Aufgabe 1: Druckerwarteschlange

Wer kennt das Problem nicht? Sie wollen nur schnell ein paar Zeilen drucken, aber der Drucker ist noch auf Stunden mit dem Druck vieler langer Produktionsaufträge beschäftigt.

Hier muß eine Warteschlangenverwaltung her, die die Einreihung nach der zu erwartenden Druckzeit vornimmt. Maßgeblich ist die Anzahl der zu druckenden Zeilen, der Aufwand für Formularwechsel und der Zeitpunkt des Eintritts in die Warteschlange, um Alterung durch Wartezeit berücksichtigen zu können. Die kontinuierlich ablaufende Zeit wird in Takte eingeteilt, wobei in jedem Takt 100 Zeilen oder eine Trennseite gedruckt werden können. Normalerweise wird auf Blankopapier gedruckt. Bei Anforderung eines Formulars sind je 3 Zeittakte vor und nach dem Druck für den Papierwechsel zu veranschlagen. Der Einfachheit halber kann in jedem Zeittakt maximal ein neuer Auftrag in die Warteschlange eingereiht werden.

Die Warteschlange ist durch eine einfach verkettete geschlossene Liste zu realisieren. Die geschlossene Liste bietet sich zur Vereinfachung an, weil die Warteschlange auch zwischendurch leer werden kann und so die Abhandlung der Sonderfälle vermieden wird.

Die auf Druck wartenden Aufträge werden durch Zeilen in der Standardeingabe dargestellt, wobei in jeder Zeile der Zeittakt des Eintritts in die Warteschlange, die Auftragsnummer, das gewünschte Formular (s=Standard (Formularnummer 0), f=besonderes Formular mit Formularnummer größer als Null) und die zu druckenden Zeilen eingetragen sind. Die erste Spalte ist aufsteigend sortiert, wobei jede Zahl maximal einmal auftritt. Die zweite Spalte ist beginnend bei 1 fortlaufend numeriert (Auftragsnummer).

Bei der Eingabe von

```
2  1  s  220
3  2  f  2  510
4  3  f  2  100
11 4  s  100
```

sollte die Ausgabe etwa folgendermaßen aussehen:

```
Takt 1  Formular 0: Warteschlange leer
Takt 2  Formular 0: Auftragsnummer 1, Trennblatt gedruckt
Takt 3  Formular 0: Auftragsnummer 1, 100 Zeilen gedruckt
Takt 4  Formular 0: Auftragsnummer 1, 100 Zeilen gedruckt
Takt 5  Formular 0: Auftragsnummer 1, 20 Zeilen gedruckt
Takt 6  Formular 0: Auftragsnummer 3, Formular 2 geholt
Takt 7  Formular 0: Auftragsnummer 3, Formular 0 entfernt
Takt 8  Formular 2: Auftragsnummer 3, Formular 2 eingelegt
Takt 9  Formular 2: Auftragsnummer 3, Trennblatt gedruckt
Takt 10 Formular 2: Auftragsnummer 3, 100 Zeilen gedruckt
Takt 11 Formular 2: Auftragsnummer 2, Trennblatt gedruckt
```

```
Takt 15 Formular 2: Auftragsnummer 2, 100 Zeilen gedruckt
Takt 16 Formular 2: Auftragsnummer 2, 100 Zeilen gedruckt
Takt 17 Formular 2: Auftragsnummer 2, 10 Zeilen gedruckt
Takt 18 Formular 2: Auftragsnummer 4, Formular 0 geholt
Takt 19 Formular 2: Auftragsnummer 4, Formular 2 entfernt
Takt 20 Formular 0: Auftragsnummer 4, Formular 0 eingelegt
Takt 21 Formular 0: Auftragsnummer 4, Trennblatt gedruckt
Takt 22 Formular 0: Auftragsnummer 4, 100 Zeilen gedruckt
Takt 23: Feierabend !
```

Als Lösungsanleitung wird das folgende Programmskelett vorgegeben:

```
program drucker ( input , output );
   type natuerlich = 0..maxint ;
        zeiger = ^auftrag ;
        auftrag = record
                   auftrag : natuerlich ;       { Auftragsnummer }
                   form : natuerlich ;          { Formularnummer }
                   formwechsel : natuerlich ;   { für Formulare }
                   zeilen : integer ;           { Anzahl der zu druckenden Zeilen }
                   eintrittszeit : natuerlich ;{ Zeittakt des Eintritts in
                                                  die Warteschange }
                   next : zeiger                { Zeiger auf den nächsten
                                                  in der Warteschange }
                end;
   var takt : natuerlich ;           { globale Variable für den Zeittakt }
       formular : natuerlich ;       { globale Variable für aktuelles Formular }
       Warteschlange,                { Zeiger auf die Warteschlange }
       neu : zeiger ;                { Zeiger auf neuen Auftrag }

   function erzeugewarteschlange : zeiger;
   { erzeugt leere geschlossenen Liste und gibt den Zeiger auf das
     spezielle Kopfelement, das selbst keine Information enthält, zurück }
   . . .

   function neuerAuftrag: zeiger;
   { liest eine Zeile aus der Standardeingabe,
     erzeugt einen Record vom Typ Auftrag und füllt die Info-Felder,
     gibt den Zeiger auf den Record zurück }
   . . .

   function suchevor ( liste, neu : zeiger ): zeiger ;
   { 1. Parameter ist der Zeiger auf die Warteschlange,
     2. Parameter ist der Zeiger auf den neu einzufügenden Auftrag,
     liefert den Zeiger auf den Auftrag in der Warteschlange, hinter dem
     der neue Auftrag einzureihen ist, als Ergebnis zurück }
   . . .
       function prio ( auftrag: zeiger ): integer ;
       { berechnet die Priorität des durch auftrag bezeichneten Record
         bezüglich der Einordnung in die Warteschlange unter
```

```
        Berücksichtigung der Wartezeit.
        kleine Zahl bedeutet hohe Priorität }
      . . .
  . . .

  procedure einfuegenach( vor, neu :zeiger);
  { fügt den durch neu bezeichneten Auftrag hinter dem durch vor bezeichneten
    Auftrag in die Warteschlange ein }
  . . .

  procedure Zeittakt ( liste :zeiger);
  { aktualisiert die durch liste bezeichnete Warteschlange und gibt die
    Aktion im aktuellen Zeittakt auf die Standardausgabe aus,
    erhöht den Zeittakt um 1 (globale Variable takt)}
    . .
     procedure loescheAuftrag( liste :zeiger);
     { löscht den ersten Auftrag in der durch liste bezeichneten
     Warteschlange }
     . . .
    . .

begin { Hauptprogramm }
  . . .
end .
```

Schreiben Sie das Hauptprogramm unter der Voraussetzung, daß die im Programmskelett aufgeführten Prozeduren und Funktionen vorgegeben sind und die im Kommentar jeweils angegebenen Aktionen ausführen. Die Funktion *prio* ist lokal zur Funktion suchevor und die Prozedur *loescheAuftrag* ist lokal zur Prozedur ***zeittakt.***

Die Verarbeitung wird durchgeführt, bis die Standardeingabe erschöpft ist und alle Aufträge bearbeitet sind.

Benotung: maximal 20 Punkte

Aufgabe 2: Erzeugen der leeren Warteschlange

Schreiben Sie die Funktion *erzeugeWarteschlange* zu Aufgabe 1.

```
function erzeugewarteschlange :zeiger;
{ erzeugt eine leere geschlossenen Liste und gibt den Zeiger auf das
  spezielle Kopfelement zurück }
```

Benotung: maximal 5 Punkte

Aufgabe 3: Einlesen der Daten eines neuen Druckauftrags

Schreiben Sie die Funktion *neuerAuftrag* zu Aufgabe 1.

```
function neuerAuftrag : zeiger ;
{ liest eine Zeile aus der Standardeingabe,
  erzeugt einen Record vom Typ Auftrag und füllt die Info-Felder,
  gibt den Zeiger auf den Record zurück }
```

Die auf Druck wartenden Aufträge werden durch Zeilen in der Standardeingabe dargestellt, wobei in jeder Zeile der Zeittakt des Eintritts in die Warteschlange, die Auftragsnummer, das gewünschte Formular (s=Standard (Formularnummer 0), f=besonderes Formular mit Formularnummer größer als Null) und die zu druckenden Zeilen eingetragen sind. Die erste Spalte ist aufsteigend sortiert, wobei jede Zahl maximal einmal auftritt. Die zweite Spalte ist beginnend bei 1 fortlaufend numeriert (Auftragsnummer), also z.B.:

```
2  1 s 220
3  2 f 2 510
4  3 f 2 100
11 4 s 100
```

Benotung: maximal 10 Punkte

Aufgabe 4: Berechnung der Priorität

Schreiben Sie die Funktion *prio* zu Aufgabe 1.

```
function prio (auftrag:zeiger): integer ;
{ berechnet die Priorität des durch auftrag bezeichneten Record
  bezüglich der Einordnung in die Warteschlange unter
  Berücksichtigung der Wartezeit.
  kleine Zahl bedeutet hohe Priorität }
```

Die Priorität beträgt
1 für Standardpapier (Formularnummer = 0)
7 für Formulardruck (Formularnummer > 0)
Sie erhöht sich um 1 für je 100 Druckzeilen, wobei 1 bis 99 wie 100, 101 bis 199 wie 200, usw. gewertet werden. Um Alterung durch Wartezeit zu berücksichtigen erniedrigt sie sich alle 3 Zeittakte um 1, beginnend mit dem Eintritt in die Warteschlange. Hierzu wird die globale Variable *takt* benötigt.

Benotung: maximal 15 Punkte

Aufgabe 5: Suchen der Einfügestelle

Schreiben Sie die Funktion *suchevor zu* Aufgabe 1.

```
function suchevor ( liste, neu :zeiger ):zeiger ;
{ 1. Parameter ist der Zeiger auf die Warteschlange,
  2. Parameter ist der Zeiger auf den neu einzufügenden Auftrag,
  liefert den Zeiger auf den Auftrag in der Warteschlange, hinter dem
  der neue Auftrag einzureihen ist, als Ergebnis zurück }
```

Diese Funktion benutzt die interne Funktion *prio (s.* Aufgabe 4).

Benotung: maximal 15 Punkte

Aufgabe 6: Einfügen eines Auftrags in die Warteschlange

Schreiben Sie die Prozedur *einfuegenach* zu Aufgabe 1.

```
procedure einfuegenach( vor, neu :zeiger);
{ fügt den durch neu bezeichneten Auftrag hinter dem durch vor bezeichneten
  Auftrag in die Warteschlange ein }
```

Benotung: maximal 5 Punkte

Aufgabe 7: Löschen eines Auftrags aus der Warteschlange

Schreiben Sie die Prozedur *loescheAuftrag* zu Aufgabe 1.

```
procedure loescheAuftrag( liste :zeiger);
{ löscht den ersten Auftrag in der durch liste bezeichneten
Warteschlange }
```

Benotung: maximal 5 Punkte

Aufgabe 8: Übergang zum nächsten Zeittakt

Schreiben Sie die Prozedur *zeittakt* zu Aufgabe 1.

```
procedure zeittakt( liste :zeiger);
{ aktualisiert die durch liste bezeichnete Warteschlange und gibt die
  Aktion im aktuellen Zeittakt auf die Standardausgabe aus,
  erhöht den Zeittakt um 1 (globale Variable takt)}
```

Diese Prozedur benutzt die interne Prozedur *loescheAuftrag* (s. Aufgabe 7).
Die kontinuierlich ablaufende Zeit wird in Takte eingeteilt, wobei in jedem Takt 100 Zeilen oder eine Trennseite gedruckt werden können. Für ein besonderes Formular sind je 3 Zeittakte vor und nach dem Druck zu protokollieren (s. Anlistung auf Seite lf). Werden zwei aufeinander folgende Aufträge auf dasselbe Formularpapier gedruckt, dann entfällt der Formularwechsel dazwischen.

Benotung: maximal 25 Punkte