CCH1A4 PRAKTIKUM 6 2019/2020

TUGAS PRAKTIKUM

1. [**pertandingan**]
   Kompetisi pemrograman tingkat nasional berlangsung ketat. Setiap peserta diberikan 8 soal yang harus dapat diselesaikan dalam waktu 5 jam saja. Peserta yang berhasil menyelesaikan soal paling banyak dalam waktu paling singkat adalah pemenangnya.

   [**TERBIMBING**]
   Buat program **gema** yang mencari pemenang dari daftar peserta yang diberikan. Program harus dibuat modular, yaitu dengan membuat prosedur hitungSkor yang mengembalikan total soal dan total skor yang dikerjakan oleh seorang peserta, melalui parameter formal. Pembacaan nama peserta dilakukan di program utama, sedangkan waktu pengerjaan dibaca didalam prosedur.

   ```
   prosedur hitungSkor(output soal, skor : integer)
   ```

   Setiap baris input dimulai dengan satu string nama peserta tersebut diikuti dengan adalah 8 integer yang menyatakan berapa lama (dalam menit) peserta tersebut menyelesaikan soal. Jika tidak berhasil atau tidak mengirimkan jawaban maka otomatis dianggap menyelesaikan dalam waktu 5 jam 1 menit (301 menit).

   Satu baris output berisi nama pemenang, jumlah soal yang diselesaikan, dan nilai yang diperoleh. Nilai adalah total waktu yang dibutuhkan untuk menyelesaikan soal yang berhasil diselesaikan.

   Contoh Input
   Astuti 20 50 301 301 61 71 75 10
   Bertha 25 47 301 26 50 60 65 21
   Selesai

   Contoh Output
   Bertha 7 294

   Keterangan
   Astuti menyelesaikan 6 soal dalam waktu 287 menit, sedangkan Bertha 7 soal dalam waktu 294 menit. Karena Bertha menyelesaikan lebih banyak, maka Bertha menang. Jika keduanya menyelesaikan sama banyak, maka pemenang adalah yang menyelesaikan dengan total waktu paling kecil.

[**MANDIRI**]
Perbaiki program diatas menjadi **gema2** dengan memperbaiki kekurangan yang ada, yaitu:
a. Menguji jika jumlah soal yang diselesaikan sama dan total pengerjaan juga sama, maka harus dinyatakan semua peserta tersebut adalah juara bersama.
Petunjuk: Gunakan operasi string untuk menyimpan semua nama pemenang.
b. Memastikan bahwa total waktu pengerjaan tidak lebih dari 5 jam. Jika ada peserta dimana total waktunya lebih dari 5 jam, maka akan otomatis gugur. Buat fungsi **disq** yang mengembalikan nilai Boolean untuk kebutuhan ini.
c. Memastikan bahwa data waktu adalah valid, yaitu tidak data negatif atau lebih dari 5 jam 1 menit. Buat fungsi **valid** mengembalikan nilai Boolean untuk kebutuhan ini.

2. [**permutasi**]
Permutasi adalah variasi susunan sejumlah obyek dengan memperhatikan urutan/posisi obyek dalam susunan. Contohnya jika diberikan kumpulan obyek {K,P,K}, maka permutasi yang dapat dibuat dari ketiga obyek tersebut ada 3, yaitu {KPK, KKP, PKK}. Jika diketahui kumpulan n obyek terdiri dari k obyek yang berbeda, yang masing-masing mempunyai ada n_1, n_2, ... n_k buah, maka banyaknya cara obyek-obyek tersebut disusun dapat dirumuskan sebagai:

$$P(n;n_1,n_2,\cdots\ n_k) = \frac{n!}{n_1! \times n_2! \times \cdots\ n_k!}$$

Untuk contoh diatas, n=3, k=2, n_1=2, n_2=1, sehingga P(3;2,1)=3!/(2!.1!)=6/2=3.

[**TERBIMBING**]
Buat program **faktorial** yang akan menghitung dan mencetak nilai faktorial dari nilai-nilai yang diberikan. Faktorial dari suatu nilai n adalah f(n)=n.(n-1).(n-2)...1. Contohnya f(5) = 5.4.3.2.1 = 120. Perhitungan faktorial harus dibuat dalam sebuah fungsi **fact** yang mempunyai 1 buah parameter integer, yaitu nilai yang akan dicari faktorialnya.

```
fungsi fact( n : integer ) --> integer
```

Input terdiri dari sejumlah baris. Setiap baris berupa integer positif. Proses harus berhenti setelah jumlah semua data integer dimulai dari baris kedua sama atau lebih besar dari data integer pada baris pertama!

Output adalah nilai-nilai faktorial dari semua integer yang diberikan.

Contoh input:
7
3
2
1
1

Contoh output:
5040
6
2
1
1

Keterangan:
Faktorial dari 7 adalah 7!=7.6.5.4.3.2.1=5040, faktorial dari 3 adalah 3!=3.2.1=6, dst. Proses berhenti karena 7 <= 3+2+1+1.

[**MANDIRI**]
Buat program **permutasi** yang akan menghitung dan mencetak nilai permutasi berdasarkan formula yang diberikan diatas. Perhitungan permutasi harus dibuat dalam sebuah fungsi **perm** yang mempunyai 1 buah parameter integer, yaitu nilai yang akan dicari permutasinya dan fungsi perm tersebut harus menggunakan fungsi **fact** yang sudah dibuat sebelumnya.

```
fungsi perm( n : integer ) --> integer
```

Input terdiri dari sejumlah baris. Setiap baris berupa integer positif. Proses harus berhenti setelah jumlah semua data integer dimulai dari baris kedua sama atau lebih besar dari data integer pada baris pertama!

Output adalah nilai permutasi yang dicari.

Contoh input:
7
3
2
1
1

Contoh output:
Permutasi=420

Keterangan:
n=7, n1=3, n2=2, n3=1, n4=1, maka sesuai formula permutasi yang diberikan, nilainya adalah 420.

Contoh lain input:
7
4
4

Contoh lain output:
Permutasi=35

Keterangan:
n=7, n1=4, n2=3. Karena jumlah sampai dengan data terakhir melebihi n, maka n2 diubah menjadi 3. Output yang diperoleh berasal dari 7!/(4!.3!) = (7.6.5)/(3.2)=35.

3. [**misteri 3n+1**]
   Skiena dan Revilla dalam *Programming Challenges* mendefinisikan sebuah deret bilangan. Deret dimulai dengan sebuah bilangan bulat n. Jika bilangan n saat itu genap, maka suku berikutnya adalah ½n, tetapi jika ganjil maka suku berikutnya bernilai 3n+1. Rumus yang sama digunakan terus menerus untuk mencari suku berikutnya. Deret berakhir ketika suku terakhir bernilai 1. Sebagai contoh jika dimulai dengan n=22, maka deret bilangan yang diperoleh adalah:

   22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

   Untuk suku awal sampai dengan 1000000, diketahui deret ini selalu mencapai suku dengan nilai 1.

   [**TERBIMBING**]
   Buat program **skiena** yang akan mencetak setiap suku dari deret yang dijelaskan diatas untuk nilai suku awal yang diberikan. Pencetakan deret harus dibuat dalam prosedur cetakDeret yang mempunyai 1 parameter formal, yaitu nilai dari suku awal.

```
prosedur cetakDeret( n : integer )
```

   Input berupa satu bilangan integer positif yang lebih kecil dari 1000000.

   Output
   Output terdiri dari satu baris saja. Setiap suku dari deret tersebut dicetak dalam baris yang dan dipisahkan oleh sebuah spasi.

   Contoh Input
   22

   Contoh Output
   22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

   [**MANDIRI**]
   Buat program **revilla** yang akan mencetak jumlah suku dari deret yang dijelaskan diatas untuk nilai suku awal yang diberikan dan suku terbesar yang muncul dalam deret tersebut. Penghitungan jumlah suku dan pencarian suku terbesar dalam deret harus dilakukan dalam prosedur bongkarDeret yang mempunyai 1 parameter input, yaitu nilai dari suku awal, dan 2 parameter output yaitu jumlah suku dan suku terbesar. Pembacaan data dan pencetak output harus dalam program utama.

```
prosedur bongkarDeret(input n:integer, output tot,max: integer)
```

   Input terdiri dari sejumlah baris, masing-masing baris berisi satu bilangan integer positif yang lebih kecil dari 1000000, yaitu nilai suku awal. Input diakhiri dengan bilangan 0.

   Output terdiri dari sejumlah baris yang sama. Setiap baris berisi nilai suku awal, nilai suku maksimum, dan jumlah deret untuk suku awal tersebut.

   Contoh Input
   22
   3
   5
   0

Contoh Output
22 52 16
3 16 8
5 16 6

[**BONUS TANTANGAN**]
Diberikan sebuah angka dengan nilai lebih dari 1000000, periksa apakah deret yang dimulai dengan angka tersebut akan berakhir di suku 1. Buat program **misteri** yang dapat menyelesaikan problem tersebut secara efisien. Anda harus membuat program secara modular, yaitu logik program harus dibuat dalam fungsi dan/atau prosedur.

Input adalah sebuah angka lebih dari 1000000.

Output adalah pesan apakah angka terebut menghasilkan deret yang berakhir di suku 1.

Contoh Input
1000001

Contoh Output
Deret yang diawali 1000001 pasti akan berhenti.