

```

Program ex_3;
type AdresaCandidat = ^Candidat;
  Candidat = record
    NumePrenume : string;
    NotaMedie : real;
    Urm : AdresaCandidat;
  end;
var r,b,v:AdresaCandidat;
    q,b1,v1:AdresaCandidat;
    i:integer;
    n:real;

procedure create();
var input:string;

begin
  while input<>'EXIT' do begin
    writeln("");
    writeln('<| |--- Type EXIT to stop list item creation ---| |>');
    writeln('      PRESS ENTER TO CONTINUE      ');
    readln(input);

    if input<>'EXIT' then begin
      inc(i);
      new(r);
      writeln(i,'# Nume Prenume : '); readln(r^.NumePrenume);
      writeln(i,'# Nota Med : '); readln(r^.NotaMedie);
      r^.Urm:=nil;
      if i = 1 then b:=r else v^.Urm:=r;
      v:=r;
    end;
  end;
end;

procedure display();
var j:integer;
begin
  r:=b;
  writeln('      Current Data :');
  writeln();
  writeln();
  while r<>nil do begin
    writeln(r^.NumePrenume,' : ',r^.NotaMedie);
    r:=r^.Urm;
  end;
end;

procedure display_q();
var j:integer;
begin
  q:=b1;
  writeln('      Current Data :');
  writeln();
  writeln();
  while q<>nil do begin
    writeln(q^.NumePrenume,' : ',q^.NotaMedie);
    q:=q^.Urm;
  end;
end;

```

```

procedure remove();
var name:string;
    save:AdresaCandidat;
begin
    writeln('Nume Student Retragere : '); readln(name);

    r:=b;
    while r<>nil do begin
        if r^.NumePrenume = name then begin
            if r=b then begin
                save:=r;
                r:=r^.Urm;
                dispose(save);
                b:=r;
            end else begin
                v^.Urm:=r^.Urm;
                save:=r^.Urm;
                dispose(r);
                r:=save;
            end;
        end else begin
            v:=r;
            r:=r^.Urm;
        end;
    end;
end;

```

```

procedure remove_grade();
var n:real;
    save:AdresaCandidat;
begin
    writeln('Remove under : '); readln(n);

    r:=b;
    while r<>nil do begin
        if r^.NotaMedie < n then begin
            if r=b then begin
                save:=r;
                r:=r^.Urm;
                dispose(save);
                b:=r;
            end else begin
                v^.Urm:=r^.Urm;
                save:=r^.Urm;
                dispose(r);
                r:=save;
            end;
        end else begin
            v:=r;
            r:=r^.Urm;
        end;
    end;
end;

```

```

procedure create_one();
begin
    writeln('Candidat Nou : ');
    writeln();
    writeln();

```

```

inc(i);
new(r);
  writeln(i, # Nume Prenume : '); readln(r^.NumePrenume);
  writeln(i, # Nota Med : '); readln(r^.NotaMedie);
r^.Urm:=nil;
v^.Urm:=r;
v:=r;
end;

```

```

procedure show(n:real);
begin
  writeln('-----');
  writeln('List Limit : ',n);
  writeln();
  writeln();
  r:=b;
  while r<>nil do begin
    if r^.NotaMedie >= n then writeln(r^.NumePrenume, ' : ',r^.NotaMedie);
    r:=r^.Urm;
  end;
end;

```

```

procedure satisfactory();
begin
  show(7.5);
end;

```

```

procedure new_list_limit(n:real);
var i:integer;
begin
  writeln();
  writeln('New List Limit : ',n);
  writeln();
  r:=b;
  while r<>nil do begin
    if r^.NotaMedie >= n then begin
      inc(i);
      new(q);
      q^.NotaMedie:=r^.NotaMedie;
      q^.NumePrenume:=r^.NumePrenume;
      q^.Urm:=nil;
      if i=1 then b1:=q else v1^.Urm:=q;
      v1:=q;
    end;
    r:=r^.Urm;
  end;
end;

```

```

procedure menu();
var ans:char;
begin
  while ans<>'E' do begin
    writeln();
    writeln();
    writeln('<----->');
    writeln('      M E N U      ');
    writeln('<----->');
    writeln('N - New List Item');
    writeln('R - Remove Based on Name');
  end;
end;

```

```

if ans = 'N' then create_one() else
  if ans = 'R' then remove() else
    if ans = 'G' then remove_grade() else
      if ans = 'L' then begin writeln('New List Grade Limit : '); readln(n); new_list_limit(n); end else
        if ans = 'C' then display_q() else
          if ans = 'S' then begin writeln('Grade Limit : '); readln(n); show(n) end else
            if ans = 'T' then satisfactory() else
              if ans = 'D' then display();
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```
create();
menu();
end.
```

```
{FIX DELETE PROCEDURE}
```