Praktikum 11.1. SORTING (BUBBLE & SHELL)

1. a. Implementasikan algoritma metode
    a. BUBBLE SORT
    b. SHELL SORT

   dengan data *array of int* berjumlah 10 elemen. Elemen dalam array tidak diinputkan, namun diinisialisasi di awal. Tampilkan hasil per iterasi dan lakukan pengurutan secara ascending.

   b. Tambahkan perhitungan total jumlah pembandingan *(comparison = C)*, jumlah penukaran *(swapping = S)* dan jumlah pergeseran *(movement = M)* pada setiap fungsi pengurutan data yang sudah dibuat dan tampilkan nilai *C, S* dan *M* untuk setiap metode.

2. Buatlah fungsi untuk generate bilangan random. Modifikasi program nomor 1 untuk mengurutkan data dengan jumlah elemen >= 20000 yang digenerate secara random.
   Tambahkan perhitungan waktu komputasinya.
   Tambahkan fungsi mode_urut() untuk memilih menu pengurutan secara ascending ataukah descending.
   Buatlah fungsi generate() untuk mengenerate sejumlah bilangan random. Gunakan fungsi rand() dan srand() yang terletak dalam file header math.h.

3. Tambahkan 2 metode ini ke dalam program menu yang telah dibuat sebelumnya (Insertion dan Selection short).
   Untuk semua metode:
   - Inisialisasi elemen array awal menggunakan fungsi generate random variabel dan juga menampilkan waktu eksekusi.
   - Tanpa menampilkan perhitungan total jumlah pembandingan *(comparison = C)*, jumlah penukaran *(swapping = S)* dan jumlah pergeseran *(movement = M).*

   Catatan:
   Contoh penggunaan fungsi random:

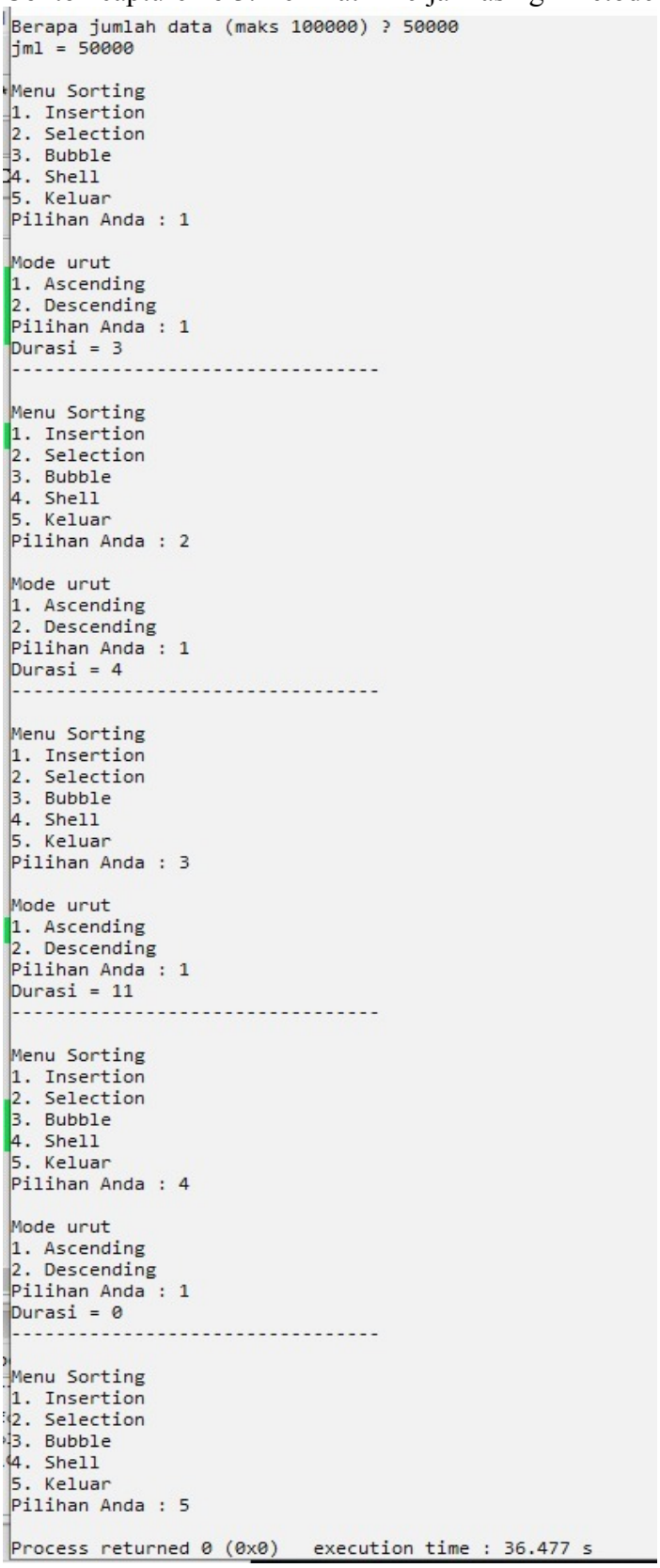
```
#include <math.h>
void generate(int x[]){
      int i;

      for(i=0; i<n; i++)
            x[i] = rand()/1000;
}
```

   sebelum panggil fungsi generate(), lakukan reset waktu
   srand(time(NULL));

Contoh penggunaan fungsi time:

```
int bubblesort(int a[]);
int random(int a[]);
int jum, pil2;
time_t t1, t2;
long waktukomputasi;

int main(){
  int i;
  int a[100000];

  puts("PRAKTIKUM BUBBLE SORTING ");
  printf("masukkan jumlah randoman : ");
  scanf("%d", &jum);
  srand(time(NULL));
  random(a);

  puts("data awal array : ");
  for(i = 0; i < jum; i++){
    printf("%d ", a[i]);
  }
  printf("\n");

  printf("\npengurutan (1)ascending atau (2)descending ? ");
  scanf("%d", &pil2);
  time(&t1);
  bubblesort(a);
  time(&t2);
  waktukomputasi = t2 - t1;

  printf("\n");
  printf("waktu komputasi = %g\n", waktukomputasi);
}
```

Contoh capture no 3. Terlihat kinerja masing2 metode berdasarkan durasi



```
Berapa jumlah data (maks 100000) ? 50000
jml = 50000

Menu Sorting
1. Insertion
2. Selection
3. Bubble
4. Shell
5. Keluar
Pilihan Anda : 1

Mode urut
1. Ascending
2. Descending
Pilihan Anda : 1
Durasi = 3
--------------------------------

Menu Sorting
1. Insertion
2. Selection
3. Bubble
4. Shell
5. Keluar
Pilihan Anda : 2

Mode urut
1. Ascending
2. Descending
Pilihan Anda : 1
Durasi = 4
--------------------------------

Menu Sorting
1. Insertion
2. Selection
3. Bubble
4. Shell
5. Keluar
Pilihan Anda : 3

Mode urut
1. Ascending
2. Descending
Pilihan Anda : 1
Durasi = 11
--------------------------------

Menu Sorting
1. Insertion
2. Selection
3. Bubble
4. Shell
5. Keluar
Pilihan Anda : 4

Mode urut
1. Ascending
2. Descending
Pilihan Anda : 1
Durasi = 0
--------------------------------

Menu Sorting
1. Insertion
2. Selection
3. Bubble
4. Shell
5. Keluar
Pilihan Anda : 5

Process returned 0 (0x0)   execution time : 36.477 s
```