      }
      Margin = j - h;
   }
```

```
public void Shell()
    {
        int h;
        h = N / 2;
        while (h>0)
        {
            BubbleSort(h);
            h = h / 2;
        }
    }
```

```
public int DSearch(string _SI)
    {  int left, middle, right;
       int SI;
       if (Int32.TryParse(_SI, out SI))
       {  left = 0;
          right = N - 1;
          while (left < right)
          {  middle = (left + right) / 2;
             if (A[middle] < SI)
                left = middle + 1;
             else
                right = middle;
          }
          if (A[right] == SI)
             return right;
          else
             return -1;
       }
       return -1;
    }
```