```
#include <stdio.h>
#include <time.h>
#include <windows.h>
#define MAX 10

int data[MAX];

void tukar(int *, int *);
void insertion();
void seleksi();
void tulis();
void bubble();
void shellsort();
void quick(int x[],int,int);
int partition(int x[], int, int);
void merge(int x[],int,int);
void gabung(int x[], int, int, int);


main()
{
      int i;
      long s;
      char pil,jwb;
      do
      {
      printf("Data sebelum diurutkan : \n");
      for(i=0;i<MAX;i++)
      {
            srand(time(&s)*(i+1));
            data[i]=rand()%100;
```

```
      printf("%d ",data[i]);
}
printf("\n\nPilihan : \n");
printf("1. Insertion \n");
printf("2. Seleksi \n");
printf("3. Gelembung\n");
printf("4. Shell\n");
printf("5. Quick\n");
printf("6. Merge\n");
printf("\nMasukkan Pilihan anda : ");
scanf("%d",&pil);
switch (pil){
case 1:
      printf("\nData setelah diurut dengan Insertion :\n");
      insertion();
      break;
case 2:
      printf("\nData setelah diurut dengan Seleksi :\n");
      seleksi();
      break;
case 3:
      printf("\nData setelah diurut dengan Bubble :\n");
      bubble();
      break;
case 4:
      printf("\nData setelah diurut dengan Shell :\n");
      shellsort();
      break;
case 5:
      printf("\nData setelah diurut dengan Quick :\n");
      quick(data,0,MAX-1);
```

```
            tulis();
            break;
      case 6:
            printf("\nData setelah diurut dengan Merge :\n");
            merge(data,0,MAX-1);
            tulis();
            break;
      default:
            printf("Pilihan anda salah. Pilih 1 s/d 6!");

      }
      fflush(stdin);
      printf("\nMau coba lagi?");
      scanf("%c",&jwb);
      }
      while(jwb=='y');

}

void tukar(int *a, int *b)
{
      int temp;
      temp = *a;
      *a = *b;
      *b = temp;
}

void insertion()
{
      int i,j,temp;
      for(i=1;i<MAX;i++)
```

```
	{
		temp=data[i];
		j=i-1;
		while(temp<data[j]&&j>=0)
		{
			data[j+1]=data[j];
			j--;
		}
		data[j+1]=temp;
		tulis();
	}
}

void seleksi()
{
	int i,j,indeks;
	for(i=MAX-1;i>=0;i--)
	{
		indeks=0;
		for(j=0;j<=i;j++)
		{
			if(data[indeks]<=data[j])
				indeks=j;
		}
		if(indeks!=i)
		tukar(&data[i],&data[indeks]);
	}
	tulis();
}
void tulis()
{
```

```
	int i;
	for(i=0;i<MAX;i++)
		printf("%d ",data[i]);
	printf("\n");
}
void bubble()
{
	int i,j;
	time_t ltime1, ltime2;
	long waktu;
	int counter=0;

	unsigned int t1,t2;

	t1=GetTickCount();

	time( &ltime1);
	for (i=MAX; i>0; i--)
		for (j=1; j<i; j++)
		{
			if (data[j-1]>data[j])
			{
				tukar(&data[j-1],&data[j]);
				++counter;
			}

			++counter;
		}
	time( &ltime2);
	t2=GetTickCount();
	waktu = ltime2-ltime1;
```

```
	tulis();
	printf("\nCounter : %d\n\n",counter);
	printf("\nWaktu awal(second) : %ld\n\n",ltime1);
	printf("\nWaktu akhir(second) : %ld\n\n",ltime2);
	printf("\nWaktu awal(ms) : %d\n\n",t1);
	printf("\nWaktu akhir(ms) : %d\n\n",t2);
	printf("\nWaktu : %ld second %d ms\n\n",waktu,t2-t1);
}

void shellsort()
{
	int Jarak, i, j;
	int Sudah;
	Jarak = MAX;
	while(Jarak > 1){
		Jarak = Jarak / 2;
		Sudah = 1;
		while(Sudah){
			Sudah = 0;
			for(j=0; j<MAX-Jarak; j++){
				i = j + Jarak;
				if(data[j] > data[i]){
				tukar(&data[j], &data[i]);
				Sudah = 1;
				}
			}
		}
		tulis();
	}

}
```

```
void QuickSort(int L, int R)
{
      int i,j,x, temp;
      x = data[(L+R)/2];
      i = L;
      j = R;
      while (i<=j){
            while(data[i] < x)
                  i++;
            while(data[j] > x)
                  j--;
            if(i <= j){
                  tukar(&data[j], &data[i]);
                  i++;
                  j--;
            }
      }
      if(L < j)
            QuickSort(data,L,j);
      if(i < R)
            QuickSort(data,i,R);

}

void merge(int x[], int l, int r)
{
      int med;

      if(l < r) {
            med = (l + r) /2;
```

```
            merge(x, l, med);
            merge(x, med+1, r);
            gabung(x, l, med, r);
      }
}


void gabung(int x[], int left, int med, int right)
{
      int kiri1, kanan1, kiri2, kanan2, i, idx;
      int hasil[MAX];

      kiri1 = left;
      kanan1 = med;
      kiri2 = med + 1;
      kanan2 = right;

      i = left;

      while((kiri1 <= kanan1) && (kiri2 <= kanan2)) {
            if(x[kiri1] < x[kiri2]) {
                  hasil[i] = x[kiri1];
                  kiri1++;
            }
            else {
                  hasil[i] = x[kiri2];
                  kiri2++;
            }
            i++;

            /*for(idx=kiri1; idx<=kanan1; idx++)
```

```
            printf("%d ", x[idx]);
        printf("   ");
        for(idx=kiri2; idx<=kanan2; idx++)
            printf("%d ", x[idx]);
        puts("");*/
    }

    while(kiri1 <= kanan1) {
        hasil[i] = x[kiri1];
        kiri1++;
        i++;
    }

    while(kiri2 <= kanan2) {
        hasil[i] = x[kiri2];
        kiri2++;
        i++;
    }

    for(i = left; i <= right; i++)
        x[i] = hasil[i];
}
```