Ödev #2
Sıralama Algoritması

Ödevin detayları:

Bellekte kendi belirlediğiniz bir alana 10 tane birer byte'lık rastgele işaretsiz sayı yerleştirip bunları küçükten büyüye sıralayıp yine aynı bellek bölgesinde saklayan programı Assembly dilinde yazınız.

Not: Yazdığınız programda komutların yanlarına yeterince açıklama satırı ekleyiniz ve sıralanacak verileri bellekte hangi adres aralığında tuttuğunuzu belirtiniz.

İpucu: Aşağıda yüksek seviyeli dilde verilen "Selection Sort" algoritmasından faydalanabilirsiniz.

Ödev teslim şekli:

BTU Moodle sistemi üzerinden bu platformda belirtilen ve bildirilen son tarihten önce ".asm" uzantılı kaynak programı teslim edin. Ödev dosya adı için sistematik bir dosya adı kullanın. Örneğin, BLM312_Odev2_AdSoyad_OgrenciNo.rar, BMB312 dersinin 2. Ödevi için iyi bir örnek dosya adıdır.

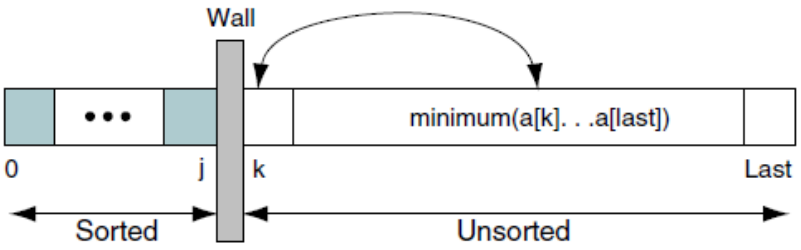

FIGURE 12-3 Selection Sort Concept

ALGORITHM 12-1 Selection Sort

```
Algorithm selectionSort (list, last)
Sorts list array by selecting smallest element in
unsorted portion of array and exchanging it with element
at the beginning of the unsorted list.
   Pre  list must contain at least one item
        last contains index to last element in the list
   Post list has been rearranged smallest to largest
 1 set current to 0
 2 loop (until last element sorted)
   1  set smallest to current
   2  set walker   to current + 1
   3  loop (walker <= last)
      1  if (walker key < smallest key)
         1  set smallest to walker
      2  increment walker
   4  end loop
      Smallest selected: exchange with current element.
   5  exchange (current, smallest)
   6  increment current
 3 end loop
end selectionSort
```

PROGRAM 12-1 Selection Sort

```
 1  /* ================= selectionSort ==================
 2     Sorts list [1…last] by selecting smallest element in
 3     unsorted portion of array and exchanging it with
 4     element at beginning of the unsorted list.
 5        Pre  list must contain at least one item
 6             last contains index to last list element
 7        Post list has been sorted smallest to largest
 8  */
 9  void selectionSort  (int list[ ], int last)
10  {
11  // Local Declarations
12     int smallest;
13     int holdData;
14
15  // Statements
16     for (int current = 0; current < last; current++)
17         {
18          smallest = current;
19          for (int walker = current + 1;
20                   walker <= last;
21                   walker++)
22              if (list[ walker ] < list[ smallest ])
23                  smallest = walker;
24
25          // Smallest selected: exchange with current
26          holdData        = list[ current ];
27          list[current]   = list[ smallest ];
28          list[smallest]  = holdData;
29         } // for current
30     return;
31  }  // selectionSort
```