If god=12
    then begin
        create_vacancy(5);
        salary(Job(person[5]))=11;
        person[5]=11;
        outstr("God has just struck down person 5. His job is vacant.", '156'='12);
    end;

If current_vacancy=null ∧ there_is_non_considered_vacancy ∧
    there_is_considered_person
    then begin
        current_vacancy=highest_vacancy_not_considered;
        considered(current_vacancy)mi;
        outstr("picked person "move(current_vacancy)&'156'='12));
    end;

If current_vacancy=null ∧ there_is_non_considered_vacancy
    then begin
        outstr("all vacancies considered so we will quit", '156'='12);
        go to finished;
    end;

If current_vacancy=null ∧ there_is_considered_person
    then begin
        integer Ti=1;
        while (person[Tj]=0 do
            begin
                considered(person[Tj]=1;
                current_person=Tj;
                outstr("new vacancy--unconsider people", '156'='12);
                end;

If current_vacancy=null ∧ current_person=null ∧ there_is_non_considered_person
    then begin
        current_person=highest_person_not_considered;
        considered(current_person)=1;
        outstr("piked person "move(current_person)&'156'='12);
    end;

If current_vacancy=null ∧ current_person=null ∧ there_is_non_considered_person
    then begin
        current_vacancy=null;
        outstr("vacancy cannot be filled", '156'='12);
    end;

If matching ∧ reputation(current_person) < reputation(current_vacancy)
    then begin

"
current_vacancy = null;
outstr("no acceptable people for vacancy" ";'158'12)
end;

if matching ^ job(current_person) != null ^
reputation(current_person) ≥ reputation(current_vacancy) ^
job_attractiveness(current_vacancy) <
appointment_attractiveness(job(current_person))
then begin
  current_person = null;
  outstr("job not attractive to current person" ";'158'12);
end;

if matching ^ job(current_person) != null ^
reputation(current_person) ≥ reputation(current_vacancy) ^
job_attractiveness(current_vacancy) >
potential_attractiveness(job(current_person))
then begin
  integer j;
  create_vacancy(current_person);
  make_appointment;
  current_vacancy = null;
outstr("person takes new job and old one becomes vacancy " ";'158'12);
end;

if matching ^ job(current_person) != null ^
reputation(current_person) ≥ reputation(current_vacancy) ^
job_attractiveness(current_vacancy) ≥
appointment_attractiveness(job(current_person)) ^
job_attractiveness(current_vacancy) ≤
potential_attractiveness(job(current_person))
then begin
  salary(appointment(current_person)) =
max_counter_offer(appointment(current_person));
mmax_counter_offer(appointment(current_person)) =
salary(appointment(current_person)) +
salary(appointment(current_person) +
salary(appointment(current_person) +
salary(appointment(current_person) +
aggressiveness(institution(appointment(current_person)))))/100;
current_person = null;
outstr("persons current salary adjusted upward" ";'158'12);
end;

if matching ^ job(current_person) != null ^
reputation(current_person) ≥ reputation(current_vacancy) ^
then begin
  job(current_person) = make_new_appointment;
  jobs(current_vacancy) = -1;
  current_vacancy = null;
outstr("unemployed person given job" ";'158'12);
end;
outstr(" done with pass through rules");
Swapping to SYS:SAIL.DMP
SAIL: ACADEM, SAIL 1 2
LOADING

LOADER 5K CORE EXECUTION
Pick ed vacancy 3
Pick ed person 5
Unemployed person given job
done with pass through rules
New vacancy—unconsidered people
done with pass through rules
Pick ed vacancy 4
Pick ed person 5
Job not attractive to current person
done with pass through rules
Pick ed person 2
Unemployed person given job
done with pass through rules
New vacancy—unconsidered people
done with pass through rules
Pick ed vacancy 1
Pick ed person 5
Job not attractive to current person
done with pass through rules
Pick ed person 2
Job not attractive to current person
done with pass through rules
Pick ed person 4
Unemployed person given job
done with pass through rules
New vacancy—unconsidered people
done with pass through rules
Pick ed vacancy 2
Pick ed person 5
Job not attractive to current person
done with pass through rules
Pick ed person 2
Job not attractive to current person
done with pass through rules
Pick ed person 4
Persons current salary adjusted upward
done with pass through rules
Pick ed person 3
Unemployed person given job
done with pass through rules
All vacancies considered so we will quit

End of SAIL execution

10
Swapping to SYS: SAIL.DMP
SAIL: ACADEM.SAI 1 2
LOADING

LOADER 5K CORE
EXECUTION
picked vacancy 3
picked person 5
unemployed person given job
done with pass through rules
new vacancy—unconsider people
done with pass through rules
picked vacancy 4
picked person 5
Job not attractive to current person
done with pass through rules
picked person 2
unemployed person given job
done with pass through rules
new vacancy—unconsider people
done with pass through rules
picked vacancy 1
picked person 5
Job not attractive to current person
done with pass through rules
picked person 2
Job not attractive to current person
done with pass through rules
picked person 4
unemployed person given job
done with pass through rules
new vacancy—unconsider people
done with pass through rules
picked vacancy 2
picked person 5
Job not attractive to current person
done with pass through rules
picked person 2
Job not attractive to current person
done with pass through rules
picked person 4
unemployed person given job
done with pass through rules
new vacancy—unconsider people
done with pass through rules
picked vacancy 2
picked person 5
Job not attractive to current person
done with pass through rules
picked person 2
Job not attractive to current person
done with pass through rules
God has just struck down person 5, his job is vacant.
picked person 4
persons current salary adjusted upward
done with pass through rules
picked person 3
unemployed person given job
done with pass through rules
new vacancy—unconsider people
done with pass through rules
picked vacancy 1
picked person 2
persons current salary adjusted upward
done with pass through rules
picked person 4
person takes new job and old one becomes vacancy
done with pass through rules
new vacancy--unconsider people
done with pass through rules
picked vacancy 2
picked person 2
job not attractive to current person
done with pass through rules
picked person 4
job not attractive to current person
done with pass through rules
picked person 3
persons current salary adjusted upward
done with pass through rules
picked person 1
unemployed person given job
done with pass through rules
all vacancies considered so we will cult
End of SAIL execution
*C
begin
label finished;
integer current_vacancy, current_person, cl, c2, max;

comment institutions have 1-budget 2-aggressiveness 3-reputation
4-reference group,
Jobs have 1-salary 2-reputation 3-institution 4-minimum acceptable
reputation 5-considered,
appointments have 1,2,3 as jobs 4-maximum counter offer,
people have 1-age 2-experience 3-ability 4-case 5-school 6-cosmopolitan
7-effort 8-productivity 9-reputation 10-colleagues 11-appointment
12-considered;

integer array institutions[1:20,1:4], persons[1:100,1:12], jobs[1:40,1:53],
appointments[1:100,1:44];

integer procedure job_attractiveness(integer jobname);
return(jobs[jobname,1]+c1*jobs[jobname,2]);

integer procedure person_attractiveness(integer personname);
return(persons[personname,4]+c2*persons[personname,9]);

integer procedure potential_appointment_attractiveness(integer jobname);
return(appointments[jobname,3]+c1*appointments[jobname,7]);

integer procedure appointment_attractiveness(integer jobname);
return(appointments[jobname,13]+c1*appointments[jobname,23]);

comment jobs is the list of vacancies, A -1 entry indicates a blank entry
and a 0 entry indicates the end of the list;

integer procedure highest_vacancy_not_considered;
begin integer max_attractiveness, high_vacancy;
max_attractiveness = 0;
while jobs[i,5]=0 do begin
    if jobs[i,5]=0 and job_attractiveness(i)>max_attractiveness
    then max_attractiveness = job_attractiveness(i);
end;
end;

integer procedure highest_person_not_considered;
begin integer max_attractiveness, high_vacancy;
max_attractiveness = 0;
while persons[i,12]=0 do begin
    if persons[i,12]=0 and person_attractiveness(i)>max_attractiveness
    then max_attractiveness = person_attractiveness(i);
end;
end;

8 Dec 1975 8:26  ACADEM.SAI(CSD.DEN) Page 2-1
begin max_attractiveness=person_attractiveness[i];
end;
end;
i+=1
end;
return(high_vacancy)
end;

procedure make[appointment];
begin integer i;
for i=1 step 1 until 3 do
appointments[person[current_person,11],1]=job[current_vacancy,1];
appointments[person[current_person,11],4]=
appointments[person[current_person,11],1];
appointments[person[current_person,11],1]=
institutions[appointments[person[current_person,11],1]/100];
end;

boolean procedure there_is_non_considered_vacancy;
begin integer i;
i=1;
while true do begin
if jobs[1,1]=0 then return(false);
if jobs[1,2]=0 ∧ jobs[1,1]=1 then return(true);
i+=1
end;
end;

boolean procedure there_is_non_considered_person;
begin integer i;
i=1;
while true do begin
if persons[1,1]=0 then return(false);
if persons[1,1]=1 then return(true);
i+=1
end;
end;

boolean procedure there_is_considered_person;
begin integer i;
i=1;
while true do begin
if persons[1,1]=0 then return(false);
if persons[1,1]=1 then return(true);
i+=1
end;
end;
integer procedure empty_job_slot;
begin
integer i;
if true do begin
    if jobs[i,1]=1 v jobs[i,1]=0 then return(i);
    i+=1
end;
end;

integer procedure empty_appointment_slot;
begin
integer i;
if true do begin
    if appointments[i,1]=1 v appointments[i,1]=0 then return(i);
    i+=1
end;
end;

procedure create_vacancy(integer person);
begin
integer new_job,i,
    new_job=empty_job_slot;
for i=1 step 1 until 3 do
    jobs[new_job,i]=appointments[persons[person,1],i];
end;

boolean procedure matching;
return(current_person<>0 A current_vacancy<>0);

institutions[1,2]=10;
institutions[2,2]=12;
jobs[1,1]=6000;
jobs[1,2]=39;
jobs[1,3]=11;
jobs[1,4]=25;
jobs[2,1]=6000;
jobs[2,2]=49;
jobs[2,3]=21;
jobs[2,4]=40;
jobs[3,1]=8000;
jobs[3,2]=45;
jobs[3,3]=21;
jobs[3,4]=35;
jobs[4,1]=8800;
while true do begin
    god = god + 1;
    if god = 120 then begin
        create_vacancy(5);  
        appointments[persons[5,11],1] = 1; 
        persons[5,3,n] = 1; 
        outstr("God has just struck down person 5, his job is vacant.")
    end;

    if current_vacancy = 5 then begin
        current_vacancy = highest_vacancy_not_considered; 
        jobs[current_vacancy,5] = 1; 
        outstr("Picked vacancy "; 
            S(vacuS(current_vacancy))[15/12]);
    end;

    if current_vacancy = 5 then begin
        outstr("All vacancies considered so we will quit"); 
        go to finished;
    end;

end;
if current_vacancy \& \& there_is_considered_person
then begin integer i = 1;
while persons[i].1 = 0 do
begin persons[i].1 = 1; i = i + 1 end;
current_person = persons[i].1;
outstr("new vacancy--unconsidered people%15a'12");
end;

if current_vacancy \& \& current_person \& \& there_is_non_considered_person
then begin
current_person = highest_person_not_considered;
persons[current_person].1 = i;
outstr("picked person %d\&s(current_person)&'15a'12");
end;

if current_vacancy \& \& current_person \& \& there_is_non_considered_person
then begin
current_vacancy = 0;
outstr("vacancy cannot be filled%15a'12");
end;

if matching \& persons[current_person].9 < jobs[current_vacancy].4]
then begin
current_vacancy = 0;
outstr("no acceptable people for vacancy%15a'12");
end;

if matching \& persons[current_person].11 \&
persons[current_person].9 \& jobs[current_vacancy].4] \&
job_attraction_vacancy[current_vacancy] <
appointment_attraction_vacancy(persons[current_person].11)
then begin
current_person = 0;
outstr("job not attractive to current person%15a'12");
end;

if matching \& persons[current_person].11 \&
persons[current_person].9 \& jobs[current_vacancy].4] \&
job_attraction_vacancy[current_vacancy] >
potential_appointment_attraction_vacancy(persons[current_person].11)
then begin
create_vacancy(current_person);
make_appointment;
current_vacancy = 0;
outstr("person takes new job and old one becomes vacancy%15a'12");
end;
if matching a person[current_person,11]#0 ∧
person[current_person,91 ≥ jobs[current_vacancy,41 ∧
job_attractiveness[current_vacancy] ≥
appointment_attractiveness(persons[current_person,11] ∧
job_attractiveness[current_vacancy] ≤
potential_appointment_attractiveness(persons[current_person,11])
then begin
appointments[persons[current_person,11],11] =
appointments[persons[current_person,11],41;
appointments[persons[current_person,11],4] =
appointments[persons[current_person,11],11] =
appointments[persons[current_person,11],11] =
institutions[appointments[persons[current_person,11],33,2] / 100;
current_person = 0;
outstr("person's current salary adjusted upward" & '158/12);
end;

if matching a person[current_person,11]' #0 ∧
person[current_person,91 ≥ jobs[current_vacancy,41]
then begin integer newapt,1;
newapt = empty; appointment[slot; 
for i = 1 step 1 until 3 do
appointments[newapt,4] = jobs[current_vacancy,11] =
appointments[newapt,4] = appointments[newapt,11] =
institutions[appointments[newapt,33,2] / 100;
jobs[current_vacancy,11] = 1;
current_vacancy = 0;
person[current_person,11] = newapt;
outstr("unemployed person given job" & '158/12);
end;
outstr("done with pass through rules" & '158/12);
end;
finished;
end;