
Latihan Soal

Olimpiade Komputer

Oleh :

dwi sakethi, s.si, m.kom

<http://dwijim.wordpress.com>

dwijim@unila.ac.id

0816 403 432

pengrajin teknologi informasi

LEMBAH (JANGAN) BANJIR
TANJUNG SENENG 2006/2011

1 Kata Pengantar

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ ◉ الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ ◉ إِنَّ الْحَمْدَ لِلَّهِ ◉ نَحْمَدُهُ وَ
نَسْتَعِينُهُ وَنَسْتَغْفِرُهُ ◉ وَنَعُوذُ بِاللَّهِ مِنْ شُرُورِ أَنْفُسِنَا ◉ وَمِنْ سَيِّئَاتِ أَعْمَالِنَا ◉ مَنْ
يَهْدِي اللَّهُ فَلَا مُضِلَّ لَهُ ◉ وَمَنْ يَضِلَّ فَلَا هَادِيَ لَهُ ◉ أَشْهَدُ أَنْ لَا إِلَهَ إِلَّا اللَّهُ ◉
وَ أَشْهَدُ أَنَّ مُحَمَّدًا رَسُولُ اللَّهِ ◉ اللَّهُمَّ صَلِّ عَلَى مُحَمَّدٍ وَصَحَابَتِهِ مِنَ الْمُهَاجِرِينَ وَ
الْأَنْصَارِ وَمَنْ تَبِعَهُمْ إِلَى يَوْمِ الدِّينِ ◉ نَشْكُرُ اللَّهَ تَعَالَى عَلَى نِعْمَتِهِ الْكَبِيرَةِ ◉ وَ
مِنْ نِعْمِهِ عَلَيْنَا مَعْرِفَةَ اللَّهِ وَالَّذِينَ وَالرُّسُولَ وَالِدَّعْوَةَ ◉ وَهَذِهِ الْجَمَاعَةُ الْمُبَارَكَةُ ◉
وَكَذَلِكَ مَعْرِفَةَ أَنْفُسِنَا ◉ نَسْأَلُ اللَّهَ تَعَالَى الْبِرَّ وَالتَّقْوَى وَالْعَافِيَةَ فِي بَدَنِنَا فِي
الْأَهْلِ وَالْأَوْلَادِ ◉

Puji yang sejati hanya untuk Allah Yang Maha Tinggi, puja yang sempurna hanya untuk Allah Yang Maha Kuasa. *Sholawat* dan *salam* semoga senantiasa dilimpahkan kepada tauladan semua yang mengaku merupakan himpunan bagian dari kelompok manusia, Muhammad saw., keluarganya, sahabatnya, pengikutnya dan seluruh muslimin kapan dan dimana *bae*.

Tulisan ini disusun sebagai rangkaian kegiatan ketika membina tim olimpiade komputer yang dimulai dengan pembinaan terhadap SMA Negeri 1 Gading Rejo Lampung Tengah. Untuk itu saya sangat berterima kasih kepada Bapak Jumiran dan anak-anak SMA Negeri 1 Gading Rejo waktu itu, yang mohon maaf sekarang saya sudah lupa lagi ... :-) Kegiatan itu kalau saya tidak salah dilakukan tahun 2004. Kemudian berlanjut membina tim olimpiade komputer SMA Negeri 2 Bandar Lampung pada bulan Agustus 2006.

Tahun 2010 bulan Juni-Juli saya menemani Shofwan (dari SMA Negeri 2 Bandar Lampung) menjadi teman 'ngobrol'. Sambil mencoba mengerjakan soal sedikit demi sedikit yang bisa dikerjakan.

Semoga ini menjadi bagian amal kebaikan yang dapat mendorong kesejahteraan di bumi pertiwi ini ... Sehingga rakyat selalu mendendangkan lagu **Di sini senang, di sana senang**, bukan lagu **Padamu Negeri**.

dwi sakethi
dwijim@unila.ac.id

<http://dwijim.wordpress.com>
hp : 0816 403 432 (nomor cantik ya ...)
pengrajin teknologi informasi

Daftar Isi

| | | |
|----------|---------------------------------------|-----------|
| 1 | Kata Pengantar | i |
| 2 | Reverse | 2 |
| 2.1 | Masukan | 2 |
| 2.2 | Keluaran | 2 |
| 2.3 | Contoh Penyelesaian | 2 |
| 2.4 | Contoh hasilnya | 3 |
| 3 | Menghitung Massa | 4 |
| 3.1 | Format Masukan | 4 |
| 3.2 | Format Keluaran | 5 |
| 3.3 | Contoh penyelesain | 5 |
| 3.4 | Contoh hasil <i>running</i> | 10 |
| 4 | Ekspresi Aljabar | 12 |
| 4.1 | Masukan | 13 |
| 4.2 | Keluaran | 13 |
| 4.3 | Contoh Penyelesaian | 14 |
| 4.4 | Contoh hasil <i>running</i> | 18 |
| 5 | MULTIPALINDROM | 19 |
| 5.1 | Format Masukan | 20 |
| 5.2 | Format Keluaran | 20 |
| 5.3 | Contoh Penyelesaian | 21 |
| 5.4 | Hasil Program | 23 |
| 6 | Menghitung Jumlah Huruf | 26 |
| 6.1 | Contoh Penyelesaian | 26 |
| 6.2 | Contoh Keluaran | 28 |
| 7 | Variasi Masalah Bilangan Prima | 29 |
| 7.1 | Contoh Penyelesain | 29 |
| 7.2 | Contoh Keluaran | 32 |
| 8 | Membuat Simulasi Ular Berjalan | 32 |
| 8.1 | Contoh Penyelesaian | 33 |
| 9 | Mencari Selisih Terbesar | 36 |
| 9.1 | Input | 37 |
| 9.2 | Output | 37 |

| | | |
|-----------|--|-----------|
| 9.3 | Contoh Penyelesaian | 37 |
| 10 | Bilangan Fibonacci | 39 |
| 10.1 | Masukan | 39 |
| 10.2 | Keluaran | 40 |
| 10.3 | Penyelesaian | 40 |
| 11 | Menjumlah Dua Bilangan Panjang | 45 |
| 11.1 | Contoh hasil eksekusi program | 45 |
| 11.2 | Contoh Penyelesaian | 46 |
| 12 | Simulasi Program Kasir | 49 |
| 12.1 | Contoh hasil eksekusi program | 49 |
| 12.2 | Contoh Penyelesaian | 50 |
| 13 | Program Game Menembak Versi Text Base | 58 |
| 13.1 | Contoh Penyelesaian | 58 |

2 Reverse

Wah ... mudah amat ya ... soal olimpiade komputer. Yah ... ini memang soal pemanasan dan pengenalan, jadi sangat mudah. Soalnya sebagai berikut :

Buatlah program REVERSE.PAS menurut penjelasan berikut ini. Sebagai latihan, Anda belajar menentukan cara bagaimana membaca masukan string yang sangat panjang.

2.1 Masukan

Program itu harus membaca masukan dari file bernama REVERSE.IN. File ini akan berisikan satu baris teks dengan *panjang* ≤ 1000 karakter. (*Hint* : ini lebih panjang dari panjang maksimum string di Pascal).

2.2 Keluaran

Program harus menuliskan keluaran dalam file REVERSE.OUT. Keluaran adalah hanya satu baris teks yaitu string hasil pembalikan yang telah anda lakukan dari string masukan.

reverse.in

Olimpiade Nasional

reverse.out

lanoisaN edaipmilO

2.3 Contoh Penyelesaian

Penyelesaian : Berikut ini salah satu contoh penyelesaian masalah di atas. Masih banyak cara yang bisa dilakukan untuk menyelesaikannya.

```
program membalik_deretan_karakter;
uses crt;
var i,jumlah : integer;
    file_data : text;
    karakter : char;
    data_asal : array[1..1000] of char;
begin
    clrscr;
```

```

jumlah := 0;
assign(file_data, 'data.txt');
reset(file_data);
writeln('data reverse');
writeln('-----');
while not eof (file_data) do
  begin
    jumlah := jumlah + 1;
    read(file_data, karakter);
    write(karakter);
    data_asal[jumlah] := karakter;
  end;
close(file_data);
writeln;
writeln('-----');
writeln('hasil reverse');
writeln('-----');
for i:= jumlah downto 1 do write(data_asal[i]);
readln;
end.

```

2.4 Contoh hasilnya

data reverse

```

-----
program membalik_deretan_karakter; salam keadilan dan kesejahteraan
untuk rakyat di bumi pertiwi. supaya yakin maka tulisan ini
dibuat lebih dari 255 karakter.
apakah anda sudah mempersiapkan bekal untuk kehidupan
setelah kematian nanti ? d
wi sakethi pengrajin teknologi informasi 0816 403 432

```

hasil reverse

```

-----
234 304 6180 isamrofni igolonket nijargnep ihtekas iwd ? itnan
naitamek haletes napudihek kutnu lakeb nakpaisrepmem
hadus adna hakapa .retkarak 552 irad hibel t
aubid ini nasilut akam nikay ayapus .iwitrep imub id
taykar kutnu naarethajesek

```

3 Menghitung Massa

Suatu molekul terdiri atas sejumlah atom dan tersusun membentuk rumus kimia yang dituliskan dengan huruf-huruf yang menyatakan masing-masing atom ini. Misalnya H menyatakan atom hidrogen, C menyatakan atom karbon, O menyatakan atom oksigen. Jadi rumus kimia COOH menyatakan suatu molekul yang berisikan satu atom karbon, dua atom oksigen dan satu atom hidrogen.

Untuk menuliskan rumus ini secara efisien kita menggunakan aturan-aturan berikut ini.

Huruf-huruf yang menyatakan beberapa atom dapat dikelompokkan dengan pembatas tanda kurung yang disebut juga dengan istilah gugus atom. Misalnya rumus CH(OH) berisi gugus OH. Dalam suatu gugus bisa terdapat gugus-gugus lebih kecil. Untuk menyederhanakan suatu rumus kimia, kemunculan sejumlah huruf secara berturut-turut dapat digantikan dengan satu huruf saja tapi diikuti oleh suatu bilangan yang menyebutkan jumlah kemunculannya. Misalnya huruf COOHHH dapat ditulis sebagai CO₂H₃ dan ia mempresentasikan suatu molekul yang berisikan satu atom karbon, dua atom oksigen dan tiga atom hidrogen.

Selanjutnya, kemunculannya yang berturut-turut dari gugus yang sama dapat digantikan dengan gugus tersebut diikuti oleh bilangan yang menyatakan jumlah kemunculan gugus tersebut. Misalnya CH (CO₂H)(CO₂H)(CO₂H) dapat dituliskan sebagai CH(CO₂H)₃ dan molekul tersebut berisikan empat atom karbon, dua atom oksigen dan tiga atom hidrogen.

Dalam rumus kimia sebenarnya tentu bilangan yang menyatakan pengulangan kemunculan suatu gugus atom tersebut bisa berharga berapapun asal ≥ 1 . Dalam soal di sini bilangan tersebut dibatasi sampai dengan 9.

Massa dari suatu molekul adalah jumlah dari massa dari setiap atom yang tergantung di dalamnya. Satu atom hidrogen memiliki massa satu, satu atom karbon memiliki massa 12 dan satu atom oksigen memiliki massa 16.

Tuliskan suatu program dengan nama MASSA.PAS yang dapat menghitung massa molekul dari rumus molekul yang diberikan.

3.1 Format Masukan

File mmasuk adalah file teks dengan nama MASSA.IN. File berisi satu baris yang didalamnya tertuliskan rumus molekul yang hendak dihi-

tung massanya. Rumus molekul hanya akan berisikan kemungkinan karakter-karakter H, C, O, (,), 2, 3, ..., 9. Panjang string tidak akan lebih dari 100 karakter.

3.2 Format Keluaran

File keluaran adalah file teks dengan nama MASSA.OUT. Satu-satunya baris keluaran hanya berisikan massa dari molekul yang dinyatakan dengan rumus yang diberikan. Bilangan massa tidak akan lebih besar dari 1000 karakter.

3.3 Contoh penyelesaian

```
uses crt;
var hasil_kurung_total, hasil_kurung, hasil_akhir, kimia : string;
    p1, p : string;
    massa, tingkatx, k, i, j, jumlah_kurung : integer;
    akhir_sebelum, awal_sebelum, panjang, tingkat : integer;
    jumlah_asal, sekarang, kode_error : integer;
    awal, akhir, tingkat_kurung_awal, sudah_dihitung : array[1..25] of integer;
    pengali, kurung_punya, tingkat_kurung_akhir, kepakai : array[1..25] of integer;
    ada_kurung, selesai : boolean;
begin
    clrscr;
{ -----
    silahkan ganti-ganti bentuk kimia sesuai yang dikehendaki
    ----- }
    kimia := 'O(CO2H)3';
    kimia := 'CH(CO2H)3';
    kimia := '((CH)2(OH2H)(C(H))O)3';
    kimia := 'COOH';
    writeln('bentuk asal : ', kimia);
    hasil_akhir := '';
    panjang := length(kimia);
    tingkat := 0;
    jumlah_kurung := 0;
    for i:=1 to panjang do
```

```

begin
  p := copy(kimia,i,1);
  kepakai[i] := 0;
  sudah_dihitung[i] := 0;
  tingkat_kurung_akhir[i]:=0;
  if p='(' then
    begin
      inc(tingkat);
      inc(jumlah_kurung);
      awal[jumlah_kurung] := i;
      tingkat_kurung_awal[jumlah_kurung] := tingkat;
      akhir[jumlah_kurung] := 0;
      kurung_punya[i] := jumlah_kurung;
    end
  else
    if p=')' then
      begin
        kurung_punya[i] :=0;
        tingkat_kurung_akhir[i] := tingkat_kurung_awal[jumlah_kurung];
        akhir[jumlah_kurung] := i;
        dec(tingkat);
      end;
    end;
  { mencari posisi kurung tutup }
  tingkatx := 0;
  jumlah_kurung := 0;
  for i:=1 to panjang do
    begin
      p := copy(kimia,i,1);
      if p='(' then
        begin
          inc(tingkatx);
          inc(jumlah_kurung);
        end
      else
        if p=')' then
          begin
            for j:=i downto 1 do
              begin
                p1 := copy(kimia,j,1);
                if p1='(' then

```

```

begin
    if kepakai[j]=0 then
        begin
            akhir[kurung_punya[j]] :=i;
            kepakai[j] := 1;
            j := 1;
        end;
    end;
end;
dec(tingkatx);
end;
end;
for i:=1 to jumlah_kurung do
begin
    write('tingkat : ',tingkat_kurung_awal[i],
        ' mulai ',awal[i],' sampai ',akhir[i]);
    p:=copy(kimia,akhir[i]+1,1); { mencari faktor pengali untuk }
    val(p,k,kode_error);        { masing-masing kurung }
    if (k=0) then k:=1;
    pengali[i] :=k;
    writeln(' : ',pengali[i]);
end;
for i:=1 to jumlah_kurung do kepakai[i]:=0;
{ kepakai di sini digunakan untuk menandai bahwa
suatu karakter sudah dihitung }

selesai := false;
hasil_kurung_total := '';
repeat
    sekarang := 0;
    for i:=1 to jumlah_kurung do
        begin
            if (tingkat_kurung_awal[i]>sekarang) and (kepakai[i]=0)
            then sekarang:=i;
            { akan memproses tingkat kurung tertinggi dan
            pada posisi itu memang belum diproses }

        end;
    write('kurung ke-',kurung_punya[awal[sekarang]],' ');
    write('tingkat ',tingkat_kurung_awal[sekarang],' : ');
    kepakai[sekarang]:=1; { menandai bahwa posisi ini sudah

```

```

                                                    diproses }
hasil_kurung := '';
for i:=awal[sekarang] to akhir[sekarang] do
begin
  p := copy(kimia,i,1);
  if ((p='C') or (p='H') or (p='O')) and (sudah_dihitung[i]=0) then
  begin
    sudah_dihitung[i]:=1;
    p1 := copy(kimia,i+1,1);
    val(p1,k,kode_error);
    if (k>=2) then
      for j:=1 to k do hasil_kurung:=hasil_kurung + p
    else
      hasil_kurung := hasil_kurung + p;

  end;
end;

{ mencari apakah di dalam kurung yang sekarang
ada kurung lagi di dalamnya }
ada_kurung := false;
for i:=awal[sekarang]-1 downto 1 do
begin
  p:=copy(kimia,i,1);
  if (p='(') then
  begin
    awal_sebelum := awal[kurung_punya[i]];
    akhir_sebelum := akhir[kurung_punya[i]];
    if (awal_sebelum<awal[sekarang]) then
      if (akhir_sebelum>akhir[sekarang]) then
      begin
        write('awal : ',awal_sebelum,' - ',awal[sekarang]);
        write('ada di dalam kurung',' ');
        ada_kurung := true;
        i := 1;
      end;
    end;
  end;
end;

{ hasil yang didapat, dikalikan dengan pengali untuk
masing-masing kurung }
k := pengali[kurung_punya[awal[sekarang]]];

```

```

hasil_kurung_total := '';
if (awal[sekarang]<>1) and (akhir[sekarang]<>panjang-1) then
  for i:=1 to k do
    begin
      hasil_kurung_total:= hasil_kurung_total+ hasil_kurung;
    end
  else
    hasil_kurung_total:= hasil_kurung_total+ hasil_kurung;
writeln('pengali : ',k);
writeln('hasil kurung : ',hasil_kurung_total);
hasil_akhir := hasil_akhir + hasil_kurung_total;
writeln('hasil akhir: ',hasil_akhir);
sekarang := 0;

{ kalau semua posisi sudah diproses artinya kepakai[i]=1
  berarti proses selesai }
selesai := true;
for i:=1 to jumlah_kurung do
  if kepakai[i]=0 then
    begin
      selesai := false;
      i := jumlah_kurung;
    end;
until selesai;

panjang := length(kimia);
hasil_kurung_total := hasil_akhir;
if not ada_kurung then
  hasil_akhir := ''
else
  hasil_akhir := hasil_akhir;

{ untukantisipasi bentuk khusus dimana
  kurung terakhir di kolom terakhir-1, tapi
  kurung awalnya di kolom 1 }
k := pengali[tingkat_kurung_awal[1]];
if (awal[1]=1) and (akhir[1]=panjang-1) then
  for i:=1 to k do hasil_akhir := hasil_akhir + hasil_kurung_total;

{ untukantisipasi bentuk khusus dimana
  kurung terakhir di kolom terakhir-1, tapi

```

```

    kurung awalnya bukan di kolom 1 }

if (awal[1]<>1) and (akhir[1]=panjang-1) then
    for i:=1 to k do hasil_akhir := hasil_akhir + hasil_kurung_total;

{
    kalau-kalau ada yang belum dihitung,
    artinya nilai sudah_dihitung=0
    ini terjadi kalau tidak ada kurung sama sekali }
for i:=1 to awal[1] do
    if sudah_dihitung[i]=0 then
        begin
            p := copy(kimia,i,1);
            if (p='C') or (p='H') or (p='O') then
                hasil_akhir := hasil_akhir + p;
            end;

writeln('hasil akhirnya : ',hasil_akhir);
panjang := length(hasil_akhir);
massa := 0;
for i:=1 to panjang do
    begin
        p := copy(hasil_akhir,i,1);
        if p='H' then
            massa:=massa+1
        else
            if p='C' then
                massa:=massa+12
            else
                massa:=massa+16;
        end;
writeln('Massa : ',massa);
readln;
end.

```

3.4 Contoh hasil *running*

```

bentuk asal : COOH
kurung ke-0 tingkat -28666 : pengali : 1280
hasil kurung :
hasil akhir:

```

hasil akhirnya : COOH
Massa : 45

bentuk asal : CH(CO₂H)₃
tingkat : 1 mulai 3 sampai 8 : 3
kurung ke-1 tingkat 1 : pengali : 3
hasil kurung : COOH
hasil akhir: COOH
hasil akhirnya : COOHCOOHCOOHCH
Massa : 148

bentuk asal : ((CH)₂(OH₂H)(C(H))O)₃
tingkat : 1 mulai 1 sampai 20 : 3
tingkat : 2 mulai 2 sampai 5 : 2
tingkat : 2 mulai 7 sampai 12 : 1
tingkat : 2 mulai 13 sampai 18 : 1
tingkat : 3 mulai 15 sampai 17 : 1
kurung ke-5 tingkat 3 : awal : 13 - 15 ada di dalam kurung pengali : 1
hasil kurung : H
hasil akhir: H
kurung ke-2 tingkat 2 : awal : 1 - 2 ada di dalam kurung pengali : 2
hasil kurung : CHCH
hasil akhir: HCHCH
kurung ke-3 tingkat 2 : awal : 1 - 7 ada di dalam kurung pengali : 1
hasil kurung : OHHH
hasil akhir: HCHCHOHHH
kurung ke-4 tingkat 2 : awal : 1 - 13 ada di dalam kurung pengali : 1
hasil kurung : C
hasil akhir: HCHCHOHHHC
kurung ke-1 tingkat 1 : pengali : 3
hasil kurung : O
hasil akhir: HCHCHOHHHCO
hasil akhirnya : HCHCHOHHHCOHCHCHOHHHCOHCHCHOHHHCO
Massa : 222

Tapi mohon maaf sebesar-besarnya ... ternyata program tersebut belum dapat menyelesaikan :-) ... bentuk seperti ini misalnya :

bentuk asal : CO₂H
kurung ke-0 tingkat -28666 : pengali : 1280
hasil kurung :

hasil akhir:
hasil akhirnya : COH
Massa : 29

Mengapa demikian ? Ya ... ini menjadi PR lanjutan untuk Anda yang masih penasaran ... :-)

4 Ekspresi Aljabar

Buatlah program EKSPRESI.PAS sebagai latihan Anda menjelang ON. Latihan ini mulai agak sulit. Tujuan latihan ini untuk Anda membiasakan diri dengan kompiler Free Pascal yang digunakan di web server saat menguji pekerjaan Anda yang mungkin berbeda dengan kompiler yang sering Anda gunakan selama ini. Selain itu anda mulai berlatih pemrograman dengan tingkat kesulitan mulai mendekati soal-soal di ON nanti.

Program anda harus dapat membaca string masukan yang berisi ekspresi aritmetika yang terdiri atas operator pangkat-kali-bagi-tambah-kurang dan menuliskan urutan pengerjaannya yang benar. Misalnya :

$$a - b + c/d * e/f \wedge g - h * j$$

Untuk menentukan urutan pengerjaannya dalam penulisannya operator-operator tersebut diberikan tingkat prioritas; pangkat paling tinggi, kemudian kali dan bagi pada prioritas yang sama, dan terakhir tambah dan kurang, pada prioritas yang sama. (Note : Dalam latihan ini tanda kurung atau operator lain belum diikutsertakan). Dengan adanya tingkat prioritas ini maka $f \wedge g$ harus dikerjakan sebelum e/f atau $g - h$. Jika prioritas sama sehingga mana yang di sebelah kiri akan dikerjakan lebih dahulu dari yang di sebelah kanan. Untuk contoh di atas c/d dikerjakan terlebih dahulu dari pada $d * e$. Dengan menggunakan nama variabel sementara xi untuk menerima hasil pengerjaan suatu operasi, maka salah satu urutan pengerjaan ekspresi tersebut adalah :

$$\begin{aligned}x1 &= a - b \\x2 &= c/d \\x3 &= x2 * e \\x4 &= f \wedge g \\x5 &= x3/x4 \\x6 &= x1 + x5 \\x7 &= h * j \\x8 &= x6 - x7\end{aligned}$$

4.1 Masukan

Program itu harus membaca masukan dari file bernama EKSPRESI.IN. File ini akan berisikan satu baris teks ekspresi aritmetika dengan panjang < 256 karakter. Operator pangkat ditulis dengan simbol ' \wedge ', operator kali dengan simbol '*', operator bagi dengan simbol '/', operator tambah dengan simbol '+', dan operator kurang dengan simbol '-'. Operand-operannya sendiri adalah menggunakan karakter huruf tunggal (a-Z, A-Z) untuk memudahkan anda membaca masukan. Dalam ekspresi tidak ada karakter spasi atau karakter lainnya selain huruf atau karakter simbol operator tersebut di atas.

4.2 Keluaran

Program harus menuliskan keluaran dalam file bernama EKSPRESI.OUT. Keluaran berisikan baris-baris operasi untuk mengerjakan ekspresi masukan yang dibantu oleh variabel-variabel sementara x_i . Agar keluaran menjadi unik maka urutan sedapat mungkin dari kiri ke kanan ekspresi kecuali kalau terkait dengan prioritas. Misalnya $a - b$ harus ditulis lebih dahulu dari c/d karena $a - b$ tidak bergantung hasil c/d . Variabel-variabel sementara x_i dituliskan sebagai karakter x dan bilangan i dengan i membesar dari baris pertama ke baris terakhir.

Contoh 1

EKSPRESI.IN

$$a - b + c/d$$

File.OUT

$$x1 = a - b$$

$$x2 = c/d$$

$$x3 = x1 + x2$$

Contoh 2

EKSPRESI.IN

$$c/d * e/f \wedge g$$

EKSPRESI.OUT

$x1 = c/d$
 $x2 = x1 * e$
 $x3 = f \wedge g$
 $x4 = x2/x3$

Contoh 3

EKSPRESI.IN

$a - b + c/d * e/f \wedge g - h * j$

EKSPRESI.OUT

$x1 = a - b$
 $x2 = c/d$
 $x3 = x2 * e$
 $x4 = f \wedge g$
 $x5 = x3/x4$
 $x6 = x1 + x2$
 $x7 = h * j$
 $x8 = x6 - x7$

4.3 Contoh Penyelesaian

Berikut ini adalah contoh program untuk menyelesaikan masalah di atas.

```
program analisa_ekpresi_aljabar;  
{ versi sabtu }  
uses crt;  
var ekspresi : string;  
    batas_kiri,batas_kanan,proses,sekarang,i,j,jumlah_tanda,panjang : byte;  
    tanda,karakter : array[1..255] of string[1];  
    str_temp,suku_kiri,suku_kanan: string[2];  
    prioritas : array[1..255] of byte;  
    cari_prioritas, selesai : boolean;  
    substitusi : string[6];  
    jumlah_tanda_asli,nilai: byte;  
    kode_error : integer;
```

```

begin
  clrscr;
  ekspresi := 'c/d*e/f^g';
  ekspresi := 'a-b+c/d*e/f^g-h*j';
  panjang := length(ekspresi);
  jumlah_tanda := 0;
  proses := 0;

{
  proses mencari jumlah operator dan operator apa saja
  yang ada beserta tingkatnya }

  for i:=1 to panjang do
    begin
      karakter[i] := copy(ekspresi,i,1);
      if (karakter[i]='-') or (karakter[i]='+') then
        begin
          inc(jumlah_tanda);
          prioritas[jumlah_tanda] := 1;
          tanda[jumlah_tanda] := karakter[i];
        end
      else
        if (karakter[i]='/') or (karakter[i]='*') then
          begin
            inc(jumlah_tanda);
            prioritas[jumlah_tanda] := 2;
            tanda[jumlah_tanda] := karakter[i];
          end
        else
          if (karakter[i]='^') then
            begin
              inc(jumlah_tanda);
              prioritas[jumlah_tanda] := 3;
              tanda[jumlah_tanda] := karakter[i];
            end;
          end;
      jumlah_tanda_asli := jumlah_tanda;

      { mencari operator mana yang akan dikerjakan terlebih dahulu }
      cari_prioritas := false;

```

```

repeat
    sekarang := 1;
    for i:=1 to jumlah_tanda do
        begin
            if prioritas[sekarang+1]>prioritas[sekarang] then
                sekarang := sekarang+1
            else
                cari_prioritas:=true;
            end;
        until cari_prioritas;
    selesai := false;
repeat
    inc(proses);
    writeln('ekspresi : ',ekspresi);
    write('tanda ke : ',sekarang,' yang mau dikerjakan ');
    textcolor(yellow+blink);writeln(tanda[sekarang]);
    textcolor(white);
    batas_kiri := 2*sekarang-1;
    batas_kanan := 2*sekarang+1;
    writeln('batas kiri : ',batas_kiri,' batas kanan : ',batas_kanan);
    writeln('karakter batas kiri : ',karakter[batas_kiri],' ',
        'karakter batas kanan : ',karakter[batas_kanan]);
    suku_kiri := karakter[batas_kiri];
    { kalau suku kiri=1 ini artinya x1,
      kalau suku kiri=2 ini artinya x2, dan seterusnya }
    val(suku_kiri,nilai,kode_error);
    if (nilai>0) then suku_kiri:='x'+suku_kiri;

    suku_kanan := karakter[batas_kanan];
    val(suku_kanan,nilai,kode_error);
    if (nilai>0) then suku_kanan:='x'+suku_kanan;

    substitusi := suku_kiri+tanda[sekarang]+suku_kanan;

    str(proses,str_temp);
    writeln('----- substitusi x',str_temp,'=',
        substitusi,' -----');
    ekspresi := '';
    for i:=1 to batas_kiri-1 do ekspresi:=ekspresi+karakter[i];
    ekspresi := ekspresi + str_temp;
    for i:=batas_kanan+1 to panjang do ekspresi:=ekspresi+karakter[i];

```

```

        writeln('ekspresi baru setelah direduksi : ',ekspresi);

{ proses seperti di awal kembali }
{
    proses mencari jumlah operator dan operator apa saja
    yang ada beserta tingkatnya }
    panjang := length(ekspresi);
    jumlah_tanda := 0;

    for i:=1 to panjang do
        begin
            karakter[i] := copy(ekspresi,i,1);
            if (karakter[i]='-') or (karakter[i]='+') then
                begin
                    inc(jumlah_tanda);
                    prioritas[jumlah_tanda] := 1;
                    tanda[jumlah_tanda] := karakter[i];
                end
            else
                if (karakter[i]='/') or (karakter[i]='*') then
                    begin
                        inc(jumlah_tanda);
                        prioritas[jumlah_tanda] := 2;
                        tanda[jumlah_tanda] := karakter[i];
                    end
                else
                    if (karakter[i]='^') then
                        begin
                            inc(jumlah_tanda);
                            prioritas[jumlah_tanda] := 3;
                            tanda[jumlah_tanda] := karakter[i];
                        end;
                    end;
        end;

    { mencari operator mana yang akan dikerjakan terlebih dahulu }
    cari_prioritas := false;
    writeln('jumlah tanda ',jumlah_tanda);
    repeat
        sekarang := 1;
        for i:=1 to jumlah_tanda do
            begin

```

```

        if prioritas[sekarang+1]>prioritas[sekarang] then
            sekarang := sekarang+1
        else
            cari_prioritas:=true;
        end;
        if jumlah_tanda=1 then cari_prioritas:=true;
until cari_prioritas;
selesai := false;
if jumlah_tanda=1 then
    begin
        str(jumlah_tanda_asli,str_temp);
        writeln('----- x',str_temp,'=x',karakter[1],karakter[2],'x',ka
            if jumlah_tanda=1 then selesai:=true;
        end;
    readln;
until selesai;
end.

```

4.4 Contoh hasil *running*

```

ekspresi : a-b+c/d*e/f^g-h*j
tanda ke : 1 yang mau dikerjakan -
batas kiri : 1 batas kanan : 3
karakter batas kiri : a karakter batas kanan : b
----- substitusi x1=a-b -----
ekspresi baru setelah direduksi : 1+c/d*e/f^g-h*j
jumlah tanda 7

```

```

ekspresi : 1+c/d*e/f^g-h*j
tanda ke : 2 yang mau dikerjakan /
batas kiri : 3 batas kanan : 5
karakter batas kiri : c karakter batas kanan : d
----- substitusi x2=c/d -----
ekspresi baru setelah direduksi : 1+2*e/f^g-h*j
jumlah tanda 6

```

```

ekspresi : 1+2*e/f^g-h*j
tanda ke : 2 yang mau dikerjakan *
batas kiri : 3 batas kanan : 5
karakter batas kiri : 2 karakter batas kanan : e
----- substitusi x3=x2*e -----

```

ekspresi baru setelah direduksi : $1+3/f^g-h*j$
jumlah tanda 5

ekspresi : $1+3/f^g-h*j$
tanda ke : 3 yang mau dikerjakan ^
batas kiri : 5 batas kanan : 7
karakter batas kiri : f karakter batas kanan : g
----- substitusi $x4=f^g$ -----
ekspresi baru setelah direduksi : $1+3/4-h*j$
jumlah tanda 4

ekspresi : $1+3/4-h*j$
tanda ke : 2 yang mau dikerjakan /
batas kiri : 3 batas kanan : 5
karakter batas kiri : 3 karakter batas kanan : 4
----- substitusi $x5=x3/x4$ -----
ekspresi baru setelah direduksi : $1+5-h*j$
jumlah tanda 3

ekspresi : $1+5-h*j$
tanda ke : 1 yang mau dikerjakan +
batas kiri : 1 batas kanan : 3
karakter batas kiri : 1 karakter batas kanan : 5
----- substitusi $x6=x1+x5$ -----
ekspresi baru setelah direduksi : $6-h*j$

jumlah tanda 2
ekspresi : $6-h*j$
tanda ke : 2 yang mau dikerjakan *
batas kiri : 3 batas kanan : 5
karakter batas kiri : h karakter batas kanan : j
----- substitusi $x7=h*j$ -----
ekspresi baru setelah direduksi : $6-7$
jumlah tanda 1
----- $x8=x6-x7$

5 MULTIPALINDROM

Palindrom adalah kata yang dapat dibaca sama saja baik dari kiri ke kanan ataupun dari kanan ke kiri. Suatu palindrom sedikitnya berisi satu

huruf. Misalnya, "malam", "a" dan "ada" masing-masing adalah palindrom. Sebaliknya, setiap kata bukan merupakan palindrom dapat dianggap sebagai deretan sejumlah palindrom. Dengan kata lain, kata tersebut dapat dipecah-pecahkan ke dalam sejumlah palindrom. Jadi, setiap kata pada dasarnya dapat dipandang sebagai multipalindrom yang tersusun atas n palindrom, dengan $n > 0$. Untuk setiap kata terdapat sejumlah kemungkinan harga n . Dengan definisi itu maka setiap palindrom adalah multipalindrom dengan jumlah minimal $n=1$. Misalnya, kata "minimisasi" terdiri atas sedikitnya 2 palindrom yaitu "minim"- "isasi" (Red : ralat dan versi sebelumnya).

Buatlah suatu program dengan nama MULTIPAL.AS yang akan menghitung jumlah palindrom minimal dari suatu kata yang diberikan.

5.1 Format Masukan

File masukan adalah MULTIPAL.IN yang hanya berisi kata untuk dipecah-pecah ke dalam sejumlah palindrom. Karakter-karakter untuk membentuk kata adalah huruf kecil (a-z). Panjang dari kata tidak akan lebih dari 100 huruf.

5.2 Format Keluaran

Keluaran dituliskan dalam file MULTIPAL.OUT yang menyebutkan jumlah terkecil palindrom yang dapat dibuat.

Contoh-contoh :

| | | |
|--------------|--------------|---------------|
| MULTIPAL.IN | MULTIPAL.IN | MULTIPAL.IN |
| anaban | abaccbcb | anavolimilana |
| MULTIPAL.OUT | MULTIPAL.OUT | MULTIPAL.OUT |
| 2 | 3 | 5 |

PENJELASAN CONTOH :

#1 a naban

#2 aba cc bcb

#3 ana v o limil ana

5.3 Contoh Penyelesaian

```
program mencari_multipaliandrom_pada_suatu_tulisan;
{ dikembangkan oleh dwi sakethi
      dwijim@unila.ac.id
      dwijim@gmail.com
      0816 403 432
      pada tanggal 18 agustus 2006 }

uses crt;

var file_data,file_hasil : text;
    tulisan_selesai,tulisan_asli,potongan_tulisan,tulisan : string;
    selesai : boolean;
    jumlah_paliandrom,panjang_tulisan_tetap,paliandrom : byte;
    mulai_tulisan_baru,jumlah_looping,batas_kanan : byte;

{
    fungsi ini memberikan nilai 1 jika kata yang dicek berupa
    paliandrom, jika kata yg dicek bukan merupakan paliandrom
    maka fungsi ini memberikan nilai 0
}
function cek_paliandrom(tulisan_yg_dicek : string):byte;
var hasil          : byte;
    ii,panjang_tulisan_x : byte;
    hasil_reverse   : string;
begin
    hasil          := 0;
    hasil_reverse  := '';
    panjang_tulisan_x := length(tulisan_yg_dicek);
    for ii:=panjang_tulisan_x downto 1 do
        begin
            hasil_reverse := hasil_reverse +
                copy (tulisan_yg_dicek,ii,1);
        end;
    if tulisan_yg_dicek=hasil_reverse then
        hasil:=1;
    cek_paliandrom := hasil;
end;
```

```

{ --- program utama --- }
begin

    clrscr;
    { membuka file data }
    assign(file_data,'multipal.in');
    reset(file_data);

    { membaca data tulisan }
    read(file_data,tulisan);
    writeln('tulisan asal : ',tulisan);

    { membuat file hasil }
    assign(file_hasil,'multipal.out');
    rewrite(file_hasil);

    { proses pencarian paliandrom dilakukan sampai batas
      akhir tulisan }
    selesai := false;
    batas_kanan := length(tulisan);
    panjang_tulisan_tetap := length(tulisan);
    potongan_tulisan := tulisan;
    tulisan_asli := tulisan;
    tulisan_selesai := '';
    jumlah_paliandrom := 0;
    repeat
        paliandrom :=cek_paliandrom(potongan_tulisan);
        writeln('asal : ',potongan_tulisan, ' cek : ',paliandrom);
        if paliandrom=0 then
            begin
                batas_kanan := batas_kanan - 1;
                potongan_tulisan := copy(potongan_tulisan,1,batas_kanan);
            end
        else
            begin
                writeln('paliandrom : ',potongan_tulisan);
                tulisan_selesai := tulisan_selesai + potongan_tulisan;
                mulai_tulisan_baru := length(potongan_tulisan)+1;
                potongan_tulisan := copy(tulisan,mulai_tulisan_baru,panjang_tulisan_tetap);
                panjang_tulisan_tetap := length(potongan_tulisan);
                batas_kanan := length(potongan_tulisan);
            end
        end repeat

```

```

        tulisan := potongan_tulisan;
        if tulisan<>'' then
            jumlah_paliandrom := jumlah_paliandrom + 1;
            readln;
        end;
        writeln('tulisan kontrol : ',tulisan_selesai);
        if tulisan_selesai=tulisan_asli then
            selesai := true;
until selesai;

writeln('jumlah paliandrom : ',jumlah_paliandrom);
readln;

{ tulisan hasil ke file output }
writeln(file_hasil,jumlah_paliandrom);
{ menutup kembali file yg telah diakses }
close(file_hasil);
close(file_data);
end.

```

5.4 Hasil Program

```

cek : 0
tulisan kontrol :
asal : anavolimilana cek : 0
tulisan kontrol :
asal : anavolimilan cek : 0
tulisan kontrol :
asal : anavolimila cek : 0
tulisan kontrol :
asal : anavolimil cek : 0
tulisan kontrol :
asal : anavolimi cek : 0
tulisan kontrol :
asal : anavolim cek : 0
tulisan kontrol :
asal : anavoli cek : 0
tulisan kontrol :
asal : anavol cek : 0
tulisan kontrol :
asal : anavo cek : 0

```

tulisan kontrol :
asal : anav cek : 0
tulisan kontrol :
asal : ana cek : 1
paliandrom : ana

tulisan kontrol : ana
asal : volimilana
cek : 0
tulisan kontrol : ana
asal : volimilana cek : 0
tulisan kontrol : ana
asal : volimilan cek : 0
tulisan kontrol : ana
asal : volimila cek : 0
tulisan kontrol : ana
asal : volimil cek : 0
tulisan kontrol : ana
asal : volimi cek : 0
tulisan kontrol : ana
asal : volim cek : 0
tulisan kontrol : ana
asal : voli cek : 0
tulisan kontrol : ana
asal : vol cek : 0
tulisan kontrol : ana
asal : vo cek : 0
tulisan kontrol : ana
asal : v cek : 1
paliandrom : v

tulisan kontrol : anav
asal : olimilana

cek : 0
tulisan kontrol : anav
asal : olimilana cek : 0
tulisan kontrol : anav
asal : olimilan cek : 0
tulisan kontrol : anav
asal : olimila cek : 0

tulisan kontrol : anav
asal : olimil cek : 0
tulisan kontrol : anav
asal : olimi cek : 0
tulisan kontrol : anav
asal : olim cek : 0
tulisan kontrol : anav
asal : oli cek : 0
tulisan kontrol : anav
asal : ol cek : 0
tulisan kontrol : anav
asal : o cek : 1
paliandrom : o

tulisan kontrol : anavo
asal : limilana

cek : 0
tulisan kontrol : anavo
asal : limilana cek : 0
tulisan kontrol : anavo
asal : limilan cek : 0
tulisan kontrol : anavo
asal : limila cek : 0
tulisan kontrol : anavo
asal : limil cek : 1
paliandrom : limil

tulisan kontrol : anavolimil
asal : ana

cek : 0
tulisan kontrol : anavolimil
asal : ana cek : 1
paliandrom : ana

tulisan kontrol : anavolimilana
asal :

cek : 1
paliandrom :

tulisan kontrol : anavolimilana

jumlah paliandrom : 5

6 Menghitung Jumlah Huruf

Masalah mencari jumlah huruf pada suatu kata atau kalimat. Soal yang lebih jelas, mudah-mudahan kapan-kapan akan ditulis di sini.

6.1 Contoh Penyelesaian

Contoh penyelesaiannya seperti berikut :

```
{ -----
program untuk menghitung jumlah huruf dan jenisnya
dibuat dengan bahasa pascal
diproses dengan sistem operasiGNU Linux Ubuntu
compiler free pascal
dwi sakethi http://dwijim.staff.unila.ac.id

nama file : hitung-jumlah-huruf.pas

----- }

uses crt;
{ karena ada perintah cetak ke layar }

var kalimat : string;
    huruf_ke, huruf_ke_isi : byte;
    jumlah_huruf_ke : array[1..100] of byte;
    isi_huruf_ke : array [1..100] of string;

{ prosedur mencetak identitas pembuat program }
procedure identitas_pembuat;
begin
    textcolor(yellow+blink);
    gotoxy(1,24);write('dwi sakethi http://dwijim.staff.unila.ac.id');
    textcolor(white);
end;
```

```

{ menentukan ke mana ular akan bergerak }
procedure masukan_kalimat;
begin
  gotoxy(1,1);write('masukan kalimatnya :');
  readln(kalimat);
end;

{ menentukan ke mana ular akan bergerak }
procedure hitung_huruf;
var panjang_kalimat : byte;
    huruf_sekarang : string;
    jumlah_huruf_yang_ada : byte;
    ada_huruf : boolean;
begin
  jumlah_huruf_yang_ada := 1;
  panjang_kalimat := length(kalimat);
  writeln('panjang kalimat : ',panjang_kalimat);
  for huruf_ke:=1 to panjang_kalimat do
    begin
      huruf_sekarang :=copy(kalimat,huruf_ke,1);
      { dari kata atau kalimat diambil per huruf selain spasi}

      if huruf_sekarang <> ' ' then
        begin
          ada_huruf := false;
          for huruf_ke_isi:=1 to jumlah_huruf_yang_ada do
            begin
              if huruf_ke = 1 then
                begin
                  { buat array yang berisi huruf-huruf yang ditemukan
                    sampai dengan proses ini,
                    untuk huruf pertama, pasti jadi elemen pertama }
                  isi_huruf_ke[1] := huruf_sekarang;
                  jumlah_huruf_ke[1] := 1;
                  ada_huruf := true;
                end
              else
                begin
                  if isi_huruf_ke[huruf_ke_isi] = huruf_sekarang then
                    begin

```

```

        inc(jumlah_huruf_ke[huruf_ke_isi]);
        ada_huruf := true;
    end;
end
end; { akhir looping for huruf_ke_isi }

{ jika huruf yang sedang diproses tidak ada di antara salah
  satu dari array huruf yang sudah ada maka
  ini berarti huruf baru dan jumlahnya pasti 1 }
if (ada_huruf = false) then
begin
    inc(jumlah_huruf_yang_ada);
    isi_huruf_ke[jumlah_huruf_yang_ada] := huruf_sekarang;
    jumlah_huruf_ke[jumlah_huruf_yang_ada] := 1;
end; { akhir if ada_huruf = false }
end; { akhir if huruf_sekarang }
end; { akhir looping kalimat }

for huruf_ke:=1 to jumlah_huruf_yang_ada do
begin
    writeln(isi_huruf_ke[huruf_ke], ' : ', jumlah_huruf_ke[huruf_ke]);
end;
end;

{ ----- program utama ----- }
begin
    clrscr;
    identitas_pembuat;
    masukan_kalimat;
    hitung_huruf;
end.

```

6.2 Contoh Keluaran

Contoh hasilnya seperti berikut :

```

dwijim@dwijim-desktop:~/Documents/olimpiade$ ppc386 hitung-jumlah-huruf.pas
Free Pascal Compiler version 2.2.2-8 [2009/01/08] for i386
Copyright (c) 1993-2008 by Florian Klaempfl
Target OS: Linux for i386
Compiling hitung-jumlah-huruf.pas

```

```
Linking hitung-jumlah-huruf
99 lines compiled, 0.1 sec
dwijim@dwijim-desktop:~/Documents/olimpiade$
masukan kalimatnya :buku tamu
panjang kalimat : 9
b : 1
u : 3
k : 1
t : 1
a : 1
m : 1
```

7 Variasi Masalah Bilangan Prima

Soal ini hanya merupakan variasi dari masalah mencari bilangan prima. Jadi jika Anda sudah bisa mencari bilangan prima maka ini bisa menjadi latihan berikutnya. Buatlah sebuah program yang bertugas untuk mencari bilangan prima ke-k, simulasinya sebagai berikut:

Pengguna program terlebih dahulu akan ditanyakan banyaknya bilangan prima yang akan dicari. Lalu pengguna akan ditanyakan kembali bilangan prima beberapa yang akan dicari, terus berulang sesuai dengan banyaknya bilangan prima yang akan dicari. Contoh: Pengguna akan ditanyakan

```
Berapa banyak bilangan prima yang dicari : dan diinputkan 3
Pengguna akan ditanya kembali
Bilangan prima ke berapa yang akan dicari : dan diinputkan 4
Bilangan prima ke berapa yang akan dicari : dan diinputkan 1
Bilangan prima ke berapa yang akan dicari : dan diinputkan 8
```

Maka hasil output nya akan menghasilkan:

```
Bilangan prima ke 4 adalah 7
Bilangan prima ke 1 adalah 2
Bilangan prima ke 8 adalah 19
```

Catatan: contoh output dan inputan hanya sebagai contoh, silahkan mengubah-ubah kalimat dengan kreasi anda sendiri, yang terpenting tujuan dan maksud dari program harus sesuai dengan simulasi yang telah dijelaskan.

7.1 Contoh Penyelesain

Contoh penyelesaiannya seperti berikut :

```

{ -----
program untuk mencetak bilangan prima sebanyak n buah
dan merupakan bilangan prima ke-i dari deretan
bilangan prima
dibuat dengan bahasa pascal
diproses dengan sistem operasiGNU Linux Ubuntu
compiler free pascal
dwi sakethi http://dwijim.staff.unila.ac.id

nama file : prima-k.pas
----- }

```

```

uses crt;
{ karena ada perintah cetak ke layar }

var banyaknya_bilangan : byte;
    bilangan_prima_ke   : array[1..100] of byte;
    bilangan_prima      : longint;
    ketemu_prima        : longint;

{ prosedur mencetak identitas pembuat program }
procedure identitas_pembuat;
begin
    textcolor(yellow+blink);
    gotoxy(1,24);write('dwi sakethi http://dwijim.staff.unila.ac.id');
    textcolor(white);
end;

{ memasukkan data banyaknya bilangan prima yang dicari
  dan urutan masing-masing bilangan prima }
procedure masukan_data;
var bilangan_ke : byte;
begin
    gotoxy(1,1);write('Berapa banyak bilangan prima :');
    readln(banyaknya_bilangan);
    for bilangan_ke:=1 to banyaknya_bilangan do
    begin
        write('Bilangan prima ke : ');
        readln(bilangan_prima_ke[bilangan_ke]);
    end;
end;

```

```

{ mengecek apakah suatu bilangan termasuk bilangan prima atau bukan}
function cek_prima_apa_bukan(bilangan_ini:longint):boolean;
var bilangan_sekarang : longint;
begin
cek_prima_apa_bukan:=TRUE;
for bilangan_sekarang:=2 to bilangan_ini-1 do
begin
if (bilangan_ini mod bilangan_sekarang)=0 then
begin
cek_prima_apa_bukan:=FALSE;
exit;
end;
end;
end;

{ ----- program utama ----- }
{ mencetak bilangan prima }
procedure mencetak_hasil;
var bilangan_ke : byte;
urutan : byte;
begin
for bilangan_ke:=1 to banyaknya_bilangan do
begin
write('Bilangan prima ke : ',bilangan_prima_ke[bilangan_ke],' adalah : ');
urutan := 0;
bilangan_prima := 2;
repeat
if cek_prima_apa_bukan(bilangan_prima)=TRUE then
begin
inc(urutan);
ketemu_prima := bilangan_prima;
{ jika suatu bilangan termasuk bilangan prima maka
urutan bertambah yang tadinya 0 menjadi 1 dst
kemudian ditandai juga bahwa bilangan itu adalah
bilangan prima }
end;
inc(bilangan_prima);
until urutan=bilangan_prima_ke[bilangan_ke];
writeln(ketemu_prima);
end;
end;

```

```

end;

begin
  clrscr;
  identitas_pembuat;
  masukan_data;
  mencetak_hasil;
  writeln;
end.

```

7.2 Contoh Keluaran

Contoh hasilnya seperti berikut :

```

dwijim@dwijim-desktop:~/Documents/olimpiade$ ppc386 prima-k.pas
Free Pascal Compiler version 2.2.2-8 [2009/01/08] for i386
Copyright (c) 1993-2008 by Florian Klaempfl
Target OS: Linux for i386
Compiling prima-k.pas
Linking prima-k
92 lines compiled, 0.1 sec
dwijim@dwijim-desktop:~/Documents/olimpiade$
Berapa banyak bilangan prima :3
Bilangan prima ke : 4
Bilangan prima ke : 1
Bilangan prima ke : 8
Bilangan prima ke : 4 adalah : 7
Bilangan prima ke : 1 adalah : 2
Bilangan prima ke : 8 adalah : 19

```

8 Membuat Simulasi Ular Berjalan

Buatlah sebuah program yang merupakan animasi ular yang berjalan secara acak, ular terdiri dari 7 karakter dengan : 1 karakter kepala 5 karakter badan 1 karakter ekor Ketentuan gerakan yang tidak boleh adalah setelah ke kiri kemudian ke kanan atau sebaliknya, setelah ke atas kemudian ke bawah atau sebaliknya. Selain gerakan tersebut diperbolehkan. Ini untuk menghindari gerakan di tempat. Badan dan ekor akan mengikuti jejak yang dilewati kepala. Ekor akan selalu melewati posisi yang pernah dilewati kepala.

8.1 Contoh Penyelesaian

Berikut ini salah satu contoh penyelesaian masalah di atas. Tidak menutup kemungkinan ada penyelesaian lain yang lebih baik.

```
{ program simulasi ular yang berjalan secara acak }

program snake;

uses crt;

const panjang_ular = 19;
      maks_arah     = 5;
      { 1 : ke kiri
        2 : ke atas
        3 : ke kanan
        4 : ke bawah }
      kolom_batas_kanan = 79;
      baris_batas_bawah = 24;

var ular : array[1..panjang_ular] of string;
    looping_badan_ular : byte;
    arah_sekarang, arah_sebelumnya : string;
    baris, kolom : integer;
    kolom_ular, baris_ular : array[1..panjang_ular] of integer;

{ untuk menampillkan nama pembuat program }
procedure pembuat;
begin
  gotoxy(1,24);write('dibuat oleh : dwi sakethi http://dwijim.staff.unila.ac.id')
end;

{ untuk menentukan gerak ular ke mana }
function gerak_ular(arah_sebelumnya:string):string;
var bilangan : byte;
    selesai : boolean;
begin
{ looping sampai dengan arah yang dibolehkan }

randomize;
selesai := false;
```

```

repeat
  bilangan := random(maks_arah);
  { 1 : ke kiri
    2 : ke atas
    3 : ke kanan
    4 : ke bawah }
  if (bilangan=1) and (arah_sebelumnya<>'kanan') then
    begin
      gerak_ular := 'kiri';
      selesai := true;
    end
  else
    if (bilangan=2) and (arah_sebelumnya<>'bawah') then
      begin
        gerak_ular := 'atas';
        selesai := true;
      end
    else
      if (bilangan=3) and (arah_sebelumnya<>'kiri') then
        begin
          gerak_ular := 'kanan';
          selesai := true;
        end
      else
        if (bilangan=4) and (arah_sebelumnya<>'atas') then
          begin
            gerak_ular := 'bawah';
            selesai := true;
          end;
        until selesai;
      end;
    until selesai;

  end;

{ program utama }
begin
  ular[1] := '@'; { kepala ular }
  for looping_badan_ular:=2 to panjang_ular-2 do
    ular[looping_badan_ular] := '#'; { badan ular }
  ular[looping_badan_ular-1] := '*'; { ini ekornya }
  ular[looping_badan_ular] := ' '; { ini ekornya }

```

```

arah_sekarang := '';
{ arah awal gerakan ular masih kosong }

baris := 12;
kolom := 40;
{ posisi awal di tengah layar }

for looping_badan_ular := 1 to panjang_ular do
begin
    baris_ular[looping_badan_ular]:=baris+looping_badan_ular;
    kolom_ular[looping_badan_ular]:=kolom+looping_badan_ular;
end;
{ posisi awal ular }

clrscr;

repeat
{ clrscr; }
{ menghapus layar }
pembuat;

    for looping_badan_ular := 1 to panjang_ular do
begin
    gotoxy(kolom_ular[looping_badan_ular],
           baris_ular[looping_badan_ular]);
    write(ular[looping_badan_ular]);
end;

for looping_badan_ular := panjang_ular downto 2 do
begin
    kolom_ular[looping_badan_ular] := kolom_ular[looping_badan_ular-1];
    baris_ular[looping_badan_ular] := baris_ular[looping_badan_ular-1];
end;

arah_sebelumnya := arah_sekarang;
arah_sekarang := gerak_ular(arah_sebelumnya);
if (arah_sekarang='kiri') then
begin
    dec(kolom_ular[1]);
    if (kolom_ular[1]=1) then kolom_ular[1]:=kolom_batas_kanan;

```

```

        end
    else
    if (arah_sekarang='atas') then
        begin
            dec(baris_ular[1]);
            if (baris_ular[1]=1) then baris_ular[1]:=baris_batas_bawah;
            end
        else
    if (arah_sekarang='bawah') then
        begin
            inc(baris_ular[1]);
            if (baris_ular[1]=baris_batas_bawah) then baris_ular[1]:=1;
            end
        else
    if (arah_sekarang='kanan') then
        begin
            inc(kolom_ular[1]);
            if (kolom_ular[1]=kolom_batas_kanan) then kolom_ular[1]:=1;
            end;
        end;

        delay(150);
    until keypressed;

    { looping sampai ada penekanan tombol sembarang }
    {
        for looping_badan_ular := 1 to panjang_ular do
            begin
                writeln(kolom_ular[looping_badan_ular], ' - ',
                    baris_ular[looping_badan_ular]);
            end;
        }
    end.
    { akhir program utama }

```

9 Mencari Selisih Terbesar

Diberikan data ketinggian yang di catat dalam perjalanan dari suatu posisi awal ke posisi akhir. Data ketinggian adalah bilangan-bilangan integer (bulat) positif. Jalan kadang menaik, kadang menurun, kadang datar saja. Posisi dimana terjadi perubahan menaik kemudian menurun (boleh diselingi

jalan datar) didefinisikan sebagai puncak dari suatu bukit. Sebaliknya, posisi terjadi perubahan dari menurun terus menaik (boleh diselingi bagian jalan yang datar) didefinisikan sebagai titik terbawah suatu lembah. Walaupun perubahan tersebut kecil saja, definisi itu tetap berlaku. Carilah beda ketinggian terbesar antara puncak bukit dengan titik terbawah lembah berikutnya atau sebaliknya antara titik terbawah lembah dengan puncak bukit berikutnya pada data perjalanan tersebut.

9.1 Input

Masukan berisi data yang bisa sangat banyak sekali. Setiap elemen data dalam baris tersendiri. Anda membacanya dari yang pertama hingga end of file; minimal ada dua data dalam masukan.

9.2 Output

Program hanya menghasilkan satu bilangan yang menyatakan beda ketinggian terbesar yang diperoleh. Perbedaan tinggi paling besar dijamin tidak akan melebihi harga long integer dalam Pascal.

Contoh:

Inputan :

10
26
26
35
35
27
30
30
45
10
8
9

Maka output nya adalah : 37

9.3 Contoh Penyelesaian

Berikut ini salah satu contoh penyelesaian masalah di atas. Tidak menutup kemungkinan ada penyelesaian lain yang lebih baik.

```

{ -----
  program untuk mencari beda ketinggian terbesar dari data bukit
  yang ada. data bukit disimpan dalam file bukit.dat
  nama file tinggi-bukit.pas
  ----- }

program mencari_beda_tertinggi;
uses crt;
var tertinggi, terendah, bukit : longint;
    { tertinggi adalah bukit yang paling tinggi
      terendah adalah bukit yang paling rendah
      bukit adalah data bukit yang ada }
file_data      : text;
    { data disimpan dalam file teks bukit.dat }
selisih        : longint;

begin
  clrscr; { hapus layar biar bersih }

  assign(file_data, 'bukit.dat');
  reset(file_data);
  { membuka file data }

  tertinggi := 0;
  { nilai tertinggi diisi dengan nilai sekecil mungkin }

  terendah := 300000;
  { nilai terendah diisi dengan nilai sebesar mungkin }

  while not eof(file_data) do
  { membaca data dari awal sampai data terakhir }

    begin
      readln(file_data, bukit);
      writeln(bukit);
      { hanya membaca satu variabel setiap kali proses baca data }

      if (bukit > tertinggi) then tertinggi := bukit;
      { jika ada bukit yang lebih tinggi maka nilai ditukar }

      if (bukit < terendah) then terendah := bukit;
      { jika ada bukit yang lebih rendah maka nilai ditukar }
    end
  end

```

```

    end;
close(file_data);
selisih :=tertinggi-terendah;
{ selisih tinggi bukit terbesar adalah dengan mencari
  bukit tertinggi dan bukit terendah kemudian dikurangkan }

writeln('tertinggi   : ',tertinggi);
writeln('terendah   : ',terendah);
writeln('Perbedaan tertinggi adalah ',selisih);
{ ini hasilnya bro ... }

end.
{ is it too easy problem ? }

```

File data berupa file teks dengan nama bukit.dat yang isinya sama dengan contoh data di atas.

10 Bilangan Fibonacci

Soal ini diambil dari soal olimpiade komputer. Isi dari soal tersebut seperti pada tulisan berikut ini.

Barisan bilangan Fibonacci didefinisikan secara rekursif sebagai berikut:

$$F(n) = \begin{cases} 0, & \text{jika } n = 0 \\ 1, & \text{jika } n = 1 \\ F(n-1) + F(n-2) & \text{jika } n > 1 \end{cases}$$

Buatlah sebuah program yang menentukan apakah $(F(1) + F(2) + \dots + F(n-1) + F(n) + 1)$ habis dibagi oleh $F(M)$.

10.1 Masukan

Masukan dibaca dari standar masukan. Baris pertama berisi bilangan bulat T ($1 \leq T \leq 1000000$) yang menyatakan jumlah kasus uji dalam masukan. Baris kedua sampai baris ke $T+1$ berisi dua buah bilangan bulat N dan M yang dipisahkan oleh sebuah spasi. ($1 \leq N \leq 100000$, $1 \leq M \leq 100000$). Contoh :

```
2
1 1
3 3
```

10.2 Keluaran

Keluaran ditulis ke standar keluaran. Untuk setiap kasus uji, keluaran hanya berupa sebuah baris dengan format Kasus #X: A dimana X menyatakan nomor kasus uji, dimulai dari satu, dan A adalah string habis dibagi jika $(F(1) + F(2) + \dots + F(n-1) + F(n) + 1)$ habis dibagi oleh $F(M)$ dan tidak habis dibagi jika tidak.

```
Kasus #1: habis dibagi
Kasus #2: tidak habis dibagi
```

10.3 Penyelesaian

Contoh penyelesaian untuk masalah bilangan Fibonacci tersebut bisa dilihat pada program berikut ini. Seperti biasanya, program ini sekedar contoh, sangat terbuka peluang untuk mencari dan menemukan solusi yang lebih baik dari ini. Apalagi yang tidak lebih baik, lebih banyak lagi peluangnya ... :-)

```
{ -----
program untuk menyelesaikan soal olimpiade tentang
fibonacci.
mulai ditulis 19 juli 2010 - dwi sakethi
http://dwijim.staff.unila.ac.id
dwijim@unila.ac.id - 0816 403 432
nama file   : fibo.pas
file data  : fibo.in (file data ini harus dibuat)
file hasil : fibo.out
----- }

program barisan_fibonacci;
{ awal program sebaiknya dimulai dengan program bla bla bla }

uses crt,dos;
{ unit yang digunakan sesuai perintah yang ada di program }
```

```

var file_data,file_hasil : text;
  { file_data untuk mengakses data pada file format text
    dan file_hasil untuk menyimpan hasil pengolahan
  }

isi_data : string;
  { isi file data dianggap bertipe string meski isinya integer
    ini disebabkan satu baris ada yang satu data ada yang
    data, jadi dianggap string kemudian dipotong-potong }

data_ke : byte;
  { untuk menyimpan data 1, 2, 3 dst }

jumlah_data : byte;
  { untuk menyimpan banyaknya data fibonacci yang akan dicek }

kode_error : integer;
  { untuk menyimpan kode error konversi string ke numerik }

bilangan_M : byte;
  { nilai data M yang jadi penentu data fibonacci }

str_M : string;
  { nilai data sementara M, masih berupa string }

bilangan_N : byte;
  { nilai data N yang jadi penentu data fibonacci }

str_n : string;
  { nilai data sementara N, masih berupa string }

panjang : byte;
  { panjang isi data karena dianggap string}

posisi_spasi : byte;
  { pembatas antara nilai N dan M }

suku_ke : integer;
  { suku ke untuk f(n) }

jumlah_bilangan : integer;

```

```

{ jumlah f(1)+f(2)+ ... }

hasil : integer;
{ hasil pembagian }
hr1, min1, se1, se2 : word;
{ untuk menyimpan nilai-nilai waktu }

{ fungsi untuk membuat bilangan fibonacci }
function fibonacci(bilangan:integer):integer;
var suku_n : integer;
begin
  case bilangan of
    0 : fibonacci:=0;
    1 : fibonacci:=1;
  else
    fibonacci := fibonacci(bilangan-1)+fibonacci(bilangan-2);
  { bingung juga dengan rekursif,
    searching di internet, mendapatkan contoh fibonacci
  }
  end;
end;

{ waktu mulai }
procedure StartClock;
begin
  GetTime (hr1,min1,se1,se2);
end;

{ waktu selesai }
procedure StopClock;
var
  siz : longint;
  hr2, min2, s1,s2 : word;
begin
  GetTime (hr2, min2, s1, s2);
  siz := (s2-se2)+(s1-se1)+(min2-min1)*60+(hr2-hr1)*60*60;
  writeln('waktu : ', siz , ' detik');
end;

begin

```

```

clrscr;
StartClock;

assign(file_hasil,'fibo.out');
{ membuka file data bernama fibo.out }

rewrite(file_hasil);
{ karena untuk ditulisi jadi rewrite }

assign(file_data,'fibo.in');
{ membuka file data bernama fibo.in }

reset(file_data);
{ karena akan membaca jadi reset }

data_ke := 1;
{ awal data diisi dengan 1 }

while not eof(file_data) do
{ membaca data dari awal sampai akhir }

    begin
        readln(file_data,isi_data);
{
        writeln(isi_data);
        }
        if (data_ke=1) then
            begin
                { isi data pertama dari file teks adalah
                banyaknya data yang akan dicek,
                setelah difikir-fikir, ini mungkin tidak
                diproses juga tidak mengapa,
                karena banyaknya data tidak dibutuhkan,
                untuk menghemat memori}
                val(isi_data,jumlah_data,kode_error);
                { writeln(jumlah_data); }
            end
        else
            begin

                panjang := length(isi_data);

```

```

{ menghitung panjang string isi data }

posisi_spasi := pos(' ',isi_data);
{ mencari pembatasnya ada di kolom berapa }

str_n := copy(isi_data,1,posisi_spasi-1);
val(str_n,bilangan_N,kode_error);
{ menentukan nilai N }

str_m := copy(isi_data,posisi_spasi+1,panjang-posisi_spasi);
val(str_n,bilangan_M,kode_error);
{ menentukan nilai M }
{ writeln(bilangan_n,' - ',bilangan_m); }

jumlah_bilangan := 0;
for suku_ke := 1 to bilangan_N do
    jumlah_bilangan := jumlah_bilangan + fibonacci(suku_ke);
    { menghitung jumlah dari f(x) }

    jumlah_bilangan := jumlah_bilangan + 1;
    { sesuai soal, jumlah terakhir ditambah 1 }

hasil := jumlah_bilangan mod fibonacci(bilangan_M);
{ membagi antara jumlah bilangan fibonacci+1
  dengan bilangan fibonacci ke M }
if hasil=0 then
    writeln(file_hasil,'Kasus #',data_ke-1,': habis dibagi')
else
    writeln(file_hasil,'Kasus #',data_ke-1,': tidak habis dibagi')
end;
inc(data_ke);
{ data_ke bertambah terus sesuai looping }
end;

close(file_data);
{ setelah selesai, file data ditutup lagi }

close(file_hasil);
{ setelah selesai, file data ditutup lagi }

StopClock;

```

```

{
writeln(fibonacci(bilangan_M),' - ',fibonacci(3));
}
{ untuk mencek apakah fibonaccinya sudah benar }
{
  for data_ke := 1 to 10 do
    write(fibonacci(data_ke),' ');
  }
  { jika hasil sudah benar, maka ini bisa di-remark saja }
readln;
end.

```

11 Menjumlah Dua Bilangan Panjang

Ini soal untuk latihan saja. sehingga soalnya juga sangat mudah untuk dikerjakan. Soal ini sudah dibahas di Jurusan Matematika FMIPA Unila pada hari Sabtu, 19 Maret 2011, hanya beberapa saat menjelang balapan perdana MotoGP di Losal Qatar.

Bagaimana membuat program untuk menjumlah dua bilangan yang mempunyai digit banyak. Banyak di sini artinya sudah melebihi batas kapasitas long integer bahkan tipe extended atau comp. Extended atau comp hanya mampu menampung sampai dengan paling banyak 20 digit. Misalnya ada 2 bilangan yang terdiri dari angka sebanyak 150 digit. Bilangan berisi nilai bilangan yang bulat. Bagaimana menjumlah dua bilangan tersebut.

11.1 Contoh hasil eksekusi program

Masukkan bilangan pertama :123456789

Masukkan bilangan kedua :1234

123456789

1234

Hasil : 1234571023

Contoh lain :

Masukkan bilangan pertama :1235678907645452345346624465654664364563453

Masukkan bilangan kedua :345234523450239485234534598723948572398457923847598234

1235678907645452345346624465654664364563453

345234523450239485234534598723948572398457923847598234

Hasil : 345234523451475164142179104101069295196864112588212161687

11.2 Contoh Penyelesaian

Berikut ini adalah contoh penyelesaian untuk masalah tersebut.

```
program menjumlah_dua_bilangan_besar;
{ -----
  program untuk menghitung dua buah bilangan
  yang mempunyai digit di atas batas kewajaran,
  misalnya ada bilangan bulat yang mempunyai
  digit sebanyak 150 digit.
  soal ini merupakan salah satu soal yang diberikan
  dalam lomba olimpiade komputer di jurusan
  matematika fmipa unila tahun 2011 yang
  diadakan pada bulan maret 2011.
  program ini mulai dibuat pada tanggal 23 feb 2011
  sambil nunggu seminar hasil yang belum dimulai
  nama file : bilangan_besar.pas
  ----- }

uses crt;
{ unit yang dipakai hanya crt }

var bilangan_1, bilangan_2 : string;
    { bilangan dideklarasikan sebagai suatu string,
      nanti dikonversi ke integer dan menjumlahkannya
      dimulai dari belakang,
      mirip cara menjumlahkan seperti ketika anak-anak
      sedang belajar penjumlahan :

      338293849349123847928347923841
        982342349234923492834928347
      ----- +
      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx }

    hasil_penjumlahan : array[1..250] of byte;
    { hasilnya berupa bilangan kecil-kecil yang berderet
      dari kanan ke kiri }

    penghitung_looping : byte;
    { variabel untuk looping }

    panjang_1, panjang_2 : integer;
```

```

{ banyaknya digit masing-masing string bilangan }

jumlah_digit_bilangan : integer;
{ digit bilangan yang paling besar di antara dua bilangan }

selisih_digit : byte;
{ selisih digit di antara dua bilangan }

simpan : byte;
{ jika hasil penjumlahan lebih dari 9, maka menyimpan angka 1 }

sisa : byte;
{ jika penjumlahan lebih dari sembilan, setelah menyimpan 1,
  maka ini adalah sisanya }

hasil_akhir : string;
{ hasil akhir penjumlahan berupa string }

kode_error : integer;
{ prosedur val memerlukan kode error }

bil_1,bil_2 : byte;
{ nilai dari suku ke, dalam bentuk numerik }

suku_ke_bil_1, suku_ke_bil_2 : string;
{ suku ke dari masing-masing bilangan }

hasil_suku_ke : string;
{ hasil penjumlahan suku ke dalam bentuk string }

jumlah : byte;
{ hasil penjumlahan suku ke ... }

begin
  clrscr; { membersihkan layar, biar enak melihatnya }

  write('Masukkan bilangan pertama :');readln(bilangan_1);
  {memasukkan bilangan pertama yang akan dicari jumlahnya}

  write('Masukkan bilangan kedua :');readln(bilangan_2);
  {memasukkan bilangan kedua yang akan dicari jumlahnya}

```

```

panjang_1 := length(bilangan_1);
panjang_2 := length(bilangan_2);
{ menghitung panjang string dari masing-masing bilangan }

if bilangan_1>bilangan_2 then
    jumlah_digit_bilangan:=length(bilangan_1)
else
    jumlah_digit_bilangan:=length(bilangan_2);
{ mencari bilangan yang digitnya paling panjang
  karena ini yang menjadi acuan untuk looping }

selisih_digit := abs(length(bilangan_1)-length(bilangan_2));
{ mencari selisih digit antara kedua bilangan }

{ memberi karakter '0' di depan angka yang pendek }
for penghitung_looping:=1 to selisih_digit do
    begin
        if panjang_2>panjang_1 then
            bilangan_1 := '0'+bilangan_1
        else
            bilangan_2 := '0'+bilangan_2;
        end;
    simpan := 0;
    hasil_akhir := '';
    for penghitung_looping:=jumlah_digit_bilangan downto 1 do
        begin
            suku_ke_bil_1 := copy(bilangan_1,penghitung_looping,1);
            suku_ke_bil_2 := copy(bilangan_2,penghitung_looping,1);
            val(suku_ke_bil_1,bil_1,kode_error);
            val(suku_ke_bil_2,bil_2,kode_error);
            jumlah := bil_1 + bil_2 + simpan;
            if jumlah>10 then
                begin
                    simpan := 1;
                    sisa := jumlah mod 10;
                end
            else
                begin
                    simpan := 0;
                    sisa := jumlah;
                end
            end;
        end;
    end;

```

```

        end;
        str(sisa,hasil_suku_ke);
        hasil_akhir := hasil_suku_ke + hasil_akhir;

    end;
    writeln(bilangan_1);
    writeln(bilangan_2);
    writeln('Hasil : ',hasil_akhir);
end.

```

12 Simulasi Program Kasir

Ketika ada pelanggan yang berbelanja ke mini market atau super market, biasanya sudah tersedia suatu sistem untuk melakukan transaksi. Salah satu fasilitas dalam proses transaksi tersebut adalah menampilkan angka dalam bentuk yang besar supaya mudah dilihat oleh pelanggan. Nah ... pada kesempatan ini, buatlah program untuk melakukan simulasi pada kasir.

12.1 Contoh hasil eksekusi program

Total belanja : 5500

```

55555  55555  00000  00000
5      5      0  0  0  0
  5555  5555  0  0  0  0
    5    5    0  0  0  0
55555  55555  00000  00000

```

Uang pembayaran : 10000

```

1      00000  00000  00000  00000
1      0  0  0  0  0  0  0  0
1      0  0  0  0  0  0  0  0
1      0  0  0  0  0  0  0  0
11111  00000  00000  00000  00000

```

Uang kembali :

```

4 4 55555 00000 00000
4 4 5      0 0 0 0
44444 5555 0 0 0 0
      4      5 0 0 0 0
      4 55555 00000 00000

```

12.2 Contoh Penyelesaian

Contoh dari penyelesaian masalah tersebut, dapat dipelajari pada program berikut.

```

{ program simulasi kasir,
  untuk mencetak angka dalam tulisan besar
  mulai diketik hari kamis 10 maret 2011
  nama file : simulasi-kasir.pas
}
program SimulasiKasir;

uses crt;

var Belanja : longint;
    { total belanja yang harus dibayar }

    Kembalian : longint;
    { uang kembalian dari pembayaran }

    UangPembayaran : longint;
    { uang yang dibayarkan oleh pembeli }

{ prosedur untuk membuat tulisan angka dalam ukuran besar
  supaya bisa dilihat oleh pembeli }
procedure tulisanbesar(bilangane:longint);

{ mendeklarasikan masing-masing bentuk angka sebagai
  pengganti angka ukuran biasa 0, 1, 2, ..., 9 }
const
bilangan01 = '00000  ';
bilangan02 = '0  0  ';
bilangan03 = '0  0  ';

```

```
bilangan04 = '0 0 ' ;
bilangan05 = '00000 ' ;

bilangan11 = ' 1 ' ;
bilangan12 = ' 1 ' ;
bilangan13 = ' 1 ' ;
bilangan14 = ' 1 ' ;
bilangan15 = '11111 ' ;

bilangan21 = '22222 ' ;
bilangan22 = ' 22 ' ;
bilangan23 = ' 2 ' ;
bilangan24 = '22 ' ;
bilangan25 = '22222 ' ;

bilangan31 = '33333 ' ;
bilangan32 = ' 3 ' ;
bilangan33 = ' 333 ' ;
bilangan34 = ' 3 ' ;
bilangan35 = '33333 ' ;

bilangan41 = '4 4 ' ;
bilangan42 = '4 4 ' ;
bilangan43 = '44444 ' ;
bilangan44 = ' 4 ' ;
bilangan45 = ' 4 ' ;

bilangan51 = '55555 ' ;
bilangan52 = '5 ' ;
bilangan53 = ' 5555 ' ;
bilangan54 = ' 5 ' ;
bilangan55 = '55555 ' ;

bilangan61 = '666 ' ;
bilangan62 = '6 ' ;
bilangan63 = '66666 ' ;
bilangan64 = '6 6 ' ;
bilangan65 = '66660 ' ;

bilangan71 = '77777 ' ;
bilangan72 = ' 7 ' ;
```

```

bilangan73 = ' 7  ' ;
bilangan74 = ' 7  ' ;
bilangan75 = '77  ' ;

bilangan81 = '88888  ' ;
bilangan82 = '8 8  ' ;
bilangan83 = '88888  ' ;
bilangan84 = '8 8  ' ;
bilangan85 = '88888  ' ;

bilangan91 = '99999  ' ;
bilangan92 = '9 9  ' ;
bilangan93 = '99999  ' ;
bilangan94 = ' 9  ' ;
bilangan95 = '99999  ' ;

banyaknyabaris = 5;

var PenghitungLooping : byte;
    StringBilangane    : string;
    PanjangBilangane   : byte;
    SukuKe              : string;
    baris               : byte;
    baris1,baris2,baris3,baris4,baris5 : string;
begin
    { memberi nilai awal kosong kepada string hasil akhir
      melihat karakter pengganti yang terdiri dar 5 baris
      maka hasil akhir juga ada 5 baris
    }
    baris1 := '';
    baris2 := '';
    baris3 := '';
    baris4 := '';
    baris5 := '';

    { bilangan dikonversi ke bentuk string supaya mudah
      mengambil karakter ke-1, 2, dst }
    str(bilangane,StringBilangane);

    { menghitung panjang string bilangan}
    PanjangBilangane := length(StringBilangane);

```

```

{ looping dari baris 1 sampai dengan baris 5 }
for baris:=1 to BanyaknyaBaris do
begin
for PenghitungLooping:=1 to PanjangBilangane do
  begin

    { menentukan sekarang sedang memproses karakter apa ?
      karakter 0, 1, 2 , 3, ... 9 }
    SukuKe := copy(StringBilangane,PenghitungLooping,1);

    case baris of
      1 : { baris ke satu }
        if SukuKe='1' then
          baris1 := baris1 + bilangan1
          { jika sekarang sedang memproses karakter '1' dan
            posisi berapa pada posisi 1, maka hasilnya adalah
              ' 1      ' demikian seterusnya }
        else
          if sukuke='2' then
            baris1 := baris1 + bilangan21
            { jika sekarang sedang memproses karakter '2' dan
              posisi berapa pada posisi 1, maka hasilnya adalah
                '22222  ' demikian seterusnya }
          else
            if sukuke='3' then
              baris1 := baris1 + bilangan31
            else
              if sukuke='4' then
                baris1 := baris1 + bilangan41
              else
                if sukuke='5' then
                  baris1 := baris1 + bilangan51
                else
                  if sukuke='6' then
                    baris1 := baris1 + bilangan61
                  else
                    if sukuke='7' then
                      baris1 := baris1 + bilangan71
                    else
                      if sukuke='8' then

```

```

        baris1 := baris1 + bilangan81
    else
    if sukuke='9' then
        baris1 := baris1 + bilangan91
    else
    if sukuke='0' then
        baris1 := baris1 + bilangan01;
2 : { baris ke dua }
    if sukuke='1' then
        baris2 := baris2 + bilangan12
    else
    if sukuke='2' then
        baris2 := baris2 + bilangan22
    else
    if sukuke='3' then
        baris2 := baris2 + bilangan32
    else
    if sukuke='4' then
        baris2 := baris2 + bilangan42
    else
    if sukuke='5' then
        baris2 := baris2 + bilangan52
    else
    if sukuke='6' then
        baris2 := baris2 + bilangan62
    else
    if sukuke='7' then
        baris2 := baris2 + bilangan72
    else
    if sukuke='8' then
        baris2 := baris2 + bilangan82
    else
    if sukuke='9' then
        baris2 := baris2 + bilangan92
    else
    if sukuke='0' then
        baris2 := baris2 + bilangan02;
3 : { baris ke tiga }
    if sukuke='1' then
        baris3 := baris3 + bilangan13
    else

```

```

if sukuke='2' then
    baris3 := baris3 + bilangan23
else
if sukuke='3' then
    baris3 := baris3 + bilangan33
else
if sukuke='4' then
    baris3 := baris3 + bilangan43
else
if sukuke='5' then
    baris3 := baris3 + bilangan53
else
if sukuke='6' then
    baris3 := baris3 + bilangan63
else
if sukuke='7' then
    baris3 := baris3 + bilangan73
else
if sukuke='8' then
    baris3 := baris3 + bilangan83
else
if sukuke='9' then
    baris3 := baris3 + bilangan93
else
if sukuke='0' then
    baris3 := baris3 + bilangan03;
4 : { baris ke empat }
if sukuke='1' then
    baris4 := baris4 + bilangan14
else
if sukuke='2' then
    baris4 := baris4 + bilangan24
else
if sukuke='3' then
    baris4 := baris4 + bilangan34
else
if sukuke='4' then
    baris4 := baris4 + bilangan44
else
if sukuke='5' then
    baris4 := baris4 + bilangan54

```

```

else
if sukuke='6' then
    baris4 := baris4 + bilangan64
else
if sukuke='7' then
    baris4 := baris4 + bilangan74
else
if sukuke='8' then
    baris4 := baris4 + bilangan84
else
if sukuke='9' then
    baris4 := baris4 + bilangan94
else
if sukuke='0' then
    baris4 := baris4 + bilangan04;
5 : { baris ke lima }
if sukuke='1' then
    baris5 := baris5 + bilangan15
else
if sukuke='2' then
    baris5 := baris5 + bilangan25
else
if sukuke='3' then
    baris5 := baris5 + bilangan35
else
if sukuke='4' then
    baris5 := baris5 + bilangan45
else
if sukuke='5' then
    baris5 := baris5 + bilangan55
else
if sukuke='6' then
    baris5 := baris5 + bilangan65
else
if sukuke='7' then
    baris5 := baris5 + bilangan75
else
if sukuke='8' then
    baris5 := baris5 + bilangan85
else
if sukuke='9' then

```

```

        baris5 := baris5 + bilangan95
    else
        if sukuke='0' then
            baris5 := baris5 + bilangan05;
        end;
    end;
end;
end;
end;
writeln;
writeln;
writeln(baris1); { mencetak hasil string gabungan
                  pada posisi paling atas }

writeln(baris2);
writeln(baris3);
writeln(baris4);
writeln(baris5); { mencetak hasil string gabungan
                  pada posisi paling bawah }

writeln;
writeln;
end;
{ akhir prosedur angka besar }

{ awal program utama }
begin
    clrscr; { menghapus layar }
    write('Total belanja : ');readln(belanja);

    { proses pemasukan data }
    tulisanbesar(belanja);

    write('Uang pembayaran : ');readln(UangPembayaran);

    { proses pemasukan data }
    tulisanbesar(UangPembayaran);

    kembalian := UangPembayaran - belanja;
    writeln('Uang kembali : ');
    tulisanbesar(kembalian);
end.
{ akhir program utama }

```

13 Program Game Menembak Versi Text Base

Buatlah program untuk melakukan simulasi permainan menembak berbasis teks. Pemain hanya bisa menggeser alat tembak ke kiri atau ke kanan. Kemudian target diletakkan secara acak. Jarak tembak berkisar antara 10-15. Semakin jauh jaraknya, jika kena maka nilainya semakin besar. Sedangkan lama waktu menembak tidak mempengaruhi penilaian. Satu target hanya diberi jatah 10 peluru. Ketepatan sasaran antara peluru pertama dengan peluru berikutnya, memberikan skor yang berbeda, demikian seterusnya. Untuk proses penembakan diberikan animasi peluru yang bergerak dari senjata menuju target.

13.1 Contoh Penyelesaian

Contoh dari penyelesaian masalah tersebut, dapat dipelajari pada program berikut.

```
{
    contoh program membuat tembak-tembakkan berbasis teks biasa
    nama file : tembakan.pas
}
```

```
program TembakTembakan;

uses crt;

const senjata = '^';
      { simulasi bentuk senjatanya }

      peluru1 = '*';
      { simulasi bentuk peluru }

      peluru2 = ' ';
      { untuk menghapus bekas jejak peluru }

BarisSenjata = 22;

TengahLayar = 40;

TombolM      = 'm'; { geser kanan }
TombolN      = 'n'; { geser kiri }
```

```

TombolX      = 'x'; { keluar }
TombolZ      = 'z'; { menembak }

BatasKiri    = 5;   { batas kiri dan kanan posisi senjata }
BatasKanan   = 75;

BatasAtasTarget = 3;

var tombol :char;

KolomSenjata : byte;
{ posisi kolom dari senjata, karena geser kiri-kanan
  maka kolom ini bisa berubah-ubah nilainya.
  akan tetapi posisi baris tidak berubah-ubah}

{ fungsi untuk melakukan penembakan pada posisi
  baris dan kolom tertentu }
function menembak(kolomx,barisx,KolomTarget,BarisTarget:byte):boolean;
var barisnya : byte;
begin
  for barisnya:=barisx downto BatasAtasTarget do
    begin
      { menghapus peluru pada baris sebelumnya }
      gotoxy(kolomx,barisnya+1);write(peluru2);

      { menggambar peluru pada posisi sekarang }
      gotoxy(kolomx,barisnya);write(peluru1);

      sound(barisnya*11);delay(barisx*5);nosound;

      { jika kena berarti posisi baris dan kolom sama }
      if (BarisTarget=barisx) and (KolomTarget=kolomx) then
        begin
          menembak := true;
          exit;
        end
      else
        menembak := false;
    end;
  { menghapus peluru pada baris sebelumnya }
  gotoxy(kolomx,barisnya);write(peluru2);

```

```

end;

begin
  clrscr;
  KolomSenjata := TengahLayar;
  gotoxy(1,24);write('Tombol : m-kanan n-kiri z-tembak x-keluar');
  repeat

    { gotoxy(75,20);writeln(tombol); }

    gotoxy(KolomSenjata,BarisSenjata);write(senjata);

    tombol := readkey;
    if tombol=TombolM then
      begin
        { jika menekan tombol M berarti senjata digeser
          ke kanan }
        KolomSenjata := KolomSenjata + 1;

        { jika sudah mencapai batas kanan maka jangan
          digeser ke kanan lagi }
        if KolomSenjata>=BatasKanan then
          KolomSenjata := BatasKanan;
        end
      end
    else
      if tombol = TombolN then
        begin
          { jika menekan tombol N maka senjata digeser
            ke kiri }
          KolomSenjata := KolomSenjata - 1;

          { jika sudah mencapai batas kiri maka
            jangan digeser ke kiri lagi }
          if KolomSenjata<=BatasKiri then
            KolomSenjata := BatasKiri;
          end
        end
      else
        if tombol = TombolZ then
          begin
            menembak(KolomSenjata+1,BarisSenjata+1,23,7);
            { plus 1 karena bentuk senjatanya ' ^ ' }
          end
        end
      end
    end
  repeat

```

```
        end;  
    until tombol='x';  
end.
```