

## Uppgift 1

Leta med datorns hjälp reda på den längsta *primtalsöken* som finns i intervallet [2,20000].

Ett primtal  $p$  är ett heltal som inte har några andra delare än 1 och  $p$ .

En primtalsöken är en följd av heltal, där inget av talen är primtal. Ex: ... 23 24 25 26 27 28 29 ...  
Inget av talen 24 till 28 är primtal. Denna primtalsöken har därför längden 5. Primtalen 23 och 29 bildar öknens gränser.

*Indata:* -

*Utdata:* Den längsta primtalsökens gränser och längd.

## Uppgift 2

*Fredagen den 13:e* betraktas av många som den verkliga otursdagen. Hur många sådana otursdagar har vi att vänta oss, från och med *januari 1991* till och med *december 2010*? Skriv ett program som på skärmen listar årtal (1991-2010) och därefter numret på de månader under året som innehåller fredagen 13:e.

- Den 1 januari 1991 var en *tisdag*.
- Ett år är *skottår* om årtalet är *jämnt delbart med 4*, utom de *sekularår*, hela århundraden, där århundradet *ej är jämnt delbart med 400*. Sålunda var 1600 skottår men inte 1800 och 1900.
- Trettio dagar har november, april, juni och september  
Februari, tjugoåtta allen  
Alla de övriga trettioen

*Indata:* -

*Utdata:* En rad för varje år från 1991 till 2010, följt av aktuella månadsnummer.

# Uppgift 3

$$3 * (45 + 23 / 28) - 45$$

är exempel på ett aritmetiskt uttryck. Denna uppgift går ut på att skriva ett program som beräknar just aritmetiska uttryck.

*Förutsättningar:*

Programmet får förutsätta att uttrycket är korrekt inskrivet enligt de regler som ges nedan.

- Indatasträngen får endast innehålla följande tecken [0..9,+,\*]
- En konstant får endast bestå av en siffra [0..9]
- Programmet ska givetvis, vid beräkningen, ta hänsyn till prioriteringsregeln: *att multiplikation går före addition.*
- Ingen hänsyn behöver tas till eventuell *overflow*, dvs alla resultat och mellanresultat förväntas ligga i intervallet [0,32767]

*Några exempel:*

- $2 + 3 * 4$  korrekt
- $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9$  korrekt
- $1 * 2 * 3 * 4 + 5 * 6 * 7 * 8$  korrekt
- $12 + 3 * 4$  felaktig, konstanten 12 är för stor
- $+3 + 4$  felaktig, plustecknet används enbart som operator

*Indata:* En sträng på maximalt 80 tecken som följer reglerna ovan.

*Utdata:* Ett heltal som motsvarar värdet av uttrycket.

## Uppgift 4

```
PROGRAM permutation;
TYPE str6=STRING[6];
VAR inord:str6;
PROCEDURE permutera(s:str6; k:integer);
  VAR i:integer;
      temp:char;
  BEGIN
    IF k=6 THEN
      writeln(s)
    ELSE
      FOR i:=k TO 6 DO BEGIN
        temp:=s[k]; s[k]:=s[i]; s[i]:=temp;
        permutera(s,k+1);
      END;
    END;
  BEGIN
    write('Skriv in 6 versaler: '); readln(inord);
    permutera(inord,1);
  END.
```

I programmet ovan finns en rekursiv procedur som genererar och skriver ut samtliga permutationer till en inmatad sträng innehållande sex tecken. ABCFDE och FEDCBA är två av de 720 (6!) permutationer man kan erhålla ur indatasträngen ABCDEF.

Uppgiften går ut på att generera alla 720 permutationerna till en inmatad bokstavssträng och bland dessa skriva ut de som följer reglerna nedan:

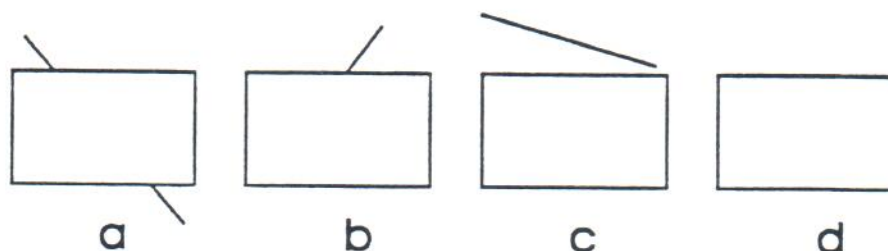
- Två *vokaler* får aldrig stå intill varandra
- Tre *konsonanter* får aldrig stå intill varandra
- Om två *konsonanter*, som inleder ett ord ska accepteras, måste kombinationen finnas i listan: *ST SK TR GR FR KR PR*.
- Om ordet innehåller två *konsonanter* som står intill varandra, där den första inte inleder ordet måste kombinationen finnas i listan: *ST SK TR GR FR KR PR TT NG LL ND NS CH RN RS TS SS KT NT RB*
- Vårt alfabet innehåller 29 bokstäver, W medtaget.
- Följande bokstäver räknas till *vokaler*: A E I O U Y Å Ä Ö
- Övriga bokstäver är *konsonanter*.

*Indata*: En sträng som innehåller sex stora bokstäver (versaler) ur vårt alfabet. Samma bokstav får förekomma flera gånger.

*Utdata*: En lista över ord bildade med hjälp av indatasträngen och som följer de regler som givits ovan. OBS Samma ord får bara förekomma en gång i utskriften. Till sist ett tal som anger hur många ord som skrivits ut.

## Uppgift 5

Uppgiften består i att på skärmen, i grafisk mod, rita en rektangel och två linjesegment som eventuellt kan vara skymda av rektangeln.



Bilderna ovan visar de fyra möjligheterna:

- a) Linjesegmentet går genom rektangeln. Båda ändpunkterna ligger utanför rektangeln.
- b) Linjesegmentet slutar någonstans inuti rektangeln. Endast en ändpunkt ligger utanför rektangeln.
- c) Linjesegmentet ligger helt utanför rektangeln. Båda ändpunkterna ligger utanför rektangeln.
- d) Linjesegmentet ligger helt inuti rektangeln. Båda ändpunkterna ligger inuti rektangeln.

För enkelhet skall användas det koordinatsystem som pascalsystemet erbjuder. För TurboPascal gäller alltså att origo finns högst upp till vänster på skärmen. Indata ges som heltal och får förutsättas ligga i det intervall som ges av aktuell grafikmod, tex: VGAhi (0..639,0..479), EGAhi (0..639,0..349)

Allt ritande ska göras med vit färg på svart bakgrund. Inga ritade linjer får tas bort. Rittekniken ska efterlikna den vid användning av plotter.

*Indata:* a) Koordinaten för rektangelns övre vänstra hörn. b) Rektangelns bas, (utbredning i horisontell led) c) Rektangelns höjd, (utbredning i vertikal led). d) Första linjesegmentets ena ändpunkt e) Första linjesegmentets andra ändpunkt f) Andra linjesegmentets ena ändpunkt g) Andra linjesegmentets andra ändpunkt.

*Utdata:* En grafisk bild som visar rektangel och linjesegment som helt eller delvis kan vara skymda av rektangeln. Låt programmet visa bilden tills operatören trycker RETURN.

## Uppgift 6

Uttrycket  $123+4-5+67-89$  är lika med 100. Det finns ytterligare 10 möjligheter att sätta in ett antal plus- och minustecken i sifferföljden 1 2 3 4 5 6 7 8 9 så att summan blir 100. Skriv ett program som listar alla 11 lösningarna.

*Indata:* -

*Utdata:* En lista över de 11 lösningarna på formen:  $123+4-5+67-89$