

Soal Ujian Tengah Semester

Algoritma dan Pemrograman (4 sks)

Dr. Aris Puji Widodo, MT.

Note:

- Kerjakan sendiri dengan JUJUR, jika ditemukan KECURANGAN akan diberikan nilai NOL.
- Kumpulkan dengan format PDF dengan nama file:
UTS_ALPRO_IF_NamaKelas_NIM_Nama.PDF.
- Kirimkan ke mail apwcourses2010@gmail.com dengan subjek email
UTS_ALPRO_IF_NamaKelas_NIM_Nama
- Dikumpulkan ke email Maksimal 09.10 WIB, jika melebihi waktu yang ditentukan diberikan nilai 0.

- Diberikan potongan teks algoritma di bawah ini, berikan penjelasan jalannya algoritma tersebut untuk menghasilkan nilai j pada akhir eksekusi!

```
{Algoritma}
j ← 0 {inisialisasi}
i ← 1
while (i ≤ 5) do
  if ((i MOD 2) = 0) then {genap}
    if ((i MOD 4) = 0) then {kelipatan 4}
      j ← j + i
    else {(i MOD 4) ≠ 0} {tidak kelipatan 4}
      j ← j + (2 * i)
    else {(i MOD 2) ≠ 0} {ganjil}
      j ← j + (j * (i * i))
  i ← i + 1 {increment}
{EndWhile}
output (j)
```

- Diberikan S dan P sebuah bilangan integer sembarang (**S > 0 dan P > 0**) yang dimasukan melalui keyboard. Jika salah satu S atau P bernilai negatif atau nol (**S ≤ 0 atau P ≤ 0**), maka akan ditampilkan pesan "**S dan P harus positif**". Nilai P selalu kurang dari sama dengan S (**P ≤ S**). Untuk nilai P yang selalu kurang dari sama dengan S, lakukan penurunan nilai secara berulang-ulang sampai nilai salah satu S atau P menjadi kurang dari sama dengan 0 (**S ≤ 0 atau P ≤ 0**), maka proses penurunan dihentikan. Adapun untuk proses penurunan nilai S atau P dilakukan dengan ketentuan sebagai berikut:

- Jika P genap, maka dilakukan penurunan nilai S sebesar 1 ($S \leftarrow S - 1$)
- Jika P ganjil, maka dilakukan penurunan nilai S sebesar 2 ($S \leftarrow S - 2$)
- Setiap proses penurunan nilai S dilakukan penurunan nilai P sebesar 2 ($P \leftarrow P - 2$)

Jika nilai P lebih besar S (**P > S**), maka akan ditampilkan pesan "**P harus kurang dari sama dengan S**".

Buatlah teks algoritma untuk menampilkan pada layar, berapa banyak iterasi yang dilakukan untuk menurunkan nilai P atau S sampai menjadi nol atau negatif, untuk setiap nilai integer S dan P sembarang yang dimasukan melalui keyboard.

- Diberikan fungsi **PanjangArr(T)** yang mengirimkan banyaknya elemen pada table T (contoh T= 7 3 9, maka dengan fungsi PanjangArr(T) akan menghasilkan 3). Dengan menggunakan fungsi

PanjangArr(T) di atas, buatlah teks algoritma untuk menyelesaikan prosedur **SplitArray(T,T1,T2)** dengan spesifikasi dibawah ini tanpa harus merealisasikan fungsi PanjangArr(T).

Procedure SplitArr (input: **T**: array [1..N] of integer, output: **T1,T2**: array [1..N] of integer)

{Membagi array T menjadi 2 array baru, yaitu T1 dan T2}

{aturan split, jika panjang array T genap maka banyaknya elemen T1 dan T2 sama. Jika panjang array T ganjil, maka elemen T1 akan lebih banyak 1 elemen dibandingkan T2}

{Contoh: T= 1 2 2 1 --- menghasilkan T1= 1 2 dan T2= 2 1}

{Contoh: T= 1 2 3 2 1 --- menghasilkan T1= 1 2 3 dan T2= 2 1}

{IS: T sembarang}

{FS: menghasilkan array baru T1 dan T2}

{Kamus Lokal}

{Algoritma}

4. Buatlah teks algoritma untuk menyelesaikan fungsi **IsPolindron(T)** dengan spesifikasi dibawah ini.

Functions IsPolindron (**T**: array [1..N] of integer) → boolean

{Mengirimkan true jika elemen array membentuk Polindron, dan false jika tidak membentuk polindron. Elemen table T polindron jika elemen pertama sama dengan yang terakhir, elemen kedua sama dengan elemen terakhir - 1, elemen ketiga sama dengan elemen terakhir - 2, ... dll}

{Contoh: T= 1 2 3 2 1 --- **true**; T= 1 2 3 4 5 --- **false**; T= 3 4 5 6 6 5 4 3 --- **true**}

{Kamus Lokal}

{Algoritma}

<<Selamat Mengerjakan>>