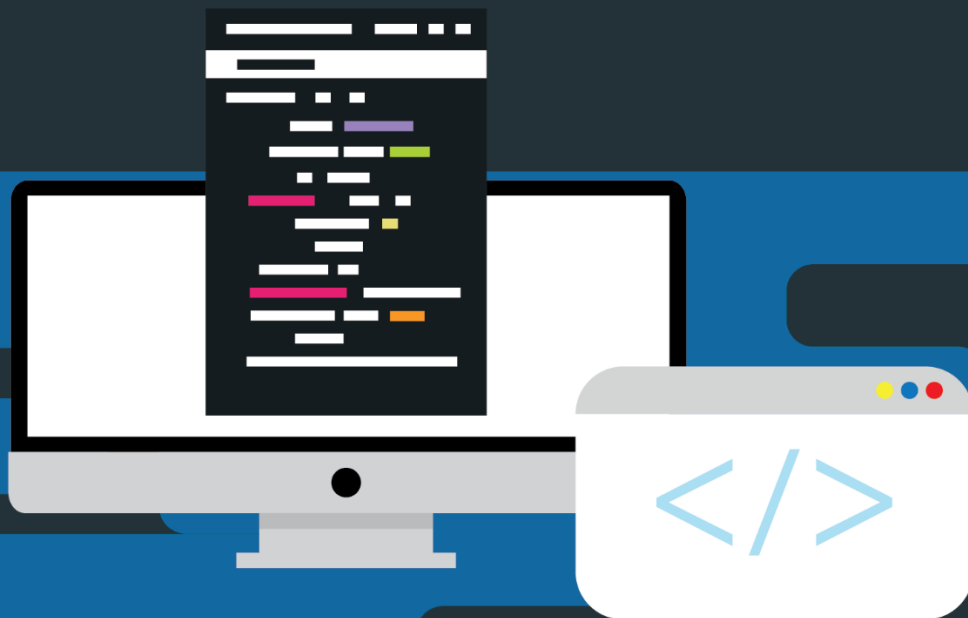


MODUL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II

SORTING
(Insertion Sort & Selection Sort)



TIM ASISTEN PEMROGRAMAN
ANGKATAN 11
ILMU KOMPUTER FPMIPA UPI

INSERTION SORT

Metode ini mengambil sebuah data sisip pada data yang diurutkan dan menggeser data yang lebih besar dari data sisip agar data sisip dapat ditempatkan pada tempat yang benar.

Code :

```
#include <stdio.h>

//Prosedur menampilkan array
void printArray(int n, int arr[])
{
    int i;
    for (i = 0; i < n; i++)
    {
        // menampilkan array sesuai index
        printf("%d ", arr[i]);
    }
    printf("\n");
}

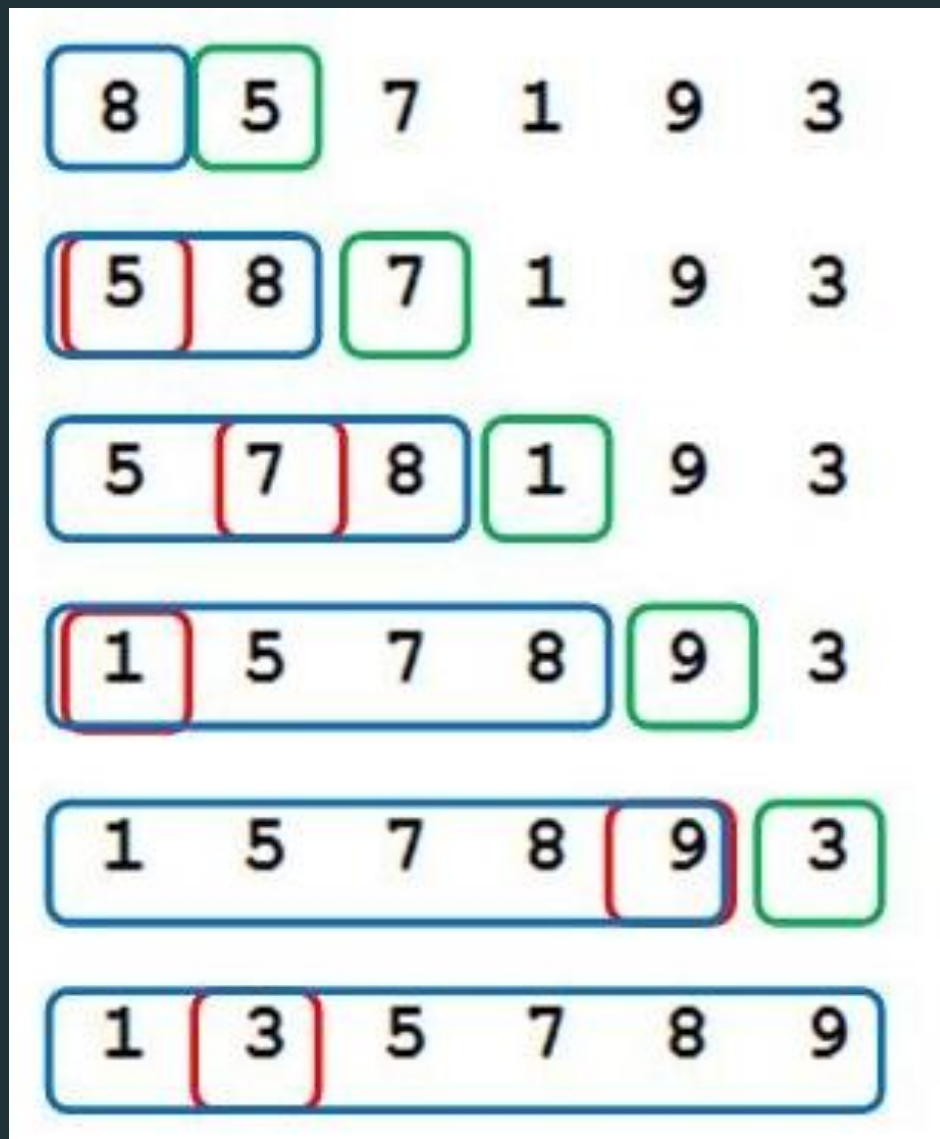
int main()
{
    int i, j, key;
    int array[6] = {8, 5, 7, 1, 9, 3};

    //perulangan sebanyak isi array dimulai dari index ke 1
    for (i = 1; i < 6; i++)
    {
        // menjadikan array ke i sebagai kunci (data sisip)
        key = array[i];
        j = i - 1;
        // perulangan selama j >= 0 karena minimal index array adalah 0
        // dan selama array dengan index j > key
        while (j >= 0 && key < array[j])
        {
            // geser ke depan atau pindah posisi
            array[j + 1] = array[j];
            j--;
        }

        // menempatkan key pada array[j+1]
        array[j + 1] = key;
    }

    printArray(6, array);
    return 0;
}
```

Ilustrasi :



Keterangan :

hijau = key (*current element*)

merah = data yang disisip

biru = sudah terurut (*sorted list*)

SELECTION SORT

Metode ini mencari nilai terkecil atau terbesar bergantung pada pengurutan menaik atau menurun yang kemudian ditempatkan pada tempat paling depan, kemudian mencari lagi nilai terkecil atau terbesar kedua sepanjang jumlah elemen array dikurangi satu, setelah ketemu, elemen kedua ditukar dengan nilai minimum, begitu seterusnya.

Code :

```
#include <stdio.h>

//Prosedur menampilkan array
void printArray(int n, int arr[])
{
    int i;
    for (i = 0; i < n; i++)
    {
        // menampilkan array sesuai index
        printf("%d ", arr[i]);
    }
    printf("\n");
}

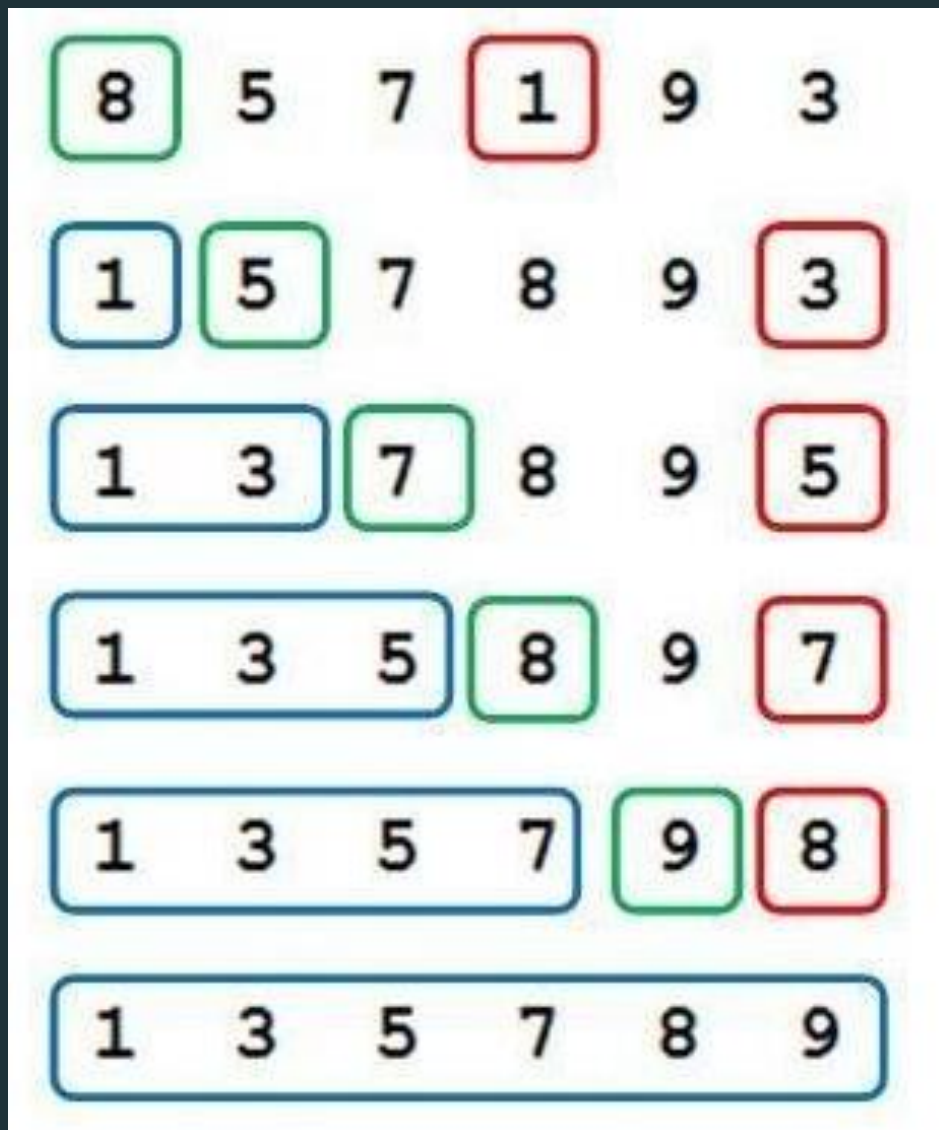
int main()
{
    int i, j, temp, minIdx;
    int array[6] = {8, 5, 7, 1, 9, 3};

    for (i = 0; i < (6 - 1); i++)
    {
        // mencari nilai minimum dari sebuah array
        minIdx = i;
        for (j = i + 1; j < 6; j++)
        {
            if (array[minIdx] > array[j])
            {
                minIdx = j;
            }
        }

        // tukar posisi nilai minimum dengan index i
        temp = array[i];
        array[i] = array[minIdx];
        array[minIdx] = temp;
    }

    printArray(6, array);
    return 0;
}
```

Ilustrasi :



Keterangan :

hijau = minIdx pertama (*current element*)

merah = minIdx terbaru (yang akan diexchange)

biru = sudah terurut (*sorted list*)

LATIHAN

Buatlah program untuk menampilkan output berikut

1. Buatlah sebuah program yang mana inputan berupa nama dan nilai sebanyak n buah. Selanjutnya program akan meminta masukan user berupa char untuk menentukan sorting berdasarkan nama (A) atau nilai (B). Hasil akhir dari program adalah menampilkan output nama dan nilai yang sudah di sorting sesuai keinginan user.

Contoh:

Input :

4

Lulu 95

Mila 99

Koko 70

Alif 81

A

Output :

Alif 81

Koko 70

Lulu 95

Mila 99

Input :

4

Lulu 95

Mila 99

Koko 70

Alif 81

B

Output :

Koko 70

Alif 81
Lulu 95
Mila 99

AKHIR KATA

Penulis mengucapkan terimakasih yang sebesar-besarnya kepada semua elemen yang telah mendukung berjalannya sesi praktikum pada mata kuliah Algoritma dan Pemrograman I kali ini. Semoga apa yang kita dapatkan kali ini bisa menjadi berkah bagi kita semua.

DAFTAR PUSTAKA

Tim Asisten Pemrograman Algoritma dan Pemrograman 1 Angkatan 10. (2019). *Sorting (Insertion Sort & Selection)*. Modul Praktikum Algoritma dan Pemrograman 2. Bandung, Jawa Barat, Indonesia

<https://cheetahonfire.blogspot.com/2009/05/selection-sort-vs-insertion-sort.html>