CLS Cohort Studies

Data Note 2

Pregnancy Histories in the Combined
NCDS/BCS70 1999/2000 Data

Brian Dodgeon

Centre for Longitudinal Studies

Centre for Longitudinal Studies
Institute of Education
20 Bedford Way
London WC1H 0AL
Tel: 020 7612 6860
Fax: 020 7612 6880
Email cls@cls.ioe.ac.uk
Web http://www.cls.ioe.ac.uk

CLS COHORT STUDIES

DATA NOTE 2

Pregnancy Histories in the combined NCDS/BCS70 1999/2000 Dataset

Brian Dodgeon

December 2002

Introduction

This document has been prepared to accompany the second deposit, with the UK Data Archive at the University of Essex, of data from the most recent follow-ups of two continuing, multidisciplinary, national, longitudinal studies – the National Child Development Study (NCDS) and the 1970 British Cohort Study (BCS70). The follow-ups took place between November 1999 and September 2000, and it is noteworthy that this was the first time that both cohorts had been surveyed at the same time. They were designed and implemented jointly by the Centre for Longitudinal Studies of the Institute of Education, University of London (CLS), and the National Centre for Social Research (NatCen), on behalf of the Joint Centre for Longitudinal Research. The work was mainly funded by the Economic and Social Research Council (ESRC), but important contributions were also made by a number of government departments, and by the Basic Skills Agency.

NCDS and BCS70

The National Child Development Study (NCDS) started life as the Perinatal Mortality Survey and examined the social and obstetric factors associated with stillbirth and infant mortality among over 17,000 babies born in Britain in the week 3-9 March 1958. Since this first study the whole cohort have been surveyed on five other occasions in order to monitor their health, education, social and economic circumstances. These surveys were carried out in 1965 (age 7), 1969 (age 11), 1974 (age 16), 1981 (age 23) and 1991 (age 33). As part of the 1991 survey, a special study was also undertaken of the children of one third of the cohort members, including assessments of the behaviour and cognitive development of approximately 5,000 children. There have also been surveys of sub-samples of the cohort, the recent occurring in 1996 (age 37) when information was collected on the basic skills of a representative sample of 10 per cent of cohort members.

The 1970 British Cohort Study (BCS70) was designed along similar lines to the NCDS, surveying over 17,000 babies born in Britain in the week 5-11 April 1970. Since the birth survey there have been four other major data collection exercises in order to monitor their health, education, social and economic circumstances. These were carried out in 1975 (age 5), 1980 (age 10), 1986 (age 16) and 1996 (age 26). As in NCDS, sub-samples have been studied at various ages: for example at age 21, paralleling the NCDS survey at age 37, a 10 per cent representative sample was assessed for basic skills difficulties.

From their original focus on the circumstances and outcomes of birth, the two cohort studies have broadened in scope to map all aspects of health, education and social development of their subjects as they passed through childhood and adolescence. In later sweeps, the information collected has covered their transitions into adult life, including leaving full-time education, entering the labour market, setting up independent homes, forming partnerships and becoming parents.

The latest rounds of data collection for NCDS and BCS70 took place in 1999/2000 when NCDS cohort members were aged 41/42 and BCS70 cohort members were aged

29/30. The main aim of these most recent surveys was to explore the factors central to the formation and maintenance of adult identity in each of the following domains:

- Lifelong learning
- Relationships, parenting and housing
- Employment and income
- Health and health behaviour
- Citizenship and values

Cohort Studies User Support Group

This provides advice and guidance on the use of Cohort Studies data; produces documentation; collates and disseminates information on uses of the data, publications, and other developments; produces and distributes a newsletter and working papers; provides access to non-computerised Cohort Studies data; and collects additional information.

Contacting the User Support Group

The User Support Group can be contacted by post, 'phone, fax, or email as shown below:

Cohort Studies User Support Group,

Centre for Longitidinal Studies,

6th Floor: Institute of Education,

20 Bedford Way,

London WC1H 0AL

Tel: +44 0207 612 6864

Fax: +44 0207 612 6880

Email: cohort@cls.ioe.ac.uk

Internet: http://www.cls.ioe.ac.uk/Cohort/Ncds/mainncds.htm

Background to problem with pregnancy history variables

There are 29 variables relating to each baby ever carried, and 10 others relating to each pregnancy. These variables are structured in such a way as to accommodate 8 possible pregnancies, with up to 5 babies born as a result of each. So each of the 29 'baby' variables are replicated forty times (e.g. birth/termination date **prege, prege2, prege3,…, prege40**), and each of the ten 'pregnancy' variables are replicated eight times (e.g. number of babies carried **pregnum, pregnum2,… , pregnum8**).

The convention is that the most recent pregnancy is entered first, and the others follow in reverse chronological order. For example the variable **prege** refers to the date of birth (or other outcome) of the 1st or only baby resulting from the most recent pregnancy, the variables **prege2-prege5** would refer to any other babies from that same (multiple-birth) pregnancy; then **prege6** refers to a baby resulting from the second most recent pregnancy, **prege11** from the third most recent,… etc. We use the terminology 'outcome date' rather than birth date, as in the case of non-livebirths, the date of miscarriage/stillbirth/abortion is entered.

An apparent error in the setting up of the CAPI instrument for data collection, or a misunderstanding of the nature of the 'number of babies carried' question resulted in the following unforeseen problems:

(A) In a number of both NCDS and BCS70 cases, babies from separate pregnancies were entered as if they were part of a multiple birth: their outcome dates, though clearly more than 9 months apart, were entered, for example, as **prege, prege2, prege3**,… etc., instead of **prege, prege6, prege11**,… etc. In these cases the variable **pregnum** (or in some cases **pregnum2-8)** appears to have been filled in incorrectly, probably as a result of construing the question 'how many babies were carried during this pregnancy?' erroneously as 'how many babies have you ever carried?' This **pregnum** error is probably what caused the automatic CAPI routing to bunch these babies wrongly. See Appendix 1 for a listing of these cases.

(B) Furthermore, in most of these cases, the outcome dates were also in the wrong order, with the oldest first, and the most recent last, or even an irregular order. See Appendix 2.

(C) This reversal or disruption of chronological order also occurred in a number of other cases where there was otherwise no confusion between single and multiple pregnancies. See Appendix 3.

(D) In the case of NCDS respondents, the intention was only to update pregnancy history information since the last full NCDS sweep (NCDS5, 1991), and so the CAPI instrument was set up to enquire about all pregnancies since March 1991. However, for many cases a complete pregnancy history seems to have been entered, with dates prior to March 1991 included. See Appendix 4.

In trying to correct the above errors systematically, a number of factors introduced complications:

(i) In the case of babies wrongly bunched together in the same pregnancy, it is possible in most cases to re-arrange the 29 'baby' variables correctly, with no loss of information. But the same does not apply to most of the 10 'pregnancy' variables. For instance, where three babies from distinct pregnancies have been bunched as though they were a multiple birth, only one set of 'pregnancy' variables has been completed. So once the three babies are correctly separated into three distinct pregnancies, there is no way of telling which of these three the questions relate to (e.g. "did you smoke during this pregnancy?"). The two exceptions are **pregnum** ("How many babies did you carry in this pregnancy?) and **morepreg** ("Have there been other pregnancies before this one?"), which can be re-calculated logically by machine algorithm. A decision was therefore taken during this data cleaning exercise to copy the same pregnancy details into each of the now-separated pregnancy slots, but care should of course be taken when analysing these cases (see list at Appendix 1).

(ii) In the case of all pregnancies not resulting in a live birth, although the date of stillbirth/miscarriage/abortion is still required, only the month and year were asked. The 'day' variable (**preged, preged2-40**) remains system-missing, and consequently the composite date variable prege (prege2-40) also remains system-missing.

(iii) In the case of respondents who were still pregnant at the interview date, no outcome date could be entered, but one would obviously expect the data to be entered in the first (i.e. most recent) birth slot. However, there are cases where the ongoing pregnancy has been entered in birth slots other than the first, and indeed some where the ongoing pregnancy is featured as an apparent multiple birth with another completed pregnancy.   See Appendix 5.

(iv) A number of outcome dates contain the missing values 99 or 9999 in one or more of the 'day', 'month' or 'year' fields (**preged, pregem, pregey**, etc). In these cases, the composite 8-digit date variable prege (or **prege2-prege40**) remains system-missing. In most cases it seems to indicate the respondent (often a single or separated male) could not remember or did not know the full birth date, but could remember the year and possibly the month (although there are odd cases such as NCDS 385037C where the respondent could apparently remember the month of birth, but not the day or year).   If at least the year has been entered, we can in most cases sort out whether the pregnancy outcome has been put in the right order. But of course if the year is entered as '9999', the best one can do is a manual scrutiny of any other dates to hazard a guess as to whether it is likely to be in the right order. See Appendix 6 for the results of this manual scrutiny.

(v) Some successive outcome dates seem questionable. There are many cases where two dates are more than 3 days apart, but less than nine months (see Appendix 7). In the case of miscarriages or abortions this is understandable, although some miscarriages from apparently different pregnancies are entered as being in successive months. One or two live births are separated in time by over a month, but less than eight months, which seems to indicate a data entry error. A surprisingly common feature is a miscarriage followed a few months later by a live birth, which seems to indicate twins where one miscarried, and the other went to full term. But where there is a 7- or 8-month gap it is sometimes hard to tell if perhaps the two outcomes were from different pregnancies.   All cases in Appendix 7 were subjected to detailed scrutiny, looking sometimes at whether the live birth was logged as premature, or the

birth weight low, to decide whether to treat it as a separate or multiple pregnancy. The results are logged in the table attached to Appendix 7.

(v) A small number of outcome dates (all in BCS70) look as though they may have been entered erroneously, i.e the year was before 1970 or when the CM was extraordinarily young.   See Appendix 8.

(vi) A small number of respondents answered 'Yes' to the question 'Have you ever been, or got anybody pregnant?", but gave no information about any dates or indeed whether the pregnancy went to full term. In these cases, there is no basis for attempting any re-ordering. See Appendix 9.

Algorithm for correcting the errors

Step 1

Create a vector **BIRDAT,** containing 40 variables, corresponding to each of the forty birth slots, using the SPSS 'YRMODA' function, which produces a value equal to the number of days elapsed from a fixed time-point to the date in question. This allows all outcome dates to be compared chronologically, except for cases mentioned in paras (ii)-(vi) above, where there is an incomplete, missing or dubious date.

Step 2

In the case of incomplete dates where at least a valid year has been entered, assign a value for **BIRDATn** by imputing '15' for a missing day, and '6' for a missing month. Although this might result in two successive outcome dates appearing to be less than nine months apart, it should, to all intents and purposes, be impossible for them to get into the wrong order as a result of the imputation (for another outcome with a specified date to happen in the same year, the other event would have to be in the first three months or the last three months of the year, except in the rare case of a spontaneous miscarriage of one child from a multiple pregnancy). It should be noted that this imputation is only a temporary device for ordering the birth slot data correctly. The values of the actual outcome date variables (**prege-40, preged-40, pregem-40, pregey-40**) are not altered.

Step 3

In the case of ongoing pregnancies, assign a value for **BIRDATn** by imputing the date 30th September 2000. As all interviews had been completed by this date, it would be impossible for there to be a subsequent outcome, ensuring the ongoing pregnancy would be placed as 'most recent' in the corrected ordering.   There were some ongoing pregnancies where two birthslots had been filled in, but one can assume these were cases where it was already known the mother was expecting twins (see Appendix 5). The imputation in these cases would result in two identical outcome dates of 30/9/2000, which would correctly enable them to be flagged up as twins under Step 7 below.

Step 4

Identify cases where any outcome date has the year entered as '9999' (birth year not known/not answered). In these cases, there is no sound basis for imputation, and it is impossible to determine by machine algorithm what order to place that baby in. These cohort members are therefore flagged to be examined on a case-by-case basis. See Appendix 6.

Cohort members with no outcome data are also flagged to be by-passed (see (vi) above, and Appendix 9).

Step 5

Excepting cases eliminated in step 4, flag up all cases where outcomes more than nine months apart have been bunched in adjacent birth slots as though they were part of one pregnancy, and cases where outcomes have been entered in the wrong chronological order (i.e. not putting the most recent pregnancy first).

Step 6

Set up a variable **TOTBABY** to log which birth slots contain any data at all (i.e. any outcome date, even if entered as '9999', or any outcome code (e.g. livebirth, miscarriage, even if entered as '9'), so that we have a check, after the re-ordering process, to ensure no baby was missed out.

Step 7

Implement an iterative procedure to compare all forty possible outcome dates (**BIRDAT1-40**), producing up to eight '**KEY'** variables indicating which of the forty slots was the most recent date found (**KEY1**), the next most recent (**KEY2**), and so on. For instance, if the third most recent event was found in slot 7 of the forty, **KEY3** would be assigned the value 7. We know from a cursory analysis that only one 1999/2000 cohort member ever carried more than eight babies (BCS70 15610009), and a manual analysis confirmed that case had all its pregnancy history data correctly ordered, so it was bypassed by this algorithm.

Step 8

Using this information, compare each successfully-ordered birth date with the adjacent one, to flag up all cases with true multiple births (i.e. births on same day or at least within 3 days), and all cases with a dubious multiple birth (i.e. more than 3 days, but less than nine months apart). As with Step 4, the intention is to look at these manually on a case-by-case basis.

Step 9

Having excluded from the algorithm all cases flagged in Steps 4 and 7, prepare to re-order all cases flagged in step 5 as having been wrongly bunched together in the same pregnancy, or wrongly ordered chronologically.

The re-ordering not only has to move each outcome date to a new slot, but also every one of the 29 variables associated with that particular baby (birth weight, outcome, etc), as well as the 10 variables associated with that particular pregnancy (number of babies in pregnancy, whether smoked, whether used contraception, etc). In particular, the corresponding **pregnum** is set to a value of 1, indicating a single, not multiple birth. As documented above, once a bogus 'multiple' pregnancy is separated out into two or more single-baby pregnancies it is unfortunately not possible to tell which of these pregnancies these latter variables (smoking/contraception, etc) relate to. A decision was made to copy the same pregnancy data into each of the new pregnancy slots; users should therefore be careful in the way these data are interpreted. The cases affected are all those listed in **Appendix 1.**

A dummy set of variables in vector format of length 40 is set up so that all the 29 'baby' variables can be copied and indexed by the slot in which they erroneously appeared.

Having been first copied to the dummy vectors, the original 29 variables are set to the system-missing value in each of the 40 slots.

Similarly, dummy vectors of length 8 are set up for the ten 'pregnancy' variables; the data are copied from the original variables, which are then set to system-missing.

Step 10

Complete the re-ordering process. Copy the data for the most recent baby into the 'slot 1' versions of all 29 variables, applying the '**KEY1'** variable from Step 6 as the index for the vector; then copy the data for the next most recent into the 'slot 6' versions by applying '**KEY2'**, etc., then 'slot 11' by applying **KEY3'**, etc. until all babies carried have been accounted for.

We know that all such pregnancies are single outcomes, as the multiples and dubious cases have been excluded from the algorithm. So the pregnancy variables in slots 2-5, 6-10, 12-15, … etc are bound to remain system-missing. It should be noted that for these cases the re-ordering addresses both problems outlined in (A)-(C) above (i.e. the 'bunching' as well as the 'out-of-order' problem).

Similarly, copy the data for eight of the ten 'pregnancy' variables into the correct positions. Of the other two, **pregnum** is set to 1 (as multiple pregnancies have been excluded) and **morepreg** is set to 1 or 2 depending on a machine-check as to whether other earlier pregnancies exist.

Step 11

Create new variable **NEWTOTBY** to count how many babies are represented in the newly re-arranged data, and check this is the same in total as the number of babies before re-arrangement (see Step 6).

Step 12

Having checked that no outcome data was lost, delete all NCDS outcomes earlier than March 1991, unless cohort member was not contacted at the 1991 survey (see Appendix 4).

Step 13

Do manual scrutiny of cases with any outcome years entered as '9999' (see step 4), true multiple births, dubious multiple births, and those still pregnant but where that pregnancy is not placed in most recent birthslot (see step 8).

Step 14

Implement syntax for case-by-case re-ordering in the light of the manual scrutiny.

Step 15

Save SPSS system file, disposing of all the working variables created during this algorithm, and saving the altered versions of all the original variables.

Appendix 1

NCDS Cases where birth outcomes were wrongly bunched with other outcomes more than nine months apart, as though they were a multiple pregnancy.

044014L	045002J	045011K	050059U	052008M	053003F
056052M	085010J	087029S	091017J	092206S	093039E
093052W	094028E	099010M	099050Z	100009C	110213J
110244V	120017Q	120080X	120141R	180007A	183016T
200075Z	218005B	235011T	280005D	280064W	282113T
285061S	286044X	380041R	380090E	385005P	400075Q
436010M	440006V	481001R	500125L	500376N	504034D
506014J	510097Q	512002L	512012P	513033C	513073R
513143L	514012Z	517045H	520056F	528014H	528021D
550188U	560008V	610087U	630006D	650212V	710034D
720027P	750043F	823501Q	850032J	860001C	880017H
882021J	882051T	933019C	937020J	950123U	950168T
982053E	984024J	985067J	986100J	986250D	986431J
X10020X	X25101F	X32005A	X41020X	X60006M	X66030R
X71009E	X77014E	X77016K	X77021B	X78086N	X82347P
Y00087K	Y01189Y	Y20134D	Y20154L	Y30150J	Y30168D
Y30169F					

97 cases in total.

This does not include cases where, although less than nine months apart, outcomes have been bunched together which are almost certainly from two separate conceptions (e.g. two miscarriages seven months apart – see Appendix 7).

Appendix 1 (cont.)

BCS70 cases where birth outcomes were wrongly bunched with other outcomes more than nine months apart, as though they were a multiple pregnancy.

00279065	00312058	00387068	00530014	00607090	00785026
01065011	01236012	01251089	01254014	01592000	01618019
01763001	02294098	02306039	02441045	02454047	02609097
02765030	02880084	02914028	02934081	02939057	03085020
03191073	03264047	03366075	03371075	03575030	04046069
04239073	04394031	04452053	05014068	05036071	05289032
05399086	05408250	05521004	05533031	05542032	05585063
05794018	06031073	06072053	06123024	06183082	06266032
06305077	06466087	06501030	06570089	06641012	06666041
06797022	06853094	06896024	06959022	07052079	07244007
07416084	07469091	07494043	08078061	08239086	08263063
08283015	08302007	08409011	08445015	08787002	08818021
08962028	08995083	09054086	09145062	09204059	09261091
09262066	09310011	09318013	09490099	09495075	09516017
09573049	09591051	09716072	09803047	09840026	09891083
10047010	10215086	10455045	10477048	10527044	10582025
10657050	10732099	10937029	11032036	11047089	11212038
11487002	11636000	11706049	12091097	12155070	12434074
12609076	12958013	13538030	13561032	13705030	13762136
13945090	14228000	14293058	14381008	14538008	14671064
14687092	14858093	14869044	14978022	15084081	15093082
15094057	15440084	15455036	15511007	15527035	15567040
15571065	15737066	15810064	16027027	16102076	16152057
16224056	16354062	16568094	16630040	16717092	16822069
17386019	18013006	18046061	20043000	20091000	20212000
20305000	21667000				

152 cases in total.

This does not include cases where, although less than nine months apart, outcomes have been bunched together which are almost certainly from two separate conceptions (e.g. two miscarriages seven months apart – see Appendix 7).

Appendix 2

NCDS Cases where birth outcomes were wrongly bunched as though they were a multiple pregnancy, and also out of chronological order.

044014L	045002J	045011K	052008M	053003F	056052M
087029S	091017J	092206S	093052W	094028E	099010M
099050Z	100009C	110213J	110244V	120017Q	120141R
180007A	183016T	200075Z	218005B	282113T	285061S
286044X	380041R	380090E	385005P	436010M	440006V
481001R	500125L	500376N	504034D	512002L	513033C
513073R	513143L	514012Z	517045H	520056F	528014H
528021D	550188U	560008V	650212V	720027P	823501Q
860001C	880017H	882021J	882051T	950168T	984024J
985067J	986100J	986431J	X10020X	X25101F	X41020X
X60006M	X66030R	X77014E	X77016K	X77021B	X78086N
X82347P	Y01189Y	Y20134D	Y30150J	Y30168D	Y30169F

72 cases in total.

BCS70 Cases where birth outcomes were wrongly bunched as though they were a multiple pregnancy, and also out of chronological order.

00279065	00312058	00530014	00607090	00785026	01236012
01618019	02294098	02306039	02441045	02454047	02609097
02765030	02880084	02914028	02934081	02939057	03085020
03191073	03264047	03366075	03371075	03575030	04046069
04239073	04394031	04452053	05014068	05533031	05542032
05585063	06031073	06123024	06183082	06266032	06305077
06466087	06570089	06797022	06959022	07244007	07469091
07494043	08078061	08239086	08283015	08302007	08409011
08445015	08818021	09054086	09261091	09310011	09318013
09490099	09495075	09573049	09591051	09716072	09840026
09891083	10047010	10215086	10477048	10657050	10732099
10937029	11032036	11047089	11212038	11487002	11706049
12155070	12434074	12609076	12958013	13538030	13705030
13945090	14671064	14687092	14858093	14869044	15084081
15093082	15094057	15440084	15455036	15511007	15527035
15571065	15737066	16102076	16152057	16224056	16354062
16568094	16630040	16717092	18013006	20043000	20212000

102 cases in total.

Appendix 3

NCDS Cases where birth outcomes were out of chronological order, although not wrongly bunched.

010004R	041034A	043013C	043058B	044001A	051003V
055013V	055086Z	081002N	083034N	084004J	084015P
084027W	090001M	091030Z	092023J	092024L	092179S
093055C	093087S	093259V	094029H	095023Z	110065V
110079H	110080R	110097K	110304M	120070U	120132Q
130007T	181038T	186004B	186081W	186125Q	188030P
216003M	223020B	235018J	239005V	280041J	280053R
282012M	282104S	286058K	288027J	289131J	300080Z
308053Q	310016W	310101M	310107Z	330055V	330073X
340061W	350110P	360027K	381111R	385025V	385110L
400027C	420022E	431038J	432006Z	435002H	500015C
500080Q	500124J	500234R	500375L	500418B	501009P
509254W	510127W	513053K	513117K	513118M	515085K
517057Q	518069C	520064E	522007C	523033K	524002C
526007Z	529004K	550026R	550534M	561006W	581050N
610028B	610046D	620075T	620087A	650132X	650172L
710063M	730049F	730050Q	730084J	750101T	783041N
821507S	824020C	835013T	845039U	882007Q	910031N
931002Y	950124W	950135B	950154F	950290Q	960031V
960072L	980004E	986059Q	986117B	986183R	987048Q
987100P	989011A	X32044M	X32084Z	X32159E	X33020B
X40028K	X40036J	X67026F	X78016Q	X78079R	X80013W
X82215V	X82287X	X82344H	X84009C	X87024Q	X87061W
Y00164A	Y00173B	Y00337F	Y01018U	Y01605L	Y20023U
Y20055J	Y20078W	Y21001P	Y30223K	Y33089Z	

149 cases in total.

BCS70 Cases where birth outcomes were out of chronological order, although not wrongly bunched.

00051056	00103002	00124030	00346088	00463092	00508088
00520038	00620015	00700040	00809095	00813019	00879028
00912021	00914072	01005054	01153062	01266041	01302060
01445068	01576049	01620094	01674076	01695003	01736099
01840000	01859054	01870029	01945054	01949055	01993085
02051012	02126037	02207037	02252042	02478000	02753003
02830002	02896011	02904052	02930080	03121039	03134041
03178047	03279000	03325095	03349048	03379077	03403069
03418021	03520221	03556053	03579031	03600098	03621025
03691059	03735080	03745056	03749057	03783010	03786036
03795037	03884063	03894039	03926033	03956062	03982090

03997042	04139096	04177050	04361077	04474056	04486083
04498009	05098079	05137023	05227024	05236025	05249027
05313024	05325051	05369057	05425028	05486061	05506028
05531081	05548084	05593089	05661087	05671063	05721207
05749013	05832088	05895071	05910062	05927065	06028023
06057077	06145027	06173005	06195008	06196084	06198034
06226027	06299087	06313002	06393012	06394088	06458061
06467062	06471087	06481063	06507082	06548062	06607059
06612059	06619086	06691094	06791071	06880097	06903090
06908066	06982024	06992000	06998052	07019000	07061080
07149006	07191086	07194011	07211053	07326083	07338009
07339085	07359037	07391040	08006077	08033080	08039031
08052057	08163086	08215032	08267064	08281065	08388069
08460092	08477095	08632068	08657097	08700066	08716094
08965054	08993032	09002054	09027083	09047035	09073063
09362043	09393047	09541070	09567074	09598078	09667125
09725073	09747076	09758027	09797057	09799007	09982059
10141036	10299097	10308012	10312037	10463071	10667026
10796057	10822000	10868056	10894084	10921001	10927053
10964032	11059015	11073016	11086018	11102085	11281097
11314066	11391050	11414043	11515096	11539049	11541023
11556076	11637076	11677081	11871084	11937007	12074070
12124066	12150094	12156045	12263073	12394054	12464002
12522024	12596059	12733030	13045095	13054096	13062021
13074048	13189078	13346001	13436002	13468082	13560057
13740059	13741034	13766063	13771063	13788066	13799017
13930037	14046048	14137024	14185055	14249028	14331027
14371032	14414078	14479011	14644061	14699018	14812036
14924040	14989074	15095032	15136027	15347033	15351058
15372086	15675043	15703036	15935070	15941045	16076033
16093060	16100025	16170059	16281088	16293014	16336060
16460014	16481042	16563017	16776074	16809043	16835071
16922046	16992080	17062034	17139009	17163087	17170037
17267065	17270014	17292017	17327037	17371067	17373017
17456068	17472019	17624043	17687026	18132061	20097000
20322000	20345000	20499000	20503000	20531000	21644000
21740000					

283 cases in total.

In addition, the following 12 cases had all their dates in chronological order, with no birth outcomes bunched together (and no 'true' multiple births) but nevertheless had no data in the first birth slot (i.e. the data were in slots 6, 11, 16, ... etc); so all data had to be pushed back, slot 6 being transferred to slot 1, slot 11 to slot 6, ... etc.

01001053	02182094	03257097	03310042	05984097	09763027
10788031	14101044	14574012	15013072	17015079	20432000

(Case 08052057 also had no data in the first birth slot, but is listed separately in Appx. 6 and Appx.7 because of other complications)

Appendix 4

NCDS Cases where birth outcomes prior to March 1991 were recorded, but where cohort member was interviewed at 1991 survey.

010101P	010111S	010184W	010185Y	010223D	043003Z
043025L	044001A	044024P	045002J	045011K	045012M
046014W	046028J	050069X	051003V	052008M	053003F
056052M	075006M	081012R	081019F	084015P	087015E
088005H	088012D	090028J	092053T	092179S	092206S
092293Q	092405Y	093032Q	093052W	093055C	093063B
093234C	093259V	093272M	093273P	094028E	094029H
094040U	095045L	097011C	099010M	100009C	100031V
100055L	110030Z	110065V	110079H	110097K	110135Q
110193E	110290C	110304M	120017Q	120048B	120141R
130008V	180001N	182007M	183005N	183016T	184047L
185014Z	186238D	187019W	188108A	200006D	200086E
226004V	233004L	233008U	235003U	280017M	280034M
280068E	282029E	282104S	282109C	284011V	285020B
285032K	285052R	285061S	286042T	287069V	287150B
287210T	288009F	289031D	289063T	289123K	289131J
289141M	289180X	300093K	300133U	325016E	330027Q
330073X	350139Q	380031N	380084L	381111R	381127H
382004V	383051L	385004M	385005P	385020K	385053A
385173M	405014V	405042A	405050Z	420025M	420038W
420052Q	421001B	421030K	422073K	422081J	423006Y
423045K	424029S	431038J	431040U	432025D	434020D
434035T	435002H	436011P	450075X	460028U	460039Z
460043Q	465048C	465051R	481044L	482031F	500032C
500057V	500080Q	500113C	500125L	500353Z	500423U
504020S	509205H	510024L	510097Q	510153X	511063B
512012P	512044C	512095W	513018H	513083U	514121E
515034R	516085Q	517012R	517016Z	517213B	518025F
518029Q	518037P	518069C	518134M	520049K	520086R
523015H	524002C	524004H	524005K	525013P	527001S
528020B	528021D	528083C	529009V	550033N	550158K
550245D	550289A	550379B	550388C	560022P	560031Q
565027B	565036C	565063F	565108B	580011W	581068J
583049Q	591036A	591052Y	593013Y	610028B	610070A
620005V	620107D	630002V	630006D	650025W	650116Z
650122U	650164M	650172L	650191Q	683009K	683019N
684035R	684054V	710068X	720019Q	720098P	730015N
730130N	750075V	781005X	782013B	782054S	782073W
782114J	805004X	815048A	821519Z	823501Q	823502S
825028A	825049K	825115V	840012V	850012B	865021M
881014F	881015J	882007Q	882019X	882021J	882037Z
882061W	936012D	950114T	950123U	950155J	950168T
950266T	960003Q	960022U	960046K	960073N	983033D
985039C	986027A	986039J	986183R	986284X	986374Y

986411B	986430F	987013U	987097D	989041L	X25029A
X25050R	X25056D	X25057F	X32005A	X32152Q	X34011F
X37008K	X38005J	X66030R	X70045D	X70066N	X76010R
X77014E	X77029U	X78019W	X78043T	X79021N	X80020T
X80034E	X82062W	X82115R	X82137B	X82250X	X82256L
X82276S	X82282M	X85003V	X86018Q	X88003M	Y00130H
Y00274J	Y20023U	Y20055J	Y20134D	Y20187B	Y21001P
Y21037M	Y30072Q	Y30168D			

309 cases in total.

The following had pre-March 1991 pregnancy history data recorded, but they were
not interviewed at the NCDS5 survey, and so it was correct in those cases to leave
that information in the data:

050058S	089007S	092107Q	093100E	099053F	188030P
200075Z	213004X	218005B	283003R	288034E	310021P
330009N	330081W	350101N	350140Y	434034R	465013H
500167C	500245W	500257D	500410K	500517D	514012Z
516069S	523054T	525015T	527060K	528006J	591050U
710063M	720095H	740122V	750066U	783041N	815018R
822524X	825018X	881023H	882051T	910031N	937039F
986117B	986380T	X32044M	X32159E	X78086N	X80088E
X82195S	X82519S	X82639C	X90012A	Y01099X	Y01565A
Y20024W	Y20046H	Y20159W	Y20274W	Y30043H	Y30263X

60 cases in total.

Appendix 5

NCDS cases where there was an ongoing pregnancy, but data out of order (i.e. not entered in first slot), or pregnancy was wrongly bunched with another outcome as an apparent multiple.

055086Z	081002N	436010M	501009P	620087A	910031N
X33020B	X40036J				

[Cases 781048S and 986275W had two 'still pregnant' entries, presumably indicating it was already known to be twins. No alteration necessary].

BCS70 cases where there was an ongoing pregnancy, but data out of order (i.e. not entered in first slot), or pregnancy was wrongly bunched with another outcome as an apparent multiple.

The following 52 cases had the ongoing pregnancy not entered in the first birth-slot, and so had to be re-ordered:

00312058	00463092	00785026	01266041	01870029	02126037
02609097	02765030	03191073	03403069	03575030	03786036
03926033	04139096	04452053	04486083	05425028	05749013
05895071	06691094	07020000	08281065	08388069	09027083
09054086	09393047	09667125	10312037	10463071	10657050
10868056	10921001	11059015	11414043	11937007	12124066
13074048	13538030	13705030	13741034	14249028	14644061
14924040	14989074	15094057	16093060	16630040	16809043
16860023	17373017	17456068	20097000		

The following 7 cases had the data for the ongoing pregnancy correctly entered in the first slot, but other birth outcomes were wrongly bunched in the same pregnancy (i.e. in slots 2, 3, ... etc):

01592000	06896024	09516017	13762136	14228000	14381008
16027027					

The following five cases had two 'still pregnant' entries, presumably indicating it was already known to be twins, so they were not altered:

00060057	05435004	07406007	10531069	12278025

One ongoing pregnancy was correctly entered in the first slot, with a miscarriage in slot 2. However, the miscarriage was sufficiently recent for it to be plausible that it was from the same pregnancy, so this was not altered:

10723098

Appendix 6.

NCDS cases where one or more outcomes had year entered as '9999', so that chronological order could not be easily ascertained.

We looked at dates of other outcomes: where there were at least two other proper dates entered, we could guess whether the whole lot were put in the wrong order or were wrongly bunched. Sometimes it was possible to get information from the type of outcome. Summary of types:-

(a) No bunching, order seems correct based on at least two other dates entered (or where one other date/type of outcome makes it clear, e.g. 'still pregnant'). Leave as is.

(b) No bunching, and no way of telling if order correct. Leave as is.

(c) Bunched in same pregnancy as another, but no way of telling if bunching is correct (otherwise chronological order seen to be correct from at least two other dates). Leave as is.

(d) Bunched in same pregnancy as another, but no way of telling if bunching is correct (also no way of telling if chronological order is correct). Leave as is.

(e) Only one baby ever carried, so no question about bunching or order. Leave as is.

(f) Miscellaneous complex cases where no choice but to leave as is. See comment.

(g) No bunching, but order incorrect on at least two other dates (or where one other date is so late as to leave no room). Reverse order.

(h) Bunching, with 2 or more others which should clearly not be bunched because >9 months apart (but correct chronol. order). Spread to slots 1,6,11, etc.

(j) Bunching, with 2 or more others which should clearly not be bunched because >9 months apart (and chronol. order wrong as well). Reverse order and spread to slots 1,6,11, etc.

(k) Bunching, with one other specific date, both live births. Assume order correct but should be two different pregnancies. Spread to slots 1,6.

(l) Miscellaneous complex cases needing alteration.

nserial	type	changed?	Comments
041034A	j	changed	
081043C	b		
082020V	e		
084017T	a		
088006K	a		
093273P	b	changed	Other pre-March 1991 birth event deleted.

094060A	e		One baby only. Outcome coded as '8' (don't know)
098030N	b		
099078Y	b		
130007T	g	changed	2 '9999' dates bunched, (assume correctly), but order of pregnancies reversed
180004U	b		
213007D	c		
283003R	d	changed	one live birth (pre-1991). 6 other '9999' dates have outcome='9' and no other data to go on. As live birth is pre-1991, delete that and leave others.
287204Y	a		
289081V	b		
340059L	a		
340069P	e		
360020U	a		
383021A	f		One stillbirth, two double-stillbirths, all dates '9999'. 2 other live births
383032F	e		
385008V	e		one other pregnancy is still ongoing, so order must be correct.
385037C	e		
385199F	a		
400111R	b		
422073K	a		
424029S	a		
503039K	e		
509003U	b		
511159R	k	changed	
517104W	b		
550108T	b		
550319H	e		
550374R	b		
561006W	g	changed	
565063F	a		
583060B	e		
620175X	a		
650025W	a		
720109R	f		miscarriage at date '9999', bunched with live birth. Leave as is.
730133U	a		
781031Y	b		
782013B	h	changed	
782014D	b		
824010Z	e		
830003N	e		
986004N	e		
986049M	e		
987021T	b		
X25050R	f	changed	Two dates '9999' (outcome='9') bunched with pre-March 91 live birth. Delete live birth, leave others.
X38005J	b	changed	Birthslot 1 was pre-March 91, slots 6 & 11 '9999'. Delete slot 1, move 6 & 11 to 1 and 6 respec.
X71001N	f		Two miscarriages, both dates '9999' bunched together. Leave as is.
Y00126S	b		
Y01097T	b		
Y20141A	a		

Y20246R e
Y20303A d
Y21037M a   changed   One pre-March 91 birth event deleted.
Y33089Z g   changed   Order of 3 other babies chaotic, so they were ordered, but '9999' outcome left.

BCS70 cases where one or more outcomes had year entered as '9999', so that chronological order could not be easily ascertained.

For key to types of situation, see NCDS table above.

bserial	type	Changed	Comments
00274089	b		
00279065	l	Changed	Two other outcomes bunched, wrong order. Separate, and reverse order.
00655020	e		
00953001	e		
00972079	e		
01441067	b		
01682001	b		
01914050	b		
01931077	a		
02082016	a		
02160091	e		
02341216	a		
02532021	e		
02624073	b		
02718075	a		
02847005	b		
03212015	b		
03262097	e		
03703000	e		
03716002	e		
05086052	e		
05186029	a		
05457007	b		
05486061	l	Changed	Two of the other five outcomes out of order, so swap them, leave rest.
05780041	b		
05794018	l	Changed	Two other outcomes bunched. Separate, and push everything along 1 slot.
05903011	b		
06025098	d		
06150027	e		
06219077	b		
06363084	a		
06406029	e		
06605008	e		
06814064	a		
06951020	e		
07020000	l	Changed	9999 event (abortion) bunched with ongoing pregnancy. Assume diff.preg.

07129054 e
07231005 b
07332058 c
07363062 a
07462064 a
07465090 e
08052057 l   changed No data in 'most recent' pregnancy. Move other 3 along one slot, reversing slots 7, 6.
08062033 b
08221007 b
08486096 a
08531015 l   changed Slots 1 and 11 both '9999', but slot 6 empty. Move 11 to 6.
08667073 b
08911072 a
08958003 c
08965054 a
08983056 b
08989007 a
09047035 g   changed Reverse order of the three birth events
09099067 e
09115033 a
09194068 e
09362043 l   changed Two of the other four outcomes out of order, so swap them, leave rest.
09541070 l   changed Two of the other four outcomes out of order, so swap them, leave rest.
09634097 c
09774079 a
09915051 a
09925027 a
09965032 c
09992035 a
10082039 e
10385097 a
10397023 d
10627021 e
10718098 e
11080067 b
11452098 a
11666029 e
11801050 b
11872059 d
12096073 a
12347099 b
12420097 e
12473003 b
13677037 e
13799017 l   changed Two of the three dated outcomes out of order, so swap them, leave rest.
13917011 a
13926012 e
13956290 a
14149051 a
14290032 a
14528032 b

14718010	a
15289011	a
15478014	e
15510032	e
15568015	e
15665067	e
15776096	e
15830016	d
15949047	b
16000048	b
16032027	b
16164084	a
16355037	a
16692048	b
17070060	a
17232060	a
17436015	a
17466044	e
20203000	a

Appendix 7.

NCDS cases where two birth outcomes are separated by more than three days, but less than 9 months.

010168Y	041001K	044001A	046040X	048074B	050068V
055086Z	083034N	083055W	084004J	092098S	092144W
093021K	094029H	095006Z	095023Z	100020Q	100023W
110304M	120128Z	120132Q	181038T	184008Z	186055V
186152T	187001A	188030P	215003F	223020B	230016B
233005N	233008U	235018J	280041J	280053R	285020B
286019Y	288016C	288027J	310017Y	310069U	310107Z
330044Q	350146M	382004V	385025V	385085Q	400004Q
400024W	410007C	421009U	421017T	424031C	431017Z
465003D	500053M	500353Z	500375L	503020M	509206K
509228V	510127W	510161W	513053K	513136P	514030B
515008Q	515085K	517006W	517033Z	524002C	528006J
529009V	550118W	550227B	565034Y	581068J	583074P
591050U	620037K	630002V	630027N	650034X	650136F
681009Y	684001X	730049F	730084J	730130N	730135Y
780019D	823509H	825073F	850027R	855004E	880048U
880088H	937058L	950123U	950154F	950172J	950288D
950290Q	960029K	960072L	980004E	985067J	986039J
986140W	986192S	986205Y	986310V	986313B	987040X
987100P	989011A	X25089V	X25101F	X32084Z	X38006L
X67014Y	X78037Y	X82164E	X82287X	X82344H	Y00070R
Y00174D	Y00247E	Y00293N	Y20154L	Y20159W	Y21019K
Y31043N					

133 cases in total.

On manual scrutiny, 62 of these were plausible, e.g. two miscarriages seven months apart, listed in separate pregnancies.

However, the following 71 cases were problematic in some way or other. The middle column contains a 'C' if the case record was changed by this cleaning exercise.

nserial	Ch?	Comments
041001K		Miscarriage 1992 (no month), abortion July 92, in same pregnancy. Assume OK.
044001A	C	Double miscarriage, with single miscarriage & three other livebirths in wrong order
055086Z	C	Live birth with miscarriage 6 months later (i.e. OK), also still pregnant, but order wrong
083034N	C	Two live births 2 weeks apart, different pregnancies, & order wrong. Move to same pregnancy
084004J	C	Chronological order completely confused
094029H	C	Two live births a month apart, and wrong order, but both pre-1991 anyway, so delete.
095006Z	C	Two live births a fortnight apart, placed in separate pregnancies. Move to same preg.
095023Z	C	Stillbirth eight days after live birth. OK, but order wrong
100023W		Miscarriage, then live birth 6 months later, same pregnancy. Assume OK.
110304M	C	Two miscarriages 4 months apart, sep.pregnancies. Assume OK, but order is wrong
120132Q	C	Order wrong, involving stillbirth and live birth and one separate pregnancy.

181038T	C	Miscarriage, then live birth 7 months later, same pregnancy. Assume OK, but order wrong
184008Z		Two miscarriages in consecutive months, down as separate pregnancies. Leave as is*
188030P	C	Miscarriage, then twins seven months later, wrong order. Assume same pregnancy
223020B	C	Chronological order completely confused
230016B		Miscarriage, then live birth 6 months later, same preg. Assume OK.
233005N		Miscarriage, then live birth 5 months later, same preg. Assume OK.
233008U	C	Miscarriage, then live birth 6 months later, same preg. Pre-March 91, so delete.
235018J	C	Miscarriage 3 months after live birth, sep.preg. Assume OK, but order wrong.
280041J	C	Abortion 4 months after miscarriage, same preg. Assume OK, but order wrong.
280053R	C	Miscarriage, then live birth 5 months later, separate pregnancy. Move to same preg.
288016C		Two live births 4 months apart both weighing 9lbs 8 oz, entered as sep.pregs. ?????????
288027J	C	Three miscarriages in 7 months, all different pregnancies. Assume OK, but order wrong
310107Z	C	Miscarriage, then live birth 8 months later, separate pregnancy. Assume OK, but order wrong
382004V		Miscarriage, then live birth 3 months later, same pregnancy. Assume OK.
385025V	C	Miscarriage eight months after live birth, diff preg, so OK but order wrong.
410007C		Two live births 7.5 months apart. But pregf=1 for the second, i.e. premature, so leave as is.
421017T		Two live births 6 months apart. pregf=3 for the second, i.e. 'on time'. Puzzling, but leave.
431017Z		Miscarriage, then live birth 5 months later, same pregnancy. Assume OK.
500353Z	C	Miscarriage, then live birth 2 months later, but pre-March 91 anyway, so delete.
500375L	C	Two miscarriages 4 months apart, different pregnancies. Assume OK, but order wrong
509228V	C	Miscarriage, then live birth 3 months later, diff.pregnancy. Move to same pregnancy.
510127W	C	3 miscarriages, 1 abortion in 17 months, all diff. pregnancies. Assume OK, but order wrong.
513053K	C	Four miscarriages 21 months, all different pregnancies. Assume OK, but order wrong
515008Q		Miscarriage, then live birth 8 months later, same pregnancy. Assume OK.
515085K	C	Stillbirth five days after live birth. Order wrong.
517006W		Misc. then live birth 6 months later, diff.preg. pregf=1 for the second, i.e. premature, so leave.
524002C	C	Two live births 6.5 months apart, order wrong. pregf=3 for the second, i.e. 'on time'. Puzzling.
528006J	C	Two miscarriages in same year. OK, but delete earlier pre-1991 baby.
529009V	C	Stillbirth 7 days after live birth, same pregnancy. Leave as is, but delete earlier pre-1991 baby.
565034Y		Misc.then live birth 5 months later, diff.preg. 2nd was v.premature, died after ten days, so leave.
591050U	C	Misc. then live birth 5 months later, diff.preg. Second 'on time', so move to same pregnancy.
620037K	C	Misc. then live birth 2 months later, diff.preg. Second 'on time', so move to same pregnancy.
650034X		Miscarriage, then live birth 6 months later, same pregnancy. Assume OK.
684001X		Miscarriage, then live birth 7 months later, same pregnancy. Assume OK.
730049F	C	Miscarriage 2 months after live birth, diff.preg. Assume OK, but order wrong.
823509H		Miscarriage, then live birth 8 months later, same pregnancy. Assume OK.
825073F		Miscarriage, then live birth 8 months later, same pregnancy. Assume OK.
855004E		Two miscarriages in consecutive months, sep.pregnancies. Puzzling, but leave as is*
950123U	C	3 births & miscarriage bunched together, incl. 2 live births 4 months apart(???). Separate out.
950154F	C	Chronological order completely confused
950288D	C	Two live births 2 weeks apart, entered as sep.pregs. Move to same pregnancy.
980004E	C	Miscarriage, then twins seven months later. Assume same pregnancy, but change order
960029K	C	Miscarriage, then live birth a month later, down as diff.pregnancy. Move to same pregnancy.
960072L	C	Two live births 3 months apart, same pregnancy(???). Wrong order. Reverse order
985067J	C	Two miscarriages 8 months apart, same preg. Move to diff.preg. Re-order 4 bunched others
986039J	C	Two live births 5 weeks apart, sep. pregnancies. But all pre-1991 anyway, so delete.
986140W		Two miscarriages in consecutive months, sep.pregnancies. Puzzling, but leave as is*
986205Y	C	Twins bunched with misc.13 months earlier, but stillbirth 6 days later not entered in same preg.
987040X		Miscarriage, then live birth 7 months later. Pregf=1, i.e. 'early', so assume OK.
987100P	C	Miscarriage, then live birth 2 months later, diff.preg. Move to same preg, re-order.
989011A	C	Miscarriage, then live birth 8 months later. Pregf=3 ('on time'): assume same preg, re-order.

X25101F C Miscarriage, then live birth 8 months later. Pregf=3 ('on time'): assume same preg, re-order.
X32084Z C Chronological order confused
X82287X C Live birth, then miscarriage 2 months later, different preg. OK, but order needs reversing.
X82344H C Two live births 3.5 weeks apart, diff.pregs. Move to same preg.
Y00174D C Double miscarriage (same date) entered as though two separate pregnancies
Y00247E   Miscarriage, then live birth 8 months later, different preg.   Pregf=1 ('early'): Assume OK.
Y00293N   Miscarriage, then twins six months later: assume same pregnancy, so leave as is.
Y20154L C Two live births 5 months apart, diff.pregs. (??????). 2 other pregs wrongly bunched.
Y31043N   Miscarriage, then live birth 7 months later, same preg.   Pregf=1 ('on time'): Assume OK.

* Although it would obviously be impossible to have a miscarriage from a new conception as soon as
one month after a previous pregnancy outcome, the fact that we never know the 'day' within the month
for miscarriages means that there could possibly be as much as eight weeks between the two events,
making it just about plausible.

Appendix 7 (cont.)

BCS70 cases where two birth outcomes are separated by more than three days, but less than 9 months.

00046056	00117080	00250035	00305007	00376016	00407035
00467093	00485095	00508088	00593098	00665097	00680073
00700040	00718018	00757048	00812044	00820070	00831021
00912021	00947026	00951051	00999058	01005054	01069012
01166064	01266041	01332089	01424040	01445068	01453094
01516092	01548071	01628096	01653048	01695003	01711070
01739024	01761051	01840000	01859054	01927052	01949055
01999036	02061089	02100033	02257018	02281096	02346044
02353095	02388000	02505018	02621047	02688031	02692056
02713099	02726000	02738027	02746053	02977011	03029064
03121039	03279000	03299052	03349048	03352098	03379077
03430072	03594007	03671006	03735080	03738005	03796187
03997042	04046069	04097025	04183025	04222070	04277027
04294054	04347076	04350025	04494008	04498009	04502049
04516026	04538029	05033045	05035096	05073050	05097003
05167052	05170001	05239051	05254027	05336002	05435004
05486061	05531081	05548084	05575087	05580087	05637009
05661087	05750012	05796069	05806060	05819062	05826012
05840013	05871017	05991047	06077029	06092005	06128000
06145027	06173005	06177006	06180130	06196084	06226027
06313002	06394088	06481063	06521083	06580065	06616060
06643063	06680042	06707036	06731013	06757017	06872071
06880097	06882047	06922067	07051003	07090033	07150005
07246058	07318057	07342034	07405032	07508284	08039031
08042081	08052057	08064084	08121030	08129032	08162010
08177063	08239086	08243010	08349039	08460092	08498197
08498271	08534041	08542067	08613091	08667073	08716094
08750047	08887080	08930049	09002054	09087040	09316063
09324089	09362043	09530018	09541070	09615019	09681052
09708046	09898009	09909076	09965032	10070012	10231037
10295096	10355068	10406039	10455045	10517068	10526069
10531069	10583000	10633097	10643073	10645023	10675052
10678078	10723098	10822000	10838027	10928028	11002007
11019010	11102085	11162042	11254094	11257019	11519097
11539049	11541023	11553050	11572027	11573002	11620073
11680030	11779008	11825003	11841055	11871084	11951008
11969087	12157020	12188024	12195075	12263073	12397080
12428099	12430073	12617001	12619052	12786037	13158074
13189078	13203094	13430051	13560057	13644083	13812058
13846088	13930037	14137024	14331027	14414078	14508080
14689042	14812036	14910063	14978022	15025099	15289011
15336082	15610009	15667017	15675043	15752042	15810064
15838018	15862096	16051004	16293014	16338010	16643042
16835071	16957051	16992080	17511064	17646046	17688001

17785053   17795029   17832023   20083000   20322000   20503000
20527000   21644000

278 cases in total.

On manual scrutiny, 142 of these were plausible, e.g. two miscarriages seven months
apart, listed in separate pregnancies.

However, the following 136 cases were problematic in some way or other. The
middle column contains a 'C' if the case record was changed by this cleaning
exercise.

bserial	Ch?	Comments
00117080		2 live births separated by 7 months, diff pregs. Pregf=3 (i.e.'on time') for second. Assume OK.
00250035	C	Miscarriage, live birth 4 months later, diff.preg. Move to same pregnancy.
00467093		Two live births 6.5 months apart, diff.pregs. Pregf=1 (i.e.'early') for second. Assume OK.
00665097	C	Abortion, live birth 5 months later, diff.preg. Pregf=2 (late) for second. Move to same preg.
00680073		Miscarriage, then live birth 6 months later, same preg. Assume OK.
00700040	C	Miscarriage, then live birth 7 months later, same preg. Assume OK, but re-order.
00757048		Two live births 7 months apart, diff.pregs. Pregf=1 (i.e.'early') for second. Assume OK.
00820070	C	Abortion, live birth 4 months later, diff.preg. Pregf=2 (late) for second. Move to same preg.
00831021		Miscarriage, live birth 8 months later, diff. preg. Pregf=3 ('on time') for second. Assume OK.
00912021	C	Two miscarriages in consecutive months, same preg. Assume OK, but re-order.
00947026		Two live births 8 months apart, diff.pregs. Pregf=1 (i.e.'early') for second. Assume OK.
00951051		Two live births 8 months apart, diff.pregs. Pregf=3 ('on time') for second. Assume OK.
01005054	C	Abortion 5 months after live birth, diff preg. OK, but re-order.
01069012		Miscarriage, then live birth 6 months later, same preg. Assume OK.
01266041	C	Miscarriage, then live birth 6 months later, same preg. Assume OK, but order wrong.
01445068	C	Miscarriage 3 months after live birth, diff. preg, so OK, but order wrong.
01453094		Miscarriage, live birth 7 months later, diff. preg. Pregf=1 ('early') for second. Assume OK .
01628096		2 births 3 months apart, diff.pregs. Pregf=2 (i.e.'late') for both. Must be date error. Leave.
01695003	C	Miscarr, live birth 7 months later,same preg. Pregf=early for second. Assume OK, but re-order
01739024		Miscarriage, then live birth 7 months later, diff.preg. Pregf=early for second. Assume OK.
01761051		Miscarriage, then live birth 7 months later, diff.preg. Pregf='on time' for second. Assume OK.
01840000	C	2 miscarriages 6 months apart, diff.preg. Assume OK, but re-order.
01859054	C	Abortion 3 months after live birth, diff. preg, so OK, but order wrong.
01949055	C	Abortion 1 month after live birth, diff. preg, so OK, but order wrong.
02100033		Miscarr, live birth 5 months later, same preg. Pregf=early for second. Assume OK.
02346044		Miscarr, live birth 8 months later, same preg. Pregf=late for second. Assume OK.
02713099	C	Miscarr, live birth 6 months later, diff. preg. Pregf=late for second. Move to same preg.
03121039	C	2 livebths 3 months apart, diff.pregs. Pregf=on time for both. V.strange. Leave, but re-order.
03279000	C	Live birth 15/5/89, two miscarr. also in 89 (no stated month). Assume separate pregs, re-order.
03349048	C	Miscarriage 4 months after live birth, diff. preg, so OK, but order wrong.
03379077	C	Miscarr, live birth 6 months later, same preg. Assume OK, but order wrong.
03735080		Miscarr.same month as live birth(on time). Must be stillbirth, or (probably) date error. Leave.
03796187		Miscarriage, then live birth 6 months later, same preg. Assume OK.
03997042	C	2 miscarriages 3 months apart, diff.preg. Assume OK, but re-order.
04046069	C	3 miscarriages in different years bunched in same pregnancy. Separate, and re-order.
04222070		2 live births 11 weeks apart, sep. preg. 1st 8lb 4 oz, 2nd 'early', 3lb. Must be date error. Leave.
04277027		2 live births 10 weeks apart, sep. preg. 1st 6lb 9, 2nd 'late', 8lb 12. Must be date error. Leave.

04350025 C Miscarriage, then live birth 4 months later, diff. preg. Move to same pregnancy.
04494008 C 2 live births 5 weeks apart, sep. preg. Both 5lb 12oz, 'early'. Move to same preg.
04498009 C Miscarriage, then live birth 4 months later, same preg. Assume OK, but re-order.
04502049   Miscarriage, then live birth 5 months later, same preg. Assume OK.
05170001 C Miscarriage, then live birth 3 months later, diff. preg. Move to same pregnancy.
05239051   Miscarriage, then live birth 3 months later, same preg. Assume OK.
05531081   2 live births 3 months apart, sep. preg. Both on time, 10lb 4, 9 lb 14. Must be date error. Leave.
05548084 C Abortion 9 months after live birth, so OK, but order wrong.
05661087 C Miscarriage 7 months after live birth, diff.preg, so OK, but order wrong.
05750012   Miscarriage, then live birth 7 months later, same preg. Assume OK.
05796069   Miscarriage, then live birth 7 months later,diff. preg. Live birth premature 1lb 7oz. Assume OK.
05806060 C Miscarriage, then live birth 7 months later, diff.preg, 'on time' 8lb 5 oz. Move to same preg.
05991047 C Abortion, then live birth 3 months later, diff.preg, 'on time' 7lb 14 oz. Move to same preg.
06092005   Miscarriage, then live birth 8 months later, same preg. Live birth 'late'. Assume OK.
06128000   2 live births 3 months apart, sep. preg. Both 'late', both 6lb 10oz. Must be coding error. Leave.
06145027 C Miscarriage, then live birth 5 months later, same preg.   Assume OK, but re-order.
06173005 C Abortion 9 months after live birth, diff. preg, so OK, but order wrong.
06177006 C Abortion, then live birth 4 months later, diff.preg, 'late' 7lb 3 oz. Move to same preg.
06180130 C Miscarriage, then live birth 2 months later, diff.preg. Move to same preg.
06196084 C 3 miscarriages in slots 1,6,11 dated Feb 98, Apr 98, Feb 98 respec. Assume twins+1 sep.preg.
06226027 C Abortion 4 months after live birth, diff. preg, so OK, but order wrong.
06313002 C Miscarriage 4 months after live birth, same preg. Move to diff. preg and re-order.
06394088 C Miscarriage, then live birth 5 months later, same preg.   Assume OK, but re-order.
06481063 C Miscarr, then live birth 8 months later, same.preg, on time, 6lb 7 oz. Assume OK, but re-order.
06521083   Miscarriage, then twins 7 months later, same pregnancy. Assume OK.
06707036   Miscarriage, then live birth 7 months later, same preg.   Assume OK.
06872071   2 live births 5 months apart, sep. preg. Both on time, 8lb 8, 9lb 2. Must be date error. Leave.
06880097 C 2 live births 7 months apart, sep. preg. 2nd 'early' 6lb 12. Could just be plausible, but re-order.
06922067 C Miscarriage, then live birth 3 months later, diff.preg. Move to same preg.
07051003 C Abortion, then stillbirth 5 months later, diff.preg. Stillbirth 'early'. Move to same preg.
07246058   2 live births 6 months apart, sep. preg. Both on time, 7lb 11, 9lb 1. Must be date error. Leave.
07342034 C Miscarriage, live birth 7 months later, same preg. Assume OK, but other babies re-ordered.
07405032 C Miscarriage, then live birth 4 months later, diff.preg. Move to same preg.
08039031 C Abortion, then twins 5 months later (each 5lbs 7 oz) 5 months later. Switch to same preg.
08042081   Miscarriage, then live birth 6 months later, diff.preg, early, 3lb 2 oz. Assume OK.
08121030   2 births 5 days apart,sep.preg, 7lb 14, 8lb 2. Later one early, other on time. Assume date error.
08162010 C Miscarriage, then live birth 7 months later, diff.preg, late, 9lb 2 oz. Move to same pregnancy.
08239086 C Miscarriage 6 months after live birth, diff.preg, so OK, but order wrong.
08460092 C Miscarr, then live birth 6 months later, same.preg, early, 7lb 6 oz. Assume OK, but re-order.
08534041   Miscarr, then live birth 6 months later, same.preg, early, 6lb 11 oz. Assume OK.
08716094 C 2 live births 4 months apart, sep. preg. Both on time, 7lb 6, 7lb 9. Must be date error. Re-order.
08930049   Miscarriage, then live birth 4 months later, same preg.   Assume OK.
09002054 C Miscarriage, then live birth 1 month later, same preg, early, 6lb 1oz. Assume OK, but re-order.
09087040   2 births 4 months apart, sep. preg, 7lb 11, 7lb 9. Later one 'early'. Must be date error. Leave.
09324089   Miscarriage, then live birth 7 months later, same preg, late, 6lb 9oz. Assume OK.
09681052   2 live births 8 months apart, sep. preg, second 2lb 1oz, 'early'. Assume OK.
09708046   2 births 11 days apart,sep.preg 7lb 12, 7lb 5. Later one early,other on time. Assume date error.
09909076   Miscarriage, then live birth 6 months later, same preg, on time, 6lb 4oz. Assume OK.
10231037 C Two live births 10 weeks apart, diff.preg. Move to same preg.
10355068 C Miscarriage, then live birth 7 months later, diff.preg, late, 7lb 13 oz. Move to same pregnancy.
10455045 C Miscarriage 8 months after live birth, diff.preg, so OK, but wrongly bunched with another baby.

10517068	C	Miscarriage, then live birth 5 months later, diff.preg, late, 6lb 13 oz. Move to same pregnancy.
10643073		Miscarriage, then live birth 8 months later, diff.preg, late, 8lb 2 oz. Assume OK.
10675052		Miscarriage, then live birth 7 months later, same preg, early, 6lb 0oz. Assume OK.
10678078		2 live births 8 months apart, sep. preg, second 7lb 9oz, on time. Assume OK.
10723098		Miscarriage Nov 99, in same pregnancy as 'still pregnant' entry. Assume OK.
10928028		2 live births 8.5 months apart, sep. preg, second 6lb 15oz, on time. Assume OK.
11102085		2 live births 3 months apart, sep. preg, 7lb 6, 7lb 5, both 'late'. Must be date error. Leave.
11162042		Miscarriage, then live birth 6 months later, same preg, late, 7lb 6 oz. Assume OK.
11257019	C	Miscarriage Jan 2000, in diff.pregnancy from 'still pregnant' entry. Move to same pregnancy.
11519097	C	Miscarriage, then live birth 2 months later, diff.preg. Move to same pregnancy.
11539049	C	Miscarriage 5 months after live birth, diff.preg, so OK, but re-order.
11541023		2 live births 3 months apart, sep. preg, 6lb 8, 5lb 6, both 'on time'. Must be date error. Leave.
11553050	C	Miscarriage, then live birth 5 months later, diff.preg, early, 5lb 8 oz. Move to same pregnancy.
11841055		2 live births 3 months apart, sep. preg, 7lb 13, 8lb 11, both 'late'. Must be date error. Leave.
11871084	C	Miscarriage 8 months after live birth, diff.preg, so OK, but re-order.
12263073	C	Miscarriage 2 months after live birth, diff.preg, so OK, but re-order.
12619052		Miscarriage, then live birth 5 months later, same preg. Assume OK.
12786037	C	Miscarriage, then live birth 3 months later, diff.preg, late, 7lb 14 oz. Move to same pregnancy.
13158074		2 live births 8 months apart, sep.preg, second 'late', 8lb 4oz. Assume OK anyway.
13189078	C	Miscarriage, then live birth 7 months later, same preg, late, 7lb 10. Assume OK, but re-order.
13203094		Miscarriage, then live birth 6 months later, same preg, on time, 8 lb 8 oz. Assume OK.
13560057		5th outcome is miscarriage in same month as first (live birth). Assume date error. Leave.
13644083		2 live births 6 months apart, sep.preg, second 'late', 6lb 14oz. Assume date error. Leave.
13930037	C	Two miscarriages in consecutive months, same preg. Assume OK, but re-order.
14137024	C	Miscarriage, then live birth 8 months later, same preg, late, 7lb 11. Assume OK, but re-order.
14331027		6 miscarriages within 2 years, all listed sep. pregs, 4 having no stated month. Assume OK.
14414078	C	Miscarriage, then live birth 4 months later, diff.preg. Move to same pregnancy.
14812036	C	Stillbirth 11 days after live birth, same preg. Assume OK, but re-order.
14978022	C	Stillbirth 8 months after livebirth, diff.preg: OK, but wrongly bunched with other birth outcome.
15336082		Miscarriage, then live birth 6-7 months later, diff.preg, early, 7lb 7 oz. Assume OK.
15675043	C	Stillbirth 6 days after live birth, same preg. Assume OK, but re-order.
15810064	C	2 miscarriages 6 months apart, diff.preg: OK, but one wrongly bunched with a 3rd miscarriage.
15838018	C	Abortion, then live birth 4 months later, diff.preg. Move to same preg.
16051004		2 live births 3 months apart, sep.preg, 11lb 10, 10lb 8, both on time. Must be date error. Leave.
16293014	C	Miscarriage a month after livebirth, diff.preg, so OK, but order wrong.
16338010		2 live births 8 months apart, sep.preg, second 9lb 0oz, 'on time'. Assume OK.
16643042	C	Miscarriage, then live birth 6-7 months later, diff.preg, late, 7lb 2 oz. Move to same preg.
16835071		2 live births 5 months apart, sep.preg, no weight data on second. Assume date error. Leave
16957051		Miscarriage, then live birth 4 days later, same preg. Assume OK (should really be 'stillbirth').
16992080	C	Miscarriage, then live birth 7 months later, same preg, early, 10lb 1. Assume OK, but re-order.
17688001		Miscarriage, then live birth 6 months later, same preg, early, 5lb 0 oz. Assume OK
17795029		2 live births 6 months apart, sep.preg, second 7lb 8oz, 'late'. Must be date error. Leave.
20322000		Miscarriage, then stillbirth 5 months later (diff.preg), then live birth 7 months after that, (same preg as still birth), 6lb 9oz, late. Assume stillbirth date erroneous. Leave.
20503000	C	2 miscarriages 3 months apart, diff.preg, so OK, but re-order.
20527000		Miscarriage, then live birth 8-9 months later, diff.preg, early, 3lb 0 oz. Assume OK.
21644000		2 live births 5 weeks apart, sep.preg, 6lb 14, 8lb 3. Later one 'early', other 'on time'. Must be date error. Leave.

Appendix 8

BCS70 cases where an outcome date seems dubious

13235073   Abortion in 1967 – clearly impossible
01620094   Live birth in 1980, when (male) respondent was aged 10
11586004   Abortion in 1982, when (female) respondent aged 12.

[There were no corresponding NCDS cases, as pregnancy history was only asked from age 33 onwards]

Appendix 9

Cases where everpreg=1, but no outcome data or dates are entered for any outcome.

Respondent answered 'Yes' to question 'Have you ever been or got anyone else pregnant?', but gave no informative data about any pregnancies.

NCDS

289155Y	383020Y	433043M	450006B	509203C	511019Y
550264J	565093R	620103V	683026K	750049U	782126R
950109A	X34002E	X80205F			

15 cases in total.

Also, NCDS cohort member 383032F had one pregnancy, outcome ='8' ('don't know'), no outcome date, no other data at all. Male CM.

BCS70

00270088	01470020	02635024	03596058	03716002	03820004
04520051	05265079	05343053	05364081	05375032	05551033
05863092	06017072	06217026	06524008	06618010	07111076
07114001	08136083	08201055	08679000	09930027	10121084
10185042	10283069	10329040	10873056	10898085	10969008
11495028	12141093	12373026	12461077	12589008	12590007
13355002	13677037	14509055	14512004	14548085	14576063
14693067	15348008	15399065	16513036	16741069	17331062
17535017	17575022	20025000			

51 cases in total.

Appendix 10

Result of manual scrutiny of all 'true' NCDS multiple births (i.e. where birth outcomes separated by no more than 3 days).

The following 117 cases had 'true' multiple outcomes. All were examined manually. 16 were problematic (see table below), of which 13 were altered (column 'Ch?'=C). All other cases were correctly entered.

043007J	043013C	044001A	044025R	050070F	055009E
055012T	083090Y	087011W	092098S	092261A	092337J
093176R	096026M	100010M	100024Y	100050Z	100076U
100080K	110251S	120135W	142009Q	183016T	184021R
186128W	186157D	186169M	187020E	188030P	200072T
215003F	235007C	235035J	286031N	287085T	287143E
289072U	289091Y	289133N	289141M	300094M	310029F
325002T	330027Q	340007Q	350145K	421030K	422030Q
422088Y	425010Z	433026M	434031K	481013Y	500015C
500069C	500208Q	500219V	500224N	500420N	500422S
500534D	503009Z	509206K	509254W	510161W	511116W
512069V	517054J	527001S	550118W	550232U	561007Y
565085S	620019H	630060L	640031K	650053B	680011D
681047H	730084J	750050C	781048S	782112D	784010F
821014S	821511H	822027H	850021C	850027R	883035A
950079U	950081E	950213V	950224A	950290Q	980004E
982067S	985071Y	986100J	986205Y	986275W	987069Y
X32040C	X37008K	X38008Q	X70072H	X70073K	X77019R
X80124F	X82025Q	X82193N	Y00174D	Y00293N	Y01578L
Y20162K	Y21034E	Y30137S			

Nserial	Ch?	Comments
043013C	C	Double miscarriage, with one other livebirth in wrong order
044001A	C	Double miscarriage, with single miscarriage & three other livebirths in wrong order
183016T	C	Twins bunched with single birth, wrong order. Delete pre-March 91 twins.
188030P	C	Miscarriage, then twins seven months later, wrong order: assume same pregnancy
500219V	C	Double miscarriage, but placed in separate pregnancies. Move to same pregnancy
500015C	C	Twins born on separate days entered in wrong order chronologically.
509254W	C	Twins born on separate days entered in wrong order chronologically.
620019H		Two miscarriages in same year, sep. pregnancies: day and month not stated. Leave as is.
730084J	C	Twins entered correctly, but one other outcome entered out of chronol.order.
850027R		Miscarriage, then twins five months later: assume same pregnancy, so leave as is.
950290Q	C	Miscarriage+double miscarriage 6 months later, then twins 1 month after that(?!), wrong order.
980004E	C	Miscarriage, then twins seven months later. Assume same pregnancy, but change order
986100J	C	Double miscarriage, wrongly bunched with miscarriage 4 years earlier
X38008Q	C	Twins entered as though in two separate pregnancies
Y00174D	C	Double miscarriage, entered as though two separate pregnancies
Y00293N		Miscarriage, then twins six months later: assume same pregnancy, so leave as is.

Appendix 10 (cont).

Result of manual scrutiny of all 'true' BCS70 multiple births (i.e. where birth outcomes separated by no more than 3 days).

The following cases had 'true' multiple outcomes. All were examined manually. 26 were problematic (see table below), of which 24 were altered (column 'Ch?'=C). All other cases were correctly entered.

00046056	00048006	00050081	00060057	00219007	00251010
00453015	00508088	00509063	00640068	00784051	01164013
01328064	01415039	01622044	01717021	01778054	01788030
02125062	02129063	02706048	02755054	02854056	02924004
03099246	03209066	03235094	03403069	03562028	03634027
03667082	03783010	03947061	04149072	05122071	05288057
05435004	05474034	05521004	05671063	05718009	05942041
06124000	06196084	06241003	06253030	06288035	06295086
06382061	06430006	06514032	06521083	06548062	06685018
06801062	06982024	07015000	07085033	07104025	07149006
07339085	07342034	07370012	07379090	07406007	07454038
08039031	08129032	08356090	08535016	08568071	08644095
08879054	08995083	09252090	09352067	09353042	09418091
09735049	09898009	10212060	10231037	10531069	10556098
10656075	10822000	11058040	11155092	11846031	12150094
12278025	12350048	12364025	12610075	12693010	12704077
12958013	13211019	13371054	13846088	13931012	13972093
14046048	14068051	14089227	14314000	14335028	14476086
14901062	14989074	15152079	15156080	15372086	15416006
15418057	15503082	15610009	15908067	16102076	16349062
16530063	16556067	17204082	17486097	20083000	20515000
21179000					

127 cases in total.

bserial	Ch?	Comments
00508088	C	3 babies in slots 1, 6, 11 had dates 20/7/99, 3/7/99, 20/7/99 respec. Assume triplets.
02706048	C	Two babies in slots 1,6 with same birth date. Move 6 to slot 2.
03403069	C	Slots 16,21 and 26 all say 'still pregnant.' Assume triplets, move to slots 1,2,3.
03783010	C	Slots 1 and 11 both say 'still pregnant'. Assume twins, move to slots 1,2.
03947061	C	Slots 1 and 6 have same birth date. Assume twins, move to slots 1,2.
05521004	C	Twins bunched in same pregnancy as baby born a year later.
05671063	C	Slots 1 and 11 have same birth date. Assume twins, move to slots 1,2.
06196084	C	3 miscarriages in slots 1,6,11 dated Feb 98, Apr 98, Feb 98 respec. Assume twins+1 sep.preg.
06521083		Miscarriage, then twins 7 months later, same pregnancy. Assume OK.
06548062	C	7 babies (of whom two are twins). The sixth was out of chronol. Order.
06982024	C	Twins, plus one other baby, out of chronol. order.
07149006	C	Double stillbirth in slots 11 and 12 more recent than other two children. Move to 1,2.

07339085 C Twins, plus two other babies, out of chronol. order.
07342034 C Twins placed in slots 26, 27 when 11,16,21 vacant. Move to slots 11-12.
08039031 C Abortion, then twins 5 months later (each 5lbs 7 oz) 5 months later. Switch to same preg.
08129032   Live birth, then double abortion 3 months later, diff.preg. Assume OK.
08995083 C Two other babies from years earlier bunched in same pregnancy as twins.
10822000 C Two babies with same d.o.b. in slots 1,11. Move to slots 1,2.
11155092 C Twins in slots 1,2, then twins in slots 6 & 7 with same birth dates. DELETE 2nd pair*.
12150094 C Twins, plus three other babies, out of chronol. order.
12693010 C Two miscarriages in same month, in slots 11 and 16. Assume double miscarriage.
12958013 C Twins, plus two other babies, out of chronol. order and all bunched in same pregnancy.
14046048 C Two abortions, both June 1988, in slots 11 and 26.   Assume 26 is coding error, and DELETE.
14989074 C 2 live births, same date, in slots 1, 11. Both 9 lb 2 oz!! Assume 11 is coding error -DELETE.
15372086 C Twins, plus one other baby, out of chronol.order.
16102076 C Twins, plus two other babies, out of chronol. order and all bunched in same pregnancy.

* The birth weights of the first two were 4lb 9 oz and 5lb 2oz. The second pair were 4 lb 9oz and
unspecified. It seems very unlikely indeed that quads could have such high birthweights.   We
therefore assumed this was a data entry duplication (the fact that two birth weights are exactly the same
also supports this hypothesis).

Appendix 11

Complete list of all cases altered by data cleaning process

NCDS

010004R	010101P	010111S	010184W	010185Y	010223D
041034A	043003Z	043013C	043025L	043058B	044001A
044014L	044024P	045002J	045011K	045012M	046014W
046028J	050059U	050069X	051003V	052008M	053003F
055013V	055086Z	056052M	075006M	081002N	081012R
081019F	083034N	084004J	084015P	084027W	085010J
087015E	087029S	088005H	088012D	090001M	090028J
091017J	091030Z	092023J	092024L	092053T	092179S
092206S	092293Q	092405Y	093032Q	093039E	093052W
093055C	093063B	093087S	093234C	093259V	093272M
093273P	094028E	094029H	094040U	095006Z	095023Z
095045L	097011C	099010M	099050Z	100009C	100031V
100055L	110030Z	110065V	110079H	110080R	110097K
110135Q	110193E	110213J	110244V	110290C	110304M
120017Q	120048B	120070U	120080X	120132Q	120141R
130007T	130008V	180001N	180007A	181038T	182007M
183005N	183016T	184047L	185014Z	186004B	186081W
186125Q	186238D	187019W	188030P	188108A	200006D
200075Z	200086E	216003M	218005B	223020B	226004V
233004L	233008U	235003U	235011T	235018J	239005V
280005D	280017M	280034M	280041J	280053R	280064W
280068E	282012M	282029E	282104S	282109C	282113T
283003R	284011V	285020B	285032K	285052R	285061S
286042T	286044X	286058K	287069V	287150B	287210T
288009F	288027J	289031D	289063T	289123K	289131J
289141M	289180X	300080Z	300093K	300133U	308053Q
310016W	310017Y	310101M	310107Z	325016E	330027Q
330055V	330073X	340061W	350110P	350139Q	360027K
380031N	380041R	380084L	380090E	381111R	381127H
382004V	383051L	385004M	385005P	385020K	385025V
385053A	385110L	385173M	400027C	400075Q	405014V
405042A	405050Z	420022E	420025M	420038W	420052Q
421001B	421030K	422073K	422081J	423006Y	423045K
424029S	431038J	431040U	432006Z	432025D	434020D
434035T	435002H	436010M	436011P	440006V	450075X
460028U	460039Z	460043Q	465048C	465051R	481001R
481044L	482031F	500015C	500032C	500057V	500080Q
500113C	500124J	500125L	500219V	500234R	500353Z
500375L	500376N	500418B	500423U	501009P	504020S
504034D	506014J	509205H	509228V	509254W	510024L
510097Q	510127W	510153X	511063B	511159R	512002L
512012P	512044C	512095W	513018H	513033C	513053K
513073R	513083U	513117K	513118M	513143L	514012Z

514121E	515034R	515085K	516085Q	517012R	517016Z
517045H	517057Q	517213B	518025F	518029Q	518037P
518069C	518134M	520049K	520056F	520064E	520086R
522007C	523015H	523033K	524002C	524004H	524005K
525013P	526007Z	527001S	528014H	528020B	528021D
528083C	529004K	529009V	550026R	550033N	550158K
550188U	550245D	550289A	550379B	550388C	550534M
560008V	560022P	560031Q	561006W	565027B	565036C
565063F	565108B	580011W	581050N	581068J	583049Q
591036A	591050U	591052Y	593013Y	610028B	610046D
610070A	610087U	620005V	620037K	620075T	620087A
620107D	630002V	630006D	650025W	650116Z	650122U
650132X	650164M	650172L	650191Q	650212V	683009K
683019N	684035R	684054V	710034D	710063M	710068X
720019Q	720027P	720098P	730015N	730049F	730050Q
730084J	730130N	750043F	750075V	750101T	781005X
782013B	782054S	782073W	782114J	783041N	805004X
815048A	821507S	821519Z	823501Q	823502S	824020C
825028A	825049K	825115V	835013T	840012V	845039U
850012B	850032J	860001C	865021M	880017H	881014F
881015J	882007Q	882019X	882021J	882037Z	882051T
882061W	910031N	931002Y	933019C	936012D	937020J
950114T	950123U	950124W	950135B	950154F	950155J
950168T	950266T	950288D	950290Q	960003Q	960022U
960029K	960031V	960046K	960072L	960073N	980004E
982053E	983033D	984024J	985039C	985067J	986027A
986039J	986059Q	986100J	986117B	986183R	986250D
986284X	986374Y	986411B	986430F	986431J	987013U
987048Q	987097D	987100P	989011A	989041L	X10020X
X25029A	X25050R	X25056D	X25057F	X25101F	X32005A
X32044M	X32084Z	X32152Q	X32159E	X33020B	X34011F
X37008K	X38005J	X38008Q	X40028K	X40036J	X41020X
X60006M	X66030R	X67026F	X70045D	X70066N	X71009E
X76010R	X77014E	X77016K	X77021B	X77029U	X78016Q
X78019W	X78043T	X78079R	X78086N	X79021N	X80013W
X80020T	X80034E	X82062W	X82115R	X82137B	X82215V
X82250X	X82256L	X82276S	X82282M	X82287X	X82344H
X82347P	X84009C	X85003V	X86018Q	X87024Q	X87061W
X88003M	Y00087K	Y00130H	Y00164A	Y00173B	Y00174D
Y00274J	Y00337F	Y01018U	Y01189Y	Y01605L	Y20023U
Y20055J	Y20078W	Y20134D	Y20154L	Y20187B	Y21001P
Y21037M	Y30072Q	Y30150J	Y30168D	Y30169F	Y30223K
Y33089Z					

511 cases in total.

BCS70

00051056	00103002	00124030	00250035	00279065	00312058
00346088	00387068	00463092	00508088	00520038	00530014
00607090	00620015	00665097	00700040	00785026	00809095
00813019	00820070	00879028	00912021	00914072	01001053
01005054	01065011	01153062	01236012	01251089	01254014
01266041	01302060	01445068	01576049	01592000	01618019
01620094	01674076	01695003	01736099	01763001	01840000
01859054	01870029	01945054	01949055	01993085	02051012
02126037	02182094	02207037	02252042	02294098	02306039
02441045	02454047	02478000	02609097	02706048	02713099
02753003	02765030	02830002	02880084	02896011	02904052
02914028	02930080	02934081	02939057	03085020	03121039
03134041	03178047	03191073	03257097	03264047	03279000
03310042	03325095	03349048	03366075	03371075	03379077
03403069	03418021	03520221	03556053	03575030	03579031
03600098	03621025	03691059	03745056	03749057	03783010
03786036	03795037	03884063	03894039	03926033	03947061
03956062	03982090	03997042	04046069	04139096	04177050
04239073	04350025	04361077	04394031	04452053	04474056
04486083	04494008	04498009	05014068	05036071	05098079
05137023	05170001	05227024	05236025	05249027	05289032
05313024	05325051	05369057	05399086	05408250	05425028
05486061	05506028	05521004	05533031	05542032	05548084
05585063	05593089	05661087	05671063	05721207	05749013
05794018	05806060	05832088	05895071	05910062	05927065
05984097	05991047	06028023	06031073	06057077	06072053
06123024	06145027	06173005	06177006	06180130	06183082
06195008	06196084	06198034	06226027	06266032	06299087
06305077	06313002	06393012	06394088	06458061	06466087
06467062	06471087	06481063	06501030	06507082	06548062
06570089	06607059	06612059	06619086	06641012	06666041
06691094	06791071	06797022	06853094	06880097	06896024
06903090	06908066	06922067	06959022	06982024	06992000
06998052	07019000	07020000	07051003	07052079	07061080
07149006	07191086	07194011	07211053	07244007	07326083
07338009	07339085	07342034	07359037	07391040	07405032
07416084	07469091	07494043	08006077	08033080	08039031
08052057	08078061	08162010	08163086	08215032	08239086
08263063	08267064	08281065	08283015	08302007	08388069
08409011	08445015	08460092	08477095	08531015	08632068
08657097	08700066	08716094	08787002	08818021	08962028
08993032	08995083	09002054	09027083	09047035	09054086
09073063	09145062	09204059	09261091	09262066	09310011
09318013	09362043	09393047	09490099	09495075	09516017
09541070	09567074	09573049	09591051	09598078	09667125
09716072	09725073	09747076	09758027	09763027	09797057
09799007	09803047	09840026	09891083	09982059	10047010
10141036	10215086	10231037	10299097	10308012	10312037

10355068	10455045	10463071	10477048	10517068	10527044
10582025	10657050	10667026	10732099	10788031	10796057
10822000	10868056	10894084	10921001	10927053	10937029
10964032	11032036	11047089	11059015	11073016	11086018
11155092	11212038	11257019	11281097	11314066	11391050
11414043	11487002	11515096	11519097	11539049	11553050
11556076	11636000	11637076	11677081	11706049	11871084
11937007	12074070	12091097	12124066	12150094	12155070
12156045	12263073	12394054	12434074	12464002	12522024
12596059	12609076	12693010	12733030	12786037	12958013
13045095	13054096	13062021	13074048	13189078	13346001
13436002	13468082	13538030	13561032	13705030	13740059
13741034	13762136	13766063	13771063	13788066	13799017
13930037	13945090	14046048	14101044	14137024	14185055
14228000	14249028	14293058	14371032	14381008	14414078
14479011	14538008	14574012	14644061	14671064	14687092
14699018	14812036	14858093	14869044	14924040	14978022
14989074	15013072	15084081	15093082	15094057	15095032
15136027	15347033	15351058	15372086	15440084	15455036
15511007	15527035	15567040	15571065	15675043	15703036
15737066	15810064	15838018	15935070	15941045	16027027
16076033	16093060	16100025	16102076	16152057	16170059
16224056	16281088	16293014	16336060	16354062	16460014
16481042	16563017	16568094	16630040	16643042	16717092
16776074	16809043	16822069	16922046	16992080	17015079
17062034	17139009	17163087	17170037	17267065	17270014
17292017	17327037	17371067	17373017	17386019	17456068
17472019	17624043	17687026	18013006	18046061	18132061
20043000	20091000	20097000	20212000	20305000	20345000
20432000	20499000	20503000	20531000	21667000	21740000

468 cases in total.

Appendix 12

SPSS Syntax for 'patch' to re-order cases with outcomes wrongly bunched or out of chronological order, to examine miscellaneous anomalous cases, and to eliminate pre-March 1991 NCDS pregnancy histories

* 'Undeclare' all missing values for outcome and date of outcome variables, otherwise SPSS 'If' test for values '99', '9999' won't work.

missing values prega prega2 prega3 prega4 prega5 prega6 prega7 prega8 prega9 prega10 prega11 prega12 prega13 prega14 prega15 prega16 prega17 prega18 prega19 prega20 prega21 prega22 prega23 prega24 prega25 prega26 prega27 prega28 prega29 prega30 prega31 prega32 prega33 prega34 prega35 prega36 prega37 prega38 prega39 prega40 preged preged2 preged3 preged4 preged5 preged6 preged7 preged8 preged9 preged10 preged11 preged12 preged13 preged14 preged15 preged16 preged17 preged18 preged19 preged20 preged21 preged22 preged23 preged24 preged25 preged26 preged27 preged28 preged29 preged30 preged31 preged32 preged33 preged34 preged35 preged36 preged37 preged38 preged39 preged40 pregem pregem2 pregem3 pregem4 pregem5 pregem6 pregem7 pregem8 pregem9 pregem10 pregem11 pregem12 pregem13 pregem14 pregem15 pregem16 pregem17 pregem18 pregem19 pregem20 pregem21 pregem22 pregem23 pregem24 pregem25 pregem26 pregem27 pregem28 pregem29 pregem30 pregem31 pregem32 pregem33 pregem34 pregem35 pregem36 pregem37 pregem38 pregem39 pregem40 pregey pregey2 pregey3 pregey4 pregey5 pregey6 pregey7 pregey8 pregey9 pregey10 pregey11 pregey12 pregey13 pregey14 pregey15 pregey16 pregey17 pregey18 pregey19 pregey20 pregey21 pregey22 pregey23 pregey24 pregey25 pregey26 pregey27 pregey28 pregey29 pregey30 pregey31 pregey32 pregey33 pregey34 pregey35 pregey36 pregey37 pregey38 pregey39 pregey40 ().

* Declare vectors corresponding to all 'baby' variables (length 40), and 'pregnancy' variables (length 8),
* so we can process all the slots using loop structures.

vector vpregnu vpregj vcg1prg vcg2prg vcg3prg vcpregb vmorprg(8, f1.0).
vector vpregk vpregkw(8,f3.0).
vector vpregl(8,f2.0).
vector vprega vpregc vpregd vkilo vpregf vpregh vwhopa vwerkd vabspa
        vabspb vabspc vabskb vabskc vabskd vabske (40,f1.0).
vector vpound vounce vprged vprgem vpregg vwhopb vabska vabsmn(40,f2.0).
vector vgramm vprgey vabsyr(40,f4.0).
vector vprege(40,a8).
vector vpregi(40,a100).

* Declare second copy of the vectors, to facilitate the process of placing everything in new positions after re-ordering
* (as the original variables are unfortunately not consecutive in the data file, it's easier to do it with a fresh lot of vectors.
* It also saves having to have hundreds of 'compute' statements to initialise all the variables to system-missing, as these

```
*  vectors all start off system-missing once declared).

vector xpregnu xpregj xcg1prg xcg2prg xcg3prg xcpregb xmorprg(8, f1.0).
vector xpregk xpregkw(8,f3.0).
vector xpregl(8,f2.0).
vector xprega xpregc xpregd xkilo xpregf xpregh xwhopa xwerkd xwatkd xabspa
        xabspb xabspc xabskb xabskc xabskd xabske(40,f1.0).
vector xpound xounce xprged xprgem xpregg xwhopb xabska xabsmn(40,f2.0).
vector xgramm xprgey xabsyr(40,f4.0).
vector xprege(40,a8).
vector xpregi(40,a100).

*Declare vector BIRDAT to compare all forty possible outcome dates chronologically.
vector birdat(40, f6.0).

*Declare vector 'key' to keep track of which slot the data relating to the nth most recent
pregnancy was originally placed in.
* This is used in the automatic re-ordering of cases with no multiple births, dubious outcomes
or other horrible complications.
vector key(8, f1.0).

*Declare vector 'key' to keep track of which of the 40 'baby' slots the data relating to the nth
most recent baby was originally
* placed in.  This is used in the re-ordering of individual cases from the manual scrutiny.
vector key40(40, f1.0).

*Declare vector 'keypreg' to keep track of which of the 8 'pregnancy' slots the data relating to
the nth most recent pregnancy was originally
* placed in.  This is used in the re-ordering of individual cases from the manual scrutiny.
vector keypreg(8, f1.0).

*Declare vector 'compdt' to create new composite date variable which includes '99' codes and
imputed figures for missing days and months.
vector compdt (40,f8.0).

*Declare vector 'outdat' to keep track of which is the most recent pregnancy outcome, during
the process of re-ordering them.
vector outdat(40, f6.0).

* Don't look at any CM who's never been/got anybody pregnant.
Do if(everpreg=1).

* Define vectors for date and pregnancy outcome variables, needed to sort out wrong
order/bunching problems etc.
* Regrettably these (and later, all the other 35 pregnancy history variables) all have to be
defined/copied in this
* long-winded way, because the 40 versions of each variable are not consecutive on the
1999/2000 dataset.

compute vprega1=prega.
compute vprega2=prega2.
compute vprega3=prega3.
compute vprega4=prega4.
compute vprega5=prega5.
compute vprega6=prega6.
compute vprega7=prega7.
compute vprega8=prega8.
compute vprega9=prega9.
compute vprega10=prega10.
compute vprega11=prega11.
```

```
compute vprega12=prega12.
compute vprega13=prega13.
compute vprega14=prega14.
compute vprega15=prega15.
compute vprega16=prega16.
compute vprega17=prega17.
compute vprega18=prega18.
compute vprega19=prega19.
compute vprega20=prega20.
compute vprega21=prega21.
compute vprega22=prega22.
compute vprega23=prega23.
compute vprega24=prega24.
compute vprega25=prega25.
compute vprega26=prega26.
compute vprega27=prega27.
compute vprega28=prega28.
compute vprega29=prega29.
compute vprega30=prega30.
compute vprega31=prega31.
compute vprega32=prega32.
compute vprega33=prega33.
compute vprega34=prega34.
compute vprega35=prega35.
compute vprega36=prega36.
compute vprega37=prega37.
compute vprega38=prega38.
compute vprega39=prega39.
compute vprega40=prega40.


compute vprged1=preged.
compute vprged2=preged2.
compute vprged3=preged3.
compute vprged4=preged4.
compute vprged5=preged5.
compute vprged6=preged6.
compute vprged7=preged7.
compute vprged8=preged8.
compute vprged9=preged9.
compute vprged10=preged10.
compute vprged11=preged11.
compute vprged12=preged12.
compute vprged13=preged13.
compute vprged14=preged14.
compute vprged15=preged15.
compute vprged16=preged16.
compute vprged17=preged17.
compute vprged18=preged18.
compute vprged19=preged19.
compute vprged20=preged20.
compute vprged21=preged21.
compute vprged22=preged22.
compute vprged23=preged23.
compute vprged24=preged24.
compute vprged25=preged25.
compute vprged26=preged26.
compute vprged27=preged27.
compute vprged28=preged28.
compute vprged29=preged29.
```

```
compute vprged30=preged30.
compute vprged31=preged31.
compute vprged32=preged32.
compute vprged33=preged33.
compute vprged34=preged34.
compute vprged35=preged35.
compute vprged36=preged36.
compute vprged37=preged37.
compute vprged38=preged38.
compute vprged39=preged39.
compute vprged40=preged40.


compute vprgem1=pregem.
compute vprgem2=pregem2.
compute vprgem3=pregem3.
compute vprgem4=pregem4.
compute vprgem5=pregem5.
compute vprgem6=pregem6.
compute vprgem7=pregem7.
compute vprgem8=pregem8.
compute vprgem9=pregem9.
compute vprgem10=pregem10.
compute vprgem11=pregem11.
compute vprgem12=pregem12.
compute vprgem13=pregem13.
compute vprgem14=pregem14.
compute vprgem15=pregem15.
compute vprgem16=pregem16.
compute vprgem17=pregem17.
compute vprgem18=pregem18.
compute vprgem19=pregem19.
compute vprgem20=pregem20.
compute vprgem21=pregem21.
compute vprgem22=pregem22.
compute vprgem23=pregem23.
compute vprgem24=pregem24.
compute vprgem25=pregem25.
compute vprgem26=pregem26.
compute vprgem27=pregem27.
compute vprgem28=pregem28.
compute vprgem29=pregem29.
compute vprgem30=pregem30.
compute vprgem31=pregem31.
compute vprgem32=pregem32.
compute vprgem33=pregem33.
compute vprgem34=pregem34.
compute vprgem35=pregem35.
compute vprgem36=pregem36.
compute vprgem37=pregem37.
compute vprgem38=pregem38.
compute vprgem39=pregem39.
compute vprgem40=pregem40.


compute vprgey1=pregey.
compute vprgey2=pregey2.
compute vprgey3=pregey3.
compute vprgey4=pregey4.
compute vprgey5=pregey5.
```

```
compute vprgey6=pregey6.
compute vprgey7=pregey7.
compute vprgey8=pregey8.
compute vprgey9=pregey9.
compute vprgey10=pregey10.
compute vprgey11=pregey11.
compute vprgey12=pregey12.
compute vprgey13=pregey13.
compute vprgey14=pregey14.
compute vprgey15=pregey15.
compute vprgey16=pregey16.
compute vprgey17=pregey17.
compute vprgey18=pregey18.
compute vprgey19=pregey19.
compute vprgey20=pregey20.
compute vprgey21=pregey21.
compute vprgey22=pregey22.
compute vprgey23=pregey23.
compute vprgey24=pregey24.
compute vprgey25=pregey25.
compute vprgey26=pregey26.
compute vprgey27=pregey27.
compute vprgey28=pregey28.
compute vprgey29=pregey29.
compute vprgey30=pregey30.
compute vprgey31=pregey31.
compute vprgey32=pregey32.
compute vprgey33=pregey33.
compute vprgey34=pregey34.
compute vprgey35=pregey35.
compute vprgey36=pregey36.
compute vprgey37=pregey37.
compute vprgey38=pregey38.
compute vprgey39=pregey39.
compute vprgey40=pregey40.
```

* Use variable totbaby to calculate number of babies carried in total, by checking each
* of the 40 possible slots for any date or outcome.

```
compute totbaby=0.
format totbaby(f1.0).
```

```
loop #i=1 to 40.
If (not(sysmis(vprega(#i))) or not(sysmis(vprgey(#i))) or not(sysmis(vprgem(#i))) or
not(sysmis(vprged(#i))))totbaby=totbaby+1.
end loop.
```

* Initialise variable 'noyrflag' to log cases where any outcome has a year '9999'.
* These cases must be excluded from the automatic re-ordering,and scrutinised manually.

```
compute noyrflag=0.
format noyrflag(f1.0).
compute pre91flg=0.
format pre91flg(f1.0).
```

* Construct date-of-birth variables which can be measured ordinally and subtracted from each
other,
* using yrmoda function, which produces number of days elapsed since 15/10/1582.
* Flag up cases where year=9999, or prega (outcome)=9 or 8, for manual scrutiny.

* Impute day and month as 15, 6 respectively if not available, as long as year has been entered sensibly.
* Impute date 30th Sept 2000 for any ongoing pregnancies.

```
loop #i=1 to 40.

Do if(vprgey(#i) ge 1900 and vprgem(#i) ge 1 and vprgem(#i) le 12 and vprged(#i) ge 1 and vprged(#i) le 31).
compute birdat(#i)=yrmoda(vprgey(#i),vprgem(#i),vprged(#i)).
compute compdt(#i)=1000000*vprged(#i)+10000*vprgem(#i)+vprgey(#i).
end if.

Do if(vprgey(#i)=9999 or vprgey(#i)=9998 or vprega(#i)=9 or vprega(#i)=8).
compute noyrflag=1.
compute compdt(#i)=99999999.
end if.

Do if(vprgey(#i) ne 9999 and vprgey(#i) ne 9998 and (vprged(#i) ne 99) and (vprgem(#i)=99 or sysmis(vprgem(#i)))).
compute birdat(#i)=yrmoda(vprgey(#i),6,vprged(#i)).
compute compdt(#i)=1000000*vprged(#i)+60000+vprgey(#i).
end if.

Do if(vprgey(#i) ne 9999 and vprgey(#i) ne 9998 and (vprged(#i)=99 or (vprged(#i)=98) or sysmis(vprged(#i))) and (vprgem(#i)=99 or sysmis(vprgem(#i)))).
compute birdat(#i)=yrmoda(vprgey(#i),6,15).
compute compdt(#i)=15060000+vprgey(#i).
end if.

Do if(vprgey(#i) ne 9999 and vprgey(#i) ne 9998 and vprgem(#i) ne 99 and (vprged(#i)=99 or vprged(#i)=98 or sysmis(vprged(#i)))).
compute birdat(#i)=yrmoda(vprgey(#i),vprgem(#i),15).
compute compdt(#i)=15000000+10000*vprgem(#i)+vprgey(#i).
end if.

Do if(vprega(#i)=5).
compute birdat(#i)=yrmoda(2000,9,30).
compute compdt(#i)=30092000.
end if.


end loop.


* Flag up all cases where birth dates are out of order.

compute ordrflag=0.
format ordrflag (f1.0).

loop #i=1 to 39.
loop #j=1 to #i.
if (birdat(1+#i)>birdat(#j))ordrflag=1.
end loop.
end loop.
```

* Flag up all cases where outcomes more than nine months apart are bunched as though part of the same pregnancy.

```
compute bunched=0.
format bunched (f1.0).
```

```
loop #i=1 to 8.
loop #j=1 to 4.
if ((birdat(5*#i-4+#j)-birdat(5*#i-4) ge 273) or (birdat(5*#i-4)-birdat(5*#i-4+#j) ge
273))bunched=1.
end loop.
end loop.
```

*Flag up all cases where there is an ongoing pregnancy.

```
compute stilpreg=0.
format stilpreg (f1.0).
loop #i=1 to 40.
if (vprega(#i)=5)stilpreg=1.
end loop.
```

* Get all valid birth dates in chronological order, up to a maximum of 8 births.
* Use 'key' vector to keep track of which birthslot the date was originally placed in, so as to
* line up all the other pregnancy variables corresponding with this birth into their correct slots.
* After each of the 8 iterations resulting in determining a 'key' number, set the outcome date
* for the slot that key points to, to zero, so it will no longer be the most recent in the next
iteration.

```
loop #j=1 to 8.
compute outdat(#j)=0.
compute key(#j)=0.
loop #i=1 to 40.
Do if (birdat(#i)>outdat(#j)).
compute outdat(#j)=birdat(#i).
compute key(#j)=#i.
end if.
end loop.
if (key(#j) ne 0)birdat(key(#j))=0.
end loop.
```

* As a check that everything worked OK, look at whether any birdat values are still non-zero.
```
compute missdany=0.
format missdany (f1.0).
```

```
loop #i=1 to 40.
If (birdat(#i) ne 0)missdany=1.
end loop.
```

* Create variable truemult to flag up all cases with true multiple births: i.e. those where the
outcome date is the same,
* or no more than 3 days apart.

```
compute truemult=0.
format truemult (f1.0).
loop #i=1 to 7.
if ((outdat(#i)-outdat(#i+1) le 3) and (outdat(#i+1) ne 0))truemult=1.
end loop.
```

* Create warning flag variable oddmult to flag outcome dates less than nine months apart, but
greater than 3 days apart.

```
compute oddmult=0.
```

```
format oddmult (f1.0).
loop #i=1 to 7.
if ((outdat(#i)-outdat(#i+1) gt 3) and (outdat(#i)-outdat(#i+1) le 273))oddmult=1.
end loop.
```

* Flag up any NCDS cases with an event prior to March 1991, who took part in the NCDS 5
survey.
* For them we delete this extraneous information, by setting the 'key' variable to 40, so that
blank data is copied into that
* birth slot (we know no-one in NCDS had a birth event in the 40th slot).
* If they weren't interviewed in 1991, we leave the data as it is anyway.

```
loop #i=1 to 8.
Do if(outdat(#i) gt 0 and (outdat(#i) le yrmoda(1991,03,01)) and (dmpart=1)).
compute key(#i)=40.
compute pre91flg=1.
end if.
end loop.
```

* Re-arrange all 39 pregnancy history variables so that all the other information about each
outcome is
* in the slot corresponding to the re-arranged birth date. Need first to copy variables to their
v-prefix vectors, use loop
* structure to copy them to their correct position (x-prefix vectors), then copy them back to
their ordinary names from
* the x-prefix vectors.

* Note this process only occurs for cases where we know the outcomes have ben wrongly
bunched in the same pregnancy,
* or where the outcomes are in the wrong chronological order. Even if this is the case, we
don't proceed if there are any
* outcomes with a year entered as '9999', or where we know there are any true multiple births,
or any dubious multiple
* births (i.e. >3 days but less than 9 months apart). All these latter cases are flagged for
manual scrutiny.

```
compute vpregnu1=pregnum.
compute vpregnu2=pregnum2.
compute vpregnu3=pregnum3.
compute vpregnu4=pregnum4.
compute vpregnu5=pregnum5.
compute vpregnu6=pregnum6.
compute vpregnu7=pregnum7.
compute vpregnu8=pregnum8.

compute vpregj1=pregj.
compute vpregj2=pregj2.
compute vpregj3=pregj3.
compute vpregj4=pregj4.
compute vpregj5=pregj5.
compute vpregj6=pregj6.
compute vpregj7=pregj7.
compute vpregj8=pregj8.

compute vcg1prg1=cgprega1.
compute vcg1prg2=cgprega4.
compute vcg1prg3=cgprega7.
compute vcg1prg4=cgpreg10.
compute vcg1prg5=cgpreg13.
```

```
compute vcg1prg6=cgpreg16.
compute vcg1prg7=cgpreg19.
compute vcg1prg8=cgpreg22.

compute vcg2prg1=cgprega2.
compute vcg2prg2=cgprega5.
compute vcg2prg3=cgprega8.
compute vcg2prg4=cgpreg11.
compute vcg2prg5=cgpreg14.
compute vcg2prg6=cgpreg17.
compute vcg2prg7=cgpreg20.
compute vcg2prg8=cgpreg23.

compute vcg3prg1=cgprega3.
compute vcg3prg2=cgprega6.
compute vcg3prg3=cgprega9.
compute vcg3prg4=cgpreg12.
compute vcg3prg5=cgpreg15.
compute vcg3prg6=cgpreg18.
compute vcg3prg7=cgpreg21.
compute vcg3prg8=cgpreg24.

compute vcpregb1=cpregb.
compute vcpregb2=cpregb2.
compute vcpregb3=cpregb3.
compute vcpregb4=cpregb4.
compute vcpregb5=cpregb5.
compute vcpregb6=cpregb6.
compute vcpregb7=cpregb7.
compute vcpregb8=cpregb8.

compute vmorprg1=morepreg.
compute vmorprg2=morepre2.
compute vmorprg3=morepre3.
compute vmorprg4=morepre4.
compute vmorprg5=morepre5.
compute vmorprg6=morepre6.
compute vmorprg7=morepre7.
compute vmorprg8=morepre8.

compute vpregk1=pregk.
compute vpregk2=pregk2.
compute vpregk3=pregk3.
compute vpregk4=pregk4.
compute vpregk5=pregk5.
compute vpregk6=pregk6.
compute vpregk7=pregk7.
compute vpregk8=pregk8.

compute vpregkw1=pregkw.
compute vpregkw2=pregkw2.
compute vpregkw3=pregkw3.
compute vpregkw4=pregkw4.
compute vpregkw5=pregkw5.
compute vpregkw6=pregkw6.
compute vpregkw7=pregkw7.
compute vpregkw8=pregkw8.

compute vpregl1=pregl.
compute vpregl2=pregl2.
```

```
compute vpregl3=pregl3.
compute vpregl4=pregl4.
compute vpregl5=pregl5.
compute vpregl6=pregl6.
compute vpregl7=pregl7.
compute vpregl8=pregl8.


compute vprega1=prega.
compute vprega2=prega2.
compute vprega3=prega3.
compute vprega4=prega4.
compute vprega5=prega5.
compute vprega6=prega6.
compute vprega7=prega7.
compute vprega8=prega8.
compute vprega9=prega9.
compute vprega10=prega10.
compute vprega11=prega11.
compute vprega12=prega12.
compute vprega13=prega13.
compute vprega14=prega14.
compute vprega15=prega15.
compute vprega16=prega16.
compute vprega17=prega17.
compute vprega18=prega18.
compute vprega19=prega19.
compute vprega20=prega20.
compute vprega21=prega21.
compute vprega22=prega22.
compute vprega23=prega23.
compute vprega24=prega24.
compute vprega25=prega25.
compute vprega26=prega26.
compute vprega27=prega27.
compute vprega28=prega28.
compute vprega29=prega29.
compute vprega30=prega30.
compute vprega31=prega31.
compute vprega32=prega32.
compute vprega33=prega33.
compute vprega34=prega34.
compute vprega35=prega35.
compute vprega36=prega36.
compute vprega37=prega37.
compute vprega38=prega38.
compute vprega39=prega39.
compute vprega40=prega40.


compute vpregc1=pregc.
compute vpregc2=pregc2.
compute vpregc3=pregc3.
compute vpregc4=pregc4.
compute vpregc5=pregc5.
compute vpregc6=pregc6.
compute vpregc7=pregc7.
compute vpregc8=pregc8.
compute vpregc9=pregc9.
compute vpregc10=pregc10.
```

```
compute vpregc11=pregc11.
compute vpregc12=pregc12.
compute vpregc13=pregc13.
compute vpregc14=pregc14.
compute vpregc15=pregc15.
compute vpregc16=pregc16.
compute vpregc17=pregc17.
compute vpregc18=pregc18.
compute vpregc19=pregc19.
compute vpregc20=pregc20.
compute vpregc21=pregc21.
compute vpregc22=pregc22.
compute vpregc23=pregc23.
compute vpregc24=pregc24.
compute vpregc25=pregc25.
compute vpregc26=pregc26.
compute vpregc27=pregc27.
compute vpregc28=pregc28.
compute vpregc29=pregc29.
compute vpregc30=pregc30.
compute vpregc31=pregc31.
compute vpregc32=pregc32.
compute vpregc33=pregc33.
compute vpregc34=pregc34.
compute vpregc35=pregc35.
compute vpregc36=pregc36.
compute vpregc37=pregc37.
compute vpregc38=pregc38.
compute vpregc39=pregc39.
compute vpregc40=pregc40.


compute vpregd1=pregd.
compute vpregd2=pregd2.
compute vpregd3=pregd3.
compute vpregd4=pregd4.
compute vpregd5=pregd5.
compute vpregd6=pregd6.
compute vpregd7=pregd7.
compute vpregd8=pregd8.
compute vpregd9=pregd9.
compute vpregd10=pregd10.
compute vpregd11=pregd11.
compute vpregd12=pregd12.
compute vpregd13=pregd13.
compute vpregd14=pregd14.
compute vpregd15=pregd15.
compute vpregd16=pregd16.
compute vpregd17=pregd17.
compute vpregd18=pregd18.
compute vpregd19=pregd19.
compute vpregd20=pregd20.
compute vpregd21=pregd21.
compute vpregd22=pregd22.
compute vpregd23=pregd23.
compute vpregd24=pregd24.
compute vpregd25=pregd25.
compute vpregd26=pregd26.
compute vpregd27=pregd27.
compute vpregd28=pregd28.
```

```
compute vpregd29=pregd29.
compute vpregd30=pregd30.
compute vpregd31=pregd31.
compute vpregd32=pregd32.
compute vpregd33=pregd33.
compute vpregd34=pregd34.
compute vpregd35=pregd35.
compute vpregd36=pregd36.
compute vpregd37=pregd37.
compute vpregd38=pregd38.
compute vpregd39=pregd39.
compute vpregd40=pregd40.


compute vkilo1=kilo.
compute vkilo2=kilo2.
compute vkilo3=kilo3.
compute vkilo4=kilo4.
compute vkilo5=kilo5.
compute vkilo6=kilo6.
compute vkilo7=kilo7.
compute vkilo8=kilo8.
compute vkilo9=kilo9.
compute vkilo10=kilo10.
compute vkilo11=kilo11.
compute vkilo12=kilo12.
compute vkilo13=kilo13.
compute vkilo14=kilo14.
compute vkilo15=kilo15.
compute vkilo16=kilo16.
compute vkilo17=kilo17.
compute vkilo18=kilo18.
compute vkilo19=kilo19.
compute vkilo20=kilo20.
compute vkilo21=kilo21.
compute vkilo22=kilo22.
compute vkilo23=kilo23.
compute vkilo24=kilo24.
compute vkilo25=kilo25.
compute vkilo26=kilo26.
compute vkilo27=kilo27.
compute vkilo28=kilo28.
compute vkilo29=kilo29.
compute vkilo30=kilo30.
compute vkilo31=kilo31.
compute vkilo32=kilo32.
compute vkilo33=kilo33.
compute vkilo34=kilo34.
compute vkilo35=kilo35.
compute vkilo36=kilo36.
compute vkilo37=kilo37.
compute vkilo38=kilo38.
compute vkilo39=kilo39.
compute vkilo40=kilo40.


compute vpregf1=pregf.
compute vpregf2=pregf2.
compute vpregf3=pregf3.
compute vpregf4=pregf4.
```

```
compute vpregf5=pregf5.
compute vpregf6=pregf6.
compute vpregf7=pregf7.
compute vpregf8=pregf8.
compute vpregf9=pregf9.
compute vpregf10=pregf10.
compute vpregf11=pregf11.
compute vpregf12=pregf12.
compute vpregf13=pregf13.
compute vpregf14=pregf14.
compute vpregf15=pregf15.
compute vpregf16=pregf16.
compute vpregf17=pregf17.
compute vpregf18=pregf18.
compute vpregf19=pregf19.
compute vpregf20=pregf20.
compute vpregf21=pregf21.
compute vpregf22=pregf22.
compute vpregf23=pregf23.
compute vpregf24=pregf24.
compute vpregf25=pregf25.
compute vpregf26=pregf26.
compute vpregf27=pregf27.
compute vpregf28=pregf28.
compute vpregf29=pregf29.
compute vpregf30=pregf30.
compute vpregf31=pregf31.
compute vpregf32=pregf32.
compute vpregf33=pregf33.
compute vpregf34=pregf34.
compute vpregf35=pregf35.
compute vpregf36=pregf36.
compute vpregf37=pregf37.
compute vpregf38=pregf38.
compute vpregf39=pregf39.
compute vpregf40=pregf40.


compute vpregh1=pregh.
compute vpregh2=pregh2.
compute vpregh3=pregh3.
compute vpregh4=pregh4.
compute vpregh5=pregh5.
compute vpregh6=pregh6.
compute vpregh7=pregh7.
compute vpregh8=pregh8.
compute vpregh9=pregh9.
compute vpregh10=pregh10.
compute vpregh11=pregh11.
compute vpregh12=pregh12.
compute vpregh13=pregh13.
compute vpregh14=pregh14.
compute vpregh15=pregh15.
compute vpregh16=pregh16.
compute vpregh17=pregh17.
compute vpregh18=pregh18.
compute vpregh19=pregh19.
compute vpregh20=pregh20.
compute vpregh21=pregh21.
compute vpregh22=pregh22.
```

```
compute vpregh23=pregh23.
compute vpregh24=pregh24.
compute vpregh25=pregh25.
compute vpregh26=pregh26.
compute vpregh27=pregh27.
compute vpregh28=pregh28.
compute vpregh29=pregh29.
compute vpregh30=pregh30.
compute vpregh31=pregh31.
compute vpregh32=pregh32.
compute vpregh33=pregh33.
compute vpregh34=pregh34.
compute vpregh35=pregh35.
compute vpregh36=pregh36.
compute vpregh37=pregh37.
compute vpregh38=pregh38.
compute vpregh39=pregh39.
compute vpregh40=pregh40.


compute vwhopa1=whopara.
compute vwhopa2=whopara2.
compute vwhopa3=whopara3.
compute vwhopa4=whopara4.
compute vwhopa5=whopara5.
compute vwhopa6=whopara6.
compute vwhopa7=whopara7.
compute vwhopa8=whopara8.
compute vwhopa9=whopara9.
compute vwhopa10=whopar10.
compute vwhopa11=whopar12.
compute vwhopa12=whopar14.
compute vwhopa13=whopar16.
compute vwhopa14=whopar18.
compute vwhopa15=whopar20.
compute vwhopa16=whopar22.
compute vwhopa17=whopar24.
compute vwhopa18=whopar26.
compute vwhopa19=whopar28.
compute vwhopa20=whopar30.
compute vwhopa21=whopar32.
compute vwhopa22=whopar34.
compute vwhopa23=whopar36.
compute vwhopa24=whopar38.
compute vwhopa25=whopar40.
compute vwhopa26=whopar42.
compute vwhopa27=whopar44.
compute vwhopa28=whopar46.
compute vwhopa29=whopar48.
compute vwhopa30=whopar50.
compute vwhopa31=whopar52.
compute vwhopa32=whopar54.
compute vwhopa33=whopar56.
compute vwhopa34=whopar58.
compute vwhopa35=whopar60.
compute vwhopa36=whopar62.
compute vwhopa37=whopar64.
compute vwhopa38=whopar66.
compute vwhopa39=whopar68.
compute vwhopa40=whopar70.
```

```
compute vwerkd1=wherkid.
compute vwerkd2=wherkid2.
compute vwerkd3=wherkid3.
compute vwerkd4=wherkid4.
compute vwerkd5=wherkid5.
compute vwerkd6=wherkid6.
compute vwerkd7=wherkid7.
compute vwerkd8=wherkid8.
compute vwerkd9=wherkid9.
compute vwerkd10=wherki10.
compute vwerkd11=wherki11.
compute vwerkd12=wherki12.
compute vwerkd13=wherki13.
compute vwerkd14=wherki14.
compute vwerkd15=wherki15.
compute vwerkd16=wherki16.
compute vwerkd17=wherki17.
compute vwerkd18=wherki18.
compute vwerkd19=wherki19.
compute vwerkd20=wherki20.
compute vwerkd21=wherki21.
compute vwerkd22=wherki22.
compute vwerkd23=wherki23.
compute vwerkd24=wherki24.
compute vwerkd25=wherki25.
compute vwerkd26=wherki26.
compute vwerkd27=wherki27.
compute vwerkd28=wherki28.
compute vwerkd29=wherki29.
compute vwerkd30=wherki30.
compute vwerkd31=wherki31.
compute vwerkd32=wherki32.
compute vwerkd33=wherki33.
compute vwerkd34=wherki34.
compute vwerkd35=wherki35.
compute vwerkd36=wherki36.
compute vwerkd37=wherki37.
compute vwerkd38=wherki38.
compute vwerkd39=wherki39.
compute vwerkd40=wherki40.


compute vwatkd1=whatkid.
compute vwatkd2=whatkid2.
compute vwatkd3=whatkid3.
compute vwatkd4=whatkid4.
compute vwatkd5=whatkid5.
compute vwatkd6=whatkid6.
compute vwatkd7=whatkid7.
compute vwatkd8=whatkid8.
compute vwatkd9=whatkid9.
compute vwatkd10=whatki10.
compute vwatkd11=whatki11.
compute vwatkd12=whatki12.
compute vwatkd13=whatki13.
compute vwatkd14=whatki14.
compute vwatkd15=whatki15.
compute vwatkd16=whatki16.
```

```
compute vwatkd17=whatki17.
compute vwatkd18=whatki18.
compute vwatkd19=whatki19.
compute vwatkd20=whatki20.
compute vwatkd21=whatki21.
compute vwatkd22=whatki22.
compute vwatkd23=whatki23.
compute vwatkd24=whatki24.
compute vwatkd25=whatki25.
compute vwatkd26=whatki26.
compute vwatkd27=whatki27.
compute vwatkd28=whatki28.
compute vwatkd29=whatki29.
compute vwatkd30=whatki30.
compute vwatkd31=whatki31.
compute vwatkd32=whatki32.
compute vwatkd33=whatki33.
compute vwatkd34=whatki34.
compute vwatkd35=whatki35.
compute vwatkd36=whatki36.
compute vwatkd37=whatki37.
compute vwatkd38=whatki38.
compute vwatkd39=whatki39.
compute vwatkd40=whatki40.

Compute vabspa1=abspara.
Compute vabspa2=abspara2.
Compute vabspa3=abspara3.
Compute vabspa4=abspara4.
Compute vabspa5=abspara5.
Compute vabspa6=abspara6.
Compute vabspa7=abspara7.
Compute vabspa8=abspara8.
Compute vabspa9=abspara9.
Compute vabspa10=abspar10.
Compute vabspa11=abspar13.
Compute vabspa12=abspar16.
Compute vabspa13=abspar19.
Compute vabspa14=abspar22.
Compute vabspa15=abspar25.
Compute vabspa16=abspar28.
Compute vabspa17=abspar31.
Compute vabspa18=abspar34.
Compute vabspa19=abspar37.
Compute vabspa20=abspar40.
Compute vabspa21=abspar43.
Compute vabspa22=abspar46.
Compute vabspa23=abspar49.
Compute vabspa24=abspar52.
Compute vabspa25=abspar55.
Compute vabspa26=abspar58.
Compute vabspa27=abspar61.
Compute vabspa28=abspar64.
Compute vabspa29=abspar67.
Compute vabspa30=abspar70.
Compute vabspa31=abspar73.
Compute vabspa32=abspar76.
Compute vabspa33=abspar79.
Compute vabspa34=abspar82.
Compute vabspa35=abspar85.
```

Compute vabspa36=abspar88.
Compute vabspa37=abspar91.
Compute vabspa38=abspar94.
Compute vabspa39=abspar97.
Compute vabspa40=abspa100.

Compute vabspb1=absparb.
Compute vabspb2=absparb2.
Compute vabspb3=absparb3.
Compute vabspb4=absparb4.
Compute vabspb5=absparb5.
Compute vabspb6=absparb6.
Compute vabspb7=absparb7.
Compute vabspb8=absparb8.
Compute vabspb9=absparb9.
Compute vabspb10=abspar11.
Compute vabspb11=abspar14.
Compute vabspb12=abspar17.
Compute vabspb13=abspar20.
Compute vabspb14=abspar23.
Compute vabspb15=abspar26.
Compute vabspb16=abspar29.
Compute vabspb17=abspar32.
Compute vabspb18=abspar35.
Compute vabspb19=abspar38.
Compute vabspb20=abspar41.
Compute vabspb21=abspar44.
Compute vabspb22=abspar47.
Compute vabspb23=abspar50.
Compute vabspb24=abspar53.
Compute vabspb25=abspar56.
Compute vabspb26=abspar59.
Compute vabspb27=abspar62.
Compute vabspb28=abspar65.
Compute vabspb29=abspar68.
Compute vabspb30=abspar71.
Compute vabspb31=abspar74.
Compute vabspb32=abspar77.
Compute vabspb33=abspar80.
Compute vabspb34=abspar83.
Compute vabspb35=abspar86.
Compute vabspb36=abspar89.
Compute vabspb37=abspar92.
Compute vabspb38=abspar95.
Compute vabspb39=abspar98.
Compute vabspb40=abspa101.

Compute vabspc1=absparc.
Compute vabspc2=absparc2.
Compute vabspc3=absparc3.
Compute vabspc4=absparc4.
Compute vabspc5=absparc5.
Compute vabspc6=absparc6.
Compute vabspc7=absparc7.
Compute vabspc8=absparc8.
Compute vabspc9=absparc9.
Compute vabspc10=abspar12.

```
Compute vabspc11=abspar15.
Compute vabspc12=abspar18.
Compute vabspc13=abspar21.
Compute vabspc14=abspar24.
Compute vabspc15=abspar27.
Compute vabspc16=abspar30.
Compute vabspc17=abspar33.
Compute vabspc18=abspar36.
Compute vabspc19=abspar39.
Compute vabspc20=abspar42.
Compute vabspc21=abspar45.
Compute vabspc22=abspar48.
Compute vabspc23=abspar51.
Compute vabspc24=abspar54.
Compute vabspc25=abspar57.
Compute vabspc26=abspar60.
Compute vabspc27=abspar63.
Compute vabspc28=abspar66.
Compute vabspc29=abspar69.
Compute vabspc30=abspar72.
Compute vabspc31=abspar75.
Compute vabspc32=abspar78.
Compute vabspc33=abspar81.
Compute vabspc34=abspar84.
Compute vabspc35=abspar87.
Compute vabspc36=abspar90.
Compute vabspc37=abspar93.
Compute vabspc38=abspar96.
Compute vabspc39=abspar99.
Compute vabspc40=abspa102.


Compute vabskb1=abskidb.
Compute vabskb2=abskidb2.
Compute vabskb3=abskidb3.
Compute vabskb4=abskidb4.
Compute vabskb5=abskidb5.
Compute vabskb6=abskidb6.
Compute vabskb7=abskidb7.
Compute vabskb8=abskidb8.
Compute vabskb9=abskidb9.
Compute vabskb10=abskid11.
Compute vabskb11=abskid16.
Compute vabskb12=abskid21.
Compute vabskb13=abskid26.
Compute vabskb14=abskid31.
Compute vabskb15=abskid36.
Compute vabskb16=abskid41.
Compute vabskb17=abskid46.
Compute vabskb18=abskid51.
Compute vabskb19=abskid56.
Compute vabskb20=abskid61.
Compute vabskb21=abskid66.
Compute vabskb22=abskid71.
Compute vabskb23=abskid76.
Compute vabskb24=abskid81.
Compute vabskb25=abskid86.
Compute vabskb26=abskid91.
Compute vabskb27=abskid96.
```

```
Compute vabskb28=abski101.
Compute vabskb29=abski106.
Compute vabskb30=abski111.
Compute vabskb31=abski116.
Compute vabskb32=abski121.
Compute vabskb33=abski126.
Compute vabskb34=abski131.
Compute vabskb35=abski136.
Compute vabskb36=abski141.
Compute vabskb37=abski146.
Compute vabskb38=abski151.
Compute vabskb39=abski156.
Compute vabskb40=abski161.


Compute vabskc1=abskidc.
Compute vabskc2=abskidc2.
Compute vabskc3=abskidc3.
Compute vabskc4=abskidc4.
Compute vabskc5=abskidc5.
Compute vabskc6=abskidc6.
Compute vabskc7=abskidc7.
Compute vabskc8=abskidc8.
Compute vabskc9=abskidc9.
Compute vabskc10=abskid12.
Compute vabskc11=abskid17.
Compute vabskc12=abskid22.
Compute vabskc13=abskid27.
Compute vabskc14=abskid32.
Compute vabskc15=abskid37.
Compute vabskc16=abskid42.
Compute vabskc17=abskid47.
Compute vabskc18=abskid52.
Compute vabskc19=abskid57.
Compute vabskc20=abskid62.
Compute vabskc21=abskid67.
Compute vabskc22=abskid72.
Compute vabskc23=abskid77.
Compute vabskc24=abskid82.
Compute vabskc25=abskid87.
Compute vabskc26=abskid92.
Compute vabskc27=abskid97.
Compute vabskc28=abski102.
Compute vabskc29=abski107.
Compute vabskc30=abski112.
Compute vabskc31=abski117.
Compute vabskc32=abski122.
Compute vabskc33=abski127.
Compute vabskc34=abski132.
Compute vabskc35=abski137.
Compute vabskc36=abski142.
Compute vabskc37=abski147.
Compute vabskc38=abski152.
Compute vabskc39=abski157.
Compute vabskc40=abski162.


Compute vabskd1=abskidd.
```

Compute vabskd2=abskidd2.
Compute vabskd3=abskidd3.
Compute vabskd4=abskidd4.
Compute vabskd5=abskidd5.
Compute vabskd6=abskidd6.
Compute vabskd7=abskidd7.
Compute vabskd8=abskidd8.
Compute vabskd9=abskidd9.
Compute vabskd10=abskid13.
Compute vabskd11=abskid18.
Compute vabskd12=abskid23.
Compute vabskd13=abskid28.
Compute vabskd14=abskid33.
Compute vabskd15=abskid38.
Compute vabskd16=abskid43.
Compute vabskd17=abskid48.
Compute vabskd18=abskid53.
Compute vabskd19=abskid58.
Compute vabskd20=abskid63.
Compute vabskd21=abskid68.
Compute vabskd22=abskid73.
Compute vabskd23=abskid78.
Compute vabskd24=abskid83.
Compute vabskd25=abskid88.
Compute vabskd26=abskid93.
Compute vabskd27=abskid98.
Compute vabskd28=abski103.
Compute vabskd29=abski108.
Compute vabskd30=abski113.
Compute vabskd31=abski118.
Compute vabskd32=abski123.
Compute vabskd33=abski128.
Compute vabskd34=abski133.
Compute vabskd35=abski138.
Compute vabskd36=abski143.
Compute vabskd37=abski148.
Compute vabskd38=abski153.
Compute vabskd39=abski158.
Compute vabskd40=abski163.

Compute vabske1=abskide.
Compute vabske2=abskide2.
Compute vabske3=abskide3.
Compute vabske4=abskide4.
Compute vabske5=abskide5.
Compute vabske6=abskide6.
Compute vabske7=abskide7.
Compute vabske8=abskide8.
Compute vabske9=abskide9.
Compute vabske10=abskid14.
Compute vabske11=abskid19.
Compute vabske12=abskid24.
Compute vabske13=abskid29.
Compute vabske14=abskid34.
Compute vabske15=abskid39.
Compute vabske16=abskid44.
Compute vabske17=abskid49.
Compute vabske18=abskid54.

```
Compute vabske19=abskid59.
Compute vabske20=abskid64.
Compute vabske21=abskid69.
Compute vabske22=abskid74.
Compute vabske23=abskid79.
Compute vabske24=abskid84.
Compute vabske25=abskid89.
Compute vabske26=abskid94.
Compute vabske27=abskid99.
Compute vabske28=abski104.
Compute vabske29=abski109.
Compute vabske30=abski114.
Compute vabske31=abski119.
Compute vabske32=abski124.
Compute vabske33=abski129.
Compute vabske34=abski134.
Compute vabske35=abski139.
Compute vabske36=abski144.
Compute vabske37=abski149.
Compute vabske38=abski154.
Compute vabske39=abski159.
Compute vabske40=abski164.


compute vpound1=pound.
compute vpound2=pound2.
compute vpound3=pound3.
compute vpound4=pound4.
compute vpound5=pound5.
compute vpound6=pound6.
compute vpound7=pound7.
compute vpound8=pound8.
compute vpound9=pound9.
compute vpound10=pound10.
compute vpound11=pound11.
compute vpound12=pound12.
compute vpound13=pound13.
compute vpound14=pound14.
compute vpound15=pound15.
compute vpound16=pound16.
compute vpound17=pound17.
compute vpound18=pound18.
compute vpound19=pound19.
compute vpound20=pound20.
compute vpound21=pound21.
compute vpound22=pound22.
compute vpound23=pound23.
compute vpound24=pound24.
compute vpound25=pound25.
compute vpound26=pound26.
compute vpound27=pound27.
compute vpound28=pound28.
compute vpound29=pound29.
compute vpound30=pound30.
compute vpound31=pound31.
compute vpound32=pound32.
compute vpound33=pound33.
compute vpound34=pound34.
compute vpound35=pound35.
compute vpound36=pound36.
```

```
compute vpound37=pound37.
compute vpound38=pound38.
compute vpound39=pound39.
compute vpound40=pound40.


compute vounce1=ounce.
compute vounce2=ounce2.
compute vounce3=ounce3.
compute vounce4=ounce4.
compute vounce5=ounce5.
compute vounce6=ounce6.
compute vounce7=ounce7.
compute vounce8=ounce8.
compute vounce9=ounce9.
compute vounce10=ounce10.
compute vounce11=ounce11.
compute vounce12=ounce12.
compute vounce13=ounce13.
compute vounce14=ounce14.
compute vounce15=ounce15.
compute vounce16=ounce16.
compute vounce17=ounce17.
compute vounce18=ounce18.
compute vounce19=ounce19.
compute vounce20=ounce20.
compute vounce21=ounce21.
compute vounce22=ounce22.
compute vounce23=ounce23.
compute vounce24=ounce24.
compute vounce25=ounce25.
compute vounce26=ounce26.
compute vounce27=ounce27.
compute vounce28=ounce28.
compute vounce29=ounce29.
compute vounce30=ounce30.
compute vounce31=ounce31.
compute vounce32=ounce32.
compute vounce33=ounce33.
compute vounce34=ounce34.
compute vounce35=ounce35.
compute vounce36=ounce36.
compute vounce37=ounce37.
compute vounce38=ounce38.
compute vounce39=ounce39.
compute vounce40=ounce40.


compute vpregg1=pregg.
compute vpregg2=pregg2.
compute vpregg3=pregg3.
compute vpregg4=pregg4.
compute vpregg5=pregg5.
compute vpregg6=pregg6.
compute vpregg7=pregg7.
compute vpregg8=pregg8.
compute vpregg9=pregg9.
compute vpregg10=pregg10.
compute vpregg11=pregg11.
compute vpregg12=pregg12.
```

```
compute vpregg13=pregg13.
compute vpregg14=pregg14.
compute vpregg15=pregg15.
compute vpregg16=pregg16.
compute vpregg17=pregg17.
compute vpregg18=pregg18.
compute vpregg19=pregg19.
compute vpregg20=pregg20.
compute vpregg21=pregg21.
compute vpregg22=pregg22.
compute vpregg23=pregg23.
compute vpregg24=pregg24.
compute vpregg25=pregg25.
compute vpregg26=pregg26.
compute vpregg27=pregg27.
compute vpregg28=pregg28.
compute vpregg29=pregg29.
compute vpregg30=pregg30.
compute vpregg31=pregg31.
compute vpregg32=pregg32.
compute vpregg33=pregg33.
compute vpregg34=pregg34.
compute vpregg35=pregg35.
compute vpregg36=pregg36.
compute vpregg37=pregg37.
compute vpregg38=pregg38.
compute vpregg39=pregg39.
compute vpregg40=pregg40.


compute vwhopb1=whoparb.
compute vwhopb2=whoparb2.
compute vwhopb3=whoparb3.
compute vwhopb4=whoparb4.
compute vwhopb5=whoparb5.
compute vwhopb6=whoparb6.
compute vwhopb7=whoparb7.
compute vwhopb8=whoparb8.
compute vwhopb9=whoparb9.
compute vwhopb10=whopar11.
compute vwhopb11=whopar13.
compute vwhopb12=whopar15.
compute vwhopb13=whopar17.
compute vwhopb14=whopar19.
compute vwhopb15=whopar21.
compute vwhopb16=whopar23.
compute vwhopb17=whopar25.
compute vwhopb18=whopar27.
compute vwhopb19=whopar29.
compute vwhopb20=whopar31.
compute vwhopb21=whopar33.
compute vwhopb22=whopar35.
compute vwhopb23=whopar37.
compute vwhopb24=whopar39.
compute vwhopb25=whopar41.
compute vwhopb26=whopar43.
compute vwhopb27=whopar45.
compute vwhopb28=whopar47.
compute vwhopb29=whopar49.
compute vwhopb30=whopar51.
```

compute vwhopb31=whopar53.
compute vwhopb32=whopar55.
compute vwhopb33=whopar57.
compute vwhopb34=whopar59.
compute vwhopb35=whopar61.
compute vwhopb36=whopar63.
compute vwhopb37=whopar65.
compute vwhopb38=whopar67.
compute vwhopb39=whopar69.
compute vwhopb40=whopar71.

Compute vabska1=abskida.
Compute vabska2=abskida2.
Compute vabska3=abskida3.
Compute vabska4=abskida4.
Compute vabska5=abskida5.
Compute vabska6=abskida6.
Compute vabska7=abskida7.
Compute vabska8=abskida8.
Compute vabska9=abskida9.
Compute vabska10=abskid10.
Compute vabska11=abskid15.
Compute vabska12=abskid20.
Compute vabska13=abskid25.
Compute vabska14=abskid30.
Compute vabska15=abskid35.
Compute vabska16=abskid40.
Compute vabska17=abskid45.
Compute vabska18=abskid50.
Compute vabska19=abskid55.
Compute vabska20=abskid60.
Compute vabska21=abskid65.
Compute vabska22=abskid70.
Compute vabska23=abskid75.
Compute vabska24=abskid80.
Compute vabska25=abskid85.
Compute vabska26=abskid90.
Compute vabska27=abskid95.
Compute vabska28=abski100.
Compute vabska29=abski105.
Compute vabska30=abski110.
Compute vabska31=abski115.
Compute vabska32=abski120.
Compute vabska33=abski125.
Compute vabska34=abski130.
Compute vabska35=abski135.
Compute vabska36=abski140.
Compute vabska37=abski145.
Compute vabska38=abski150.
Compute vabska39=abski155.
Compute vabska40=abski160.

Compute vabsmn1=absmon.
Compute vabsmn2=absmon2.
Compute vabsmn3=absmon3.
Compute vabsmn4=absmon4.

```
Compute vabsmn5=absmon5.
Compute vabsmn6=absmon6.
Compute vabsmn7=absmon7.
Compute vabsmn8=absmon8.
Compute vabsmn9=absmon9.
Compute vabsmn10=absmon10.
Compute vabsmn11=absmon11.
Compute vabsmn12=absmon12.
Compute vabsmn13=absmon13.
Compute vabsmn14=absmon14.
Compute vabsmn15=absmon15.
Compute vabsmn16=absmon16.
Compute vabsmn17=absmon17.
Compute vabsmn18=absmon18.
Compute vabsmn19=absmon19.
Compute vabsmn20=absmon20.
Compute vabsmn21=absmon21.
Compute vabsmn22=absmon22.
Compute vabsmn23=absmon23.
Compute vabsmn24=absmon24.
Compute vabsmn25=absmon25.
Compute vabsmn26=absmon26.
Compute vabsmn27=absmon27.
Compute vabsmn28=absmon28.
Compute vabsmn29=absmon29.
Compute vabsmn30=absmon30.
Compute vabsmn31=absmon31.
Compute vabsmn32=absmon32.
Compute vabsmn33=absmon33.
Compute vabsmn34=absmon34.
Compute vabsmn35=absmon35.
Compute vabsmn36=absmon36.
Compute vabsmn37=absmon37.
Compute vabsmn38=absmon38.
Compute vabsmn39=absmon39.
Compute vabsmn40=absmon40.


compute vgramm1=gramm.
compute vgramm2=gramm2.
compute vgramm3=gramm3.
compute vgramm4=gramm4.
compute vgramm5=gramm5.
compute vgramm6=gramm6.

compute vgramm7=gramm7.
compute vgramm8=gramm8.
compute vgramm9=gramm9.
compute vgramm10=gramm10.
compute vgramm11=gramm11.
compute vgramm12=gramm12.
compute vgramm13=gramm13.
compute vgramm14=gramm14.
compute vgramm15=gramm15.
compute vgramm16=gramm16.
compute vgramm17=gramm17.
compute vgramm18=gramm18.
compute vgramm19=gramm19.
compute vgramm20=gramm20.
compute vgramm21=gramm21.
```

```
compute vgramm22=gramm22.
compute vgramm23=gramm23.
compute vgramm24=gramm24.
compute vgramm25=gramm25.
compute vgramm26=gramm26.
compute vgramm27=gramm27.
compute vgramm28=gramm28.
compute vgramm29=gramm29.
compute vgramm30=gramm30.
compute vgramm31=gramm31.
compute vgramm32=gramm32.
compute vgramm33=gramm33.
compute vgramm34=gramm34.
compute vgramm35=gramm35.
compute vgramm36=gramm36.
compute vgramm37=gramm37.
compute vgramm38=gramm38.
compute vgramm39=gramm39.
compute vgramm40=gramm40.


Compute vabsyr1=absyr.
Compute vabsyr2=absyr2.
Compute vabsyr3=absyr3.
Compute vabsyr4=absyr4.
Compute vabsyr5=absyr5.
Compute vabsyr6=absyr6.
Compute vabsyr7=absyr7.
Compute vabsyr8=absyr8.
Compute vabsyr9=absyr9.
Compute vabsyr10=absyr10.
Compute vabsyr11=absyr11.
Compute vabsyr12=absyr12.
Compute vabsyr13=absyr13.
Compute vabsyr14=absyr14.
Compute vabsyr15=absyr15.
Compute vabsyr16=absyr16.
Compute vabsyr17=absyr17.
Compute vabsyr18=absyr18.
Compute vabsyr19=absyr19.
Compute vabsyr20=absyr20.
Compute vabsyr21=absyr21.
Compute vabsyr22=absyr22.
Compute vabsyr23=absyr23.
Compute vabsyr24=absyr24.
Compute vabsyr25=absyr25.
Compute vabsyr26=absyr26.
Compute vabsyr27=absyr27.
Compute vabsyr28=absyr28.
Compute vabsyr29=absyr29.
Compute vabsyr30=absyr30.
Compute vabsyr31=absyr31.
Compute vabsyr32=absyr32.
Compute vabsyr33=absyr33.
Compute vabsyr34=absyr34.
Compute vabsyr35=absyr35.
Compute vabsyr36=absyr36.
Compute vabsyr37=absyr37.
Compute vabsyr38=absyr38.
Compute vabsyr39=absyr39.
```

Compute vabsyr40=absyr40.

Compute vprege1=prege.
Compute vprege2=prege2.
Compute vprege3=prege3.
Compute vprege4=prege4.
Compute vprege5=prege5.
Compute vprege6=prege6.
Compute vprege7=prege7.
Compute vprege8=prege8.
Compute vprege9=prege9.
Compute vprege10=prege10.
Compute vprege11=prege11.
Compute vprege12=prege12.
Compute vprege13=prege13.
Compute vprege14=prege14.
Compute vprege15=prege15.
Compute vprege16=prege16.
Compute vprege17=prege17.
Compute vprege18=prege18.
Compute vprege19=prege19.
Compute vprege20=prege20.
Compute vprege21=prege21.
Compute vprege22=prege22.
Compute vprege23=prege23.
Compute vprege24=prege24.
Compute vprege25=prege25.
Compute vprege26=prege26.
Compute vprege27=prege27.
Compute vprege28=prege28.
Compute vprege29=prege29.
Compute vprege30=prege30.
Compute vprege31=prege31.
Compute vprege32=prege32.
Compute vprege33=prege33.
Compute vprege34=prege34.
Compute vprege35=prege35.
Compute vprege36=prege36.
Compute vprege37=prege37.
Compute vprege38=prege38.
Compute vprege39=prege39.
Compute vprege40=prege40.

Compute vpregi1=pregi.
Compute vpregi2=pregi2.
Compute vpregi3=pregi3.
Compute vpregi4=pregi4.
Compute vpregi5=pregi5.
Compute vpregi6=pregi6.
Compute vpregi7=pregi7.
Compute vpregi8=pregi8.
Compute vpregi9=pregi9.
Compute vpregi10=pregi10.
Compute vpregi11=pregi11.
Compute vpregi12=pregi12.
Compute vpregi13=pregi13.
Compute vpregi14=pregi14.
Compute vpregi15=pregi15.

```
Compute vpregi16=pregi16.
Compute vpregi17=pregi17.
Compute vpregi18=pregi18.
Compute vpregi19=pregi19.
Compute vpregi20=pregi20.
Compute vpregi21=pregi21.
Compute vpregi22=pregi22.
Compute vpregi23=pregi23.
Compute vpregi24=pregi24.
Compute vpregi25=pregi25.
Compute vpregi26=pregi26.
Compute vpregi27=pregi27.
Compute vpregi28=pregi28.
Compute vpregi29=pregi29.
Compute vpregi30=pregi30.
Compute vpregi31=pregi31.
Compute vpregi32=pregi32.
Compute vpregi33=pregi33.
Compute vpregi34=pregi34.
Compute vpregi35=pregi35.
Compute vpregi36=pregi36.
Compute vpregi37=pregi37.
Compute vpregi38=pregi38.
Compute vpregi39=pregi39.
Compute vpregi40=pregi40.
```

* Having copied all the data into manageable v-prefix vectors, use loop structure to copy them to their correct
* position (x-prefix vectors), then copy them back to their ordinary names from the x-prefix vectors.
* As we can be sure at this stage we are only interested in single-baby pregnancies, the
* destination slots are numbered 1,6,11,16,21,26,31 and 36 (i.e. 5*#i-4).

* Note this process only occurs for cases where we know the outcomes have been wrongly bunched in the same pregnancy,
* or where the outcomes are in the wrong chronological order, or (in the case of NCDS) where a birth event prior to March 1991 is
* recorded, and we know the member was interviewed at that last sweep in 1991 . Even in the above cases, we don't proceed if there are any
* outcomes with a year entered as '9999', or where we know there are any true multiple outcomes, or any dubious multiple
* outcomes (i.e. >3 days but less than 9 months apart).   The only exceptions are 26 'dubious' cases referred to specifically by
* serial number, where manual scrutiny revealed they contain no true multiple outcomes, and have no '9999' years, plus a further
* 12 cases where outcomes were not out of chronological order, but where slot 1 wasn't filled in (i.e. data started at slot 6, then
* slot 11,... etc.) So all data had to be moved 'back' by one pregnancy.

* Initiate variable 'changed', which flags up all cases that needed any kind of alteration (automatic or manual). If 'changed'
* remains zero, no need to overwrite existing variables with values of x-prefix vectors (which in any case, won't have been defined).

Compute changed=0.

Do if (((bunched=1 or ordrflag=1 or pre91flg=1) and (noyrflag=0 and truemult=0 and oddmult=0)) or
(nserial='055086Z' or nserial='084004J' or nserial='110304M' or nserial='223020B' or

nserial='235018J' or nserial='288027J' or nserial='310017Y' or nserial='310107Z' or nserial='385025V' or
nserial='500375L' or nserial='510127W' or nserial='513053K' or nserial='524002C' or nserial='950123U' or nserial='950154F' or nserial='985067J' or
nserial='X32084Z' or nserial='X82287X' or
nserial='Y20154L') or
(bserial='01005054' or bserial='01445068' or bserial='01840000' or bserial='01859054' or bserial='01949055' or bserial='03121039'
or bserial='03349048' or bserial='03579031'
or bserial='03997042' or bserial='04046069' or bserial='05548084' or bserial='05661087' or bserial='06173005'
or bserial='06226027' or bserial='06880097' or bserial='08239086' or bserial='08388069' or bserial='08716094' or bserial='10455045'
or bserial='11539049' or bserial='11871084' or bserial='12263073' or bserial='13045095' or bserial='14978022' or bserial='15810064'
or bserial='16293014' or bserial='20503000'
or bserial='01001053' or bserial='10788031' or bserial='14101044' or bserial='14574012' or bserial='15013072' or bserial='17015079'
or bserial='20432000' or bserial='02182094' or bserial='03257097' or bserial='03310042' or bserial='05984097' or bserial='09763027')).

Compute changed=1.

loop #i=1 to 8.
Do if(key(#i) ne 0).
compute xprega(5*#i-4)=vprega(key(#i)).
compute xpregc(5*#i-4)=vpregc(key(#i)).
compute xpregd(5*#i-4)=vpregd(key(#i)).
compute xkilo(5*#i-4)=vkilo(key(#i)).
compute xpregf(5*#i-4)=vpregf(key(#i)).
compute xpregh(5*#i-4)=vpregh(key(#i)).
compute xwhopa(5*#i-4)=vwhopa(key(#i)).
compute xwerkd(5*#i-4)=vwerkd(key(#i)).
compute xwatkd(5*#i-4)=vwatkd(key(#i)).
compute xabspa(5*#i-4)=vabspa(key(#i)).
compute xabspb(5*#i-4)=vabspb(key(#i)).
compute xabspc(5*#i-4)=vabspc(key(#i)).
compute xabskb(5*#i-4)=vabskb(key(#i)).
compute xabskc(5*#i-4)=vabskc(key(#i)).
compute xabskd(5*#i-4)=vabskd(key(#i)).
compute xabske(5*#i-4)=vabske(key(#i)).
compute xpound(5*#i-4)=vpound(key(#i)).
compute xounce(5*#i-4)=vounce(key(#i)).
compute xprged(5*#i-4)=vprged(key(#i)).
compute xprgem(5*#i-4)=vprgem(key(#i)).
compute xpregg(5*#i-4)=vpregg(key(#i)).
compute xwhopb(5*#i-4)=vwhopb(key(#i)).
compute xabska(5*#i-4)=vabska(key(#i)).
compute xabsmn(5*#i-4)=vabsmn(key(#i)).
compute xgramm(5*#i-4)=vgramm(key(#i)).
compute xprgey(5*#i-4)=vprgey(key(#i)).
compute xabsyr(5*#i-4)=vabsyr(key(#i)).
compute xprege(5*#i-4)=vprege(key(#i)).
compute xpregi(5*#i-4)=vpregi(key(#i)).

* set 'pregnum' for the new pregnancy slot (1-8) associated with the new position for this
* baby, to exactly 1, as we know it's not a multiple birth (i.e. we know truemult=0).
* In the case of NCDS pre-March 91 birth events, though, we want 'pregnum' to be zero.

compute xpregnu(#i)=1.

* In the case of NCDS pre-March 91 birth events, though, we want 'pregnum' to be zero,
* as we're deleting all reference to that event.

If (pre91flg=1 and key(#i)=40)xpregnu(#i)=0.

end if.
end loop.

* Copy the 'pregnancy' variables associated with that baby to the correct pregnancy slot (1-8).
* Note that within the 3-loop structure (#i,#j,#k) the copying is independent of the value of #k.
* i.e. anything up to five babies wrongly bunched in one pregnancy will have identical
* pregnancy data copied to different new pregnancy slots.

* Note also that, in the case of NCDS pre-March 91 birth events, the data for the 'eighth' pregnancy
* (i.e. that corresponding to the 40th birth event) will be copied, and in all such NCDS cases we know this is blank.

loop #i=1 to 8.
loop #j=1 to 8.
loop #k=1 to 5.
Do if (key(#i)=(5*#j+#k-5)).
compute xpregj(#i)=vpregj(#j).
compute xcg1prg(#i)=vcg1prg(#j).
compute xcg2prg(#i)=vcg2prg(#j).
compute xcg3prg(#i)=vcg3prg(#j).
compute xcpregb(#i)=vcpregb(#j).
compute xpregk(#i)=vpregk(#j).
compute xpregkw(#i)=vpregkw(#j).
compute xpregl(#i)=vpregl(#j).
end if.
end loop.
end loop.
end loop.

end if.

* Now address every single case that needed a manual alteration.

Do if (nserial='083034N').
Compute changed=2.
Compute key40(1)=21.
Compute key40(6)=11.
Compute key40(7)=16.
Compute key40(11)=6.
Compute key40(16)=1.
Compute keypreg(1)=5.
Compute keypreg(2)=3.
Compute keypreg(3)=2.
Compute keypreg(4)=1.
End if.

* The following cases all had exclusively pre-March 1991 birth event data, and were
* interviewed at NCDS5, so delete all birthslot information (i.e. make changed=2, but no key values).

Do if (nserial='094029H' or nserial='330027Q' or nserial='986039J').

```
Compute changed =2.
End if.

Do if (nserial='093273P').
Compute changed=2.
Compute key40(1)=1.
Compute keypreg(1)=1.
End if.

Do if (nserial='095023Z' or nserial='181038T' or nserial='960072L').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=1.
Compute keypreg(1)=1.
End if.

Do if (nserial='095006Z' or nserial='509228V' or nserial='950288D' or nserial='960029K').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute keypreg(1)=1.
End if.

Do if (nserial='120132Q').
Compute changed=2.
Compute key40(1)=6.
Compute key40(6)=1.
Compute key40(7)=2.
Compute keypreg(1)=2.
Compute keypreg(2)=1.
End if.

Do if (nserial='233008U' or nserial='285020B' or nserial='500353Z' or nserial='527001S' or
nserial='565063F' or nserial='650025W').
Compute changed=2.
Compute key40(1)=1.
Compute keypreg(1)=1.
End if.

Do if (nserial='280041J').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute key40(16)=16.
Compute key40(21)=21.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
Compute keypreg(4)=4.
Compute keypreg(5)=5.
End if.


Do if (nserial='280053R').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=16.
```

```
Compute key40(12)=11.
Compute key40(16)=21.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=4.
Compute keypreg(4)=5.
End if.

Do if (nserial='289141M' or nserial='382004V' or nserial='421030K' or nserial='529009V' or
nserial='X25050R').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=2.
Compute keypreg(1)=1.
End if.

Do if (nserial='515085K').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=7.
Compute key40(7)=6.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.

Do if (nserial='581068J').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute key40(16)=16.
Compute key40(21)=21.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
Compute keypreg(4)=4.
Compute keypreg(5)=5.
End if.

Do if (nserial='591050U').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute key40(6)=11.
Compute key40(11)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=3.
Compute keypreg(3)=4.
End if.

Do if (nserial='620037K').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute key40(12)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
End if.
```

```
Do if (nserial='630002V').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute key40(16)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
Compute keypreg(4)=4.
End if.

Do if (nserial='730049F').
Compute changed=2.
Compute key40(1)=6.
Compute key40(6)=1.
Compute key40(11)=11.
Compute key40(16)=16.
Compute keypreg(1)=2.
Compute keypreg(2)=1.
Compute keypreg(3)=3.
Compute keypreg(4)=4.
End if.

Do if (nserial='730130N').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(7)=7.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.

Do if (nserial='987100P').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=11.
Compute key40(6)=6.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.

Do if (nserial='989011A').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=7.
Compute key40(7)=6.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.

Do if (nserial='X25101F').
Compute changed=2.
Compute key40(1)=3.
Compute key40(2)=2.
Compute key40(6)=1.
Compute keypreg(1)=1.
Compute keypreg(6)=1.
End if.
```

```
Do if (nserial='X37008K').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=2.
Compute key40(6)=6.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.

Do if (nserial='X38005J').
Compute changed=2.
Compute key40(1)=6.
Compute key40(6)=11.
Compute keypreg(1)=2.
Compute keypreg(2)=3.
End if.

Do if (nserial='X82344H').
Compute changed=2.
Compute key40(1)=6.
Compute key40(2)=1.
Compute keypreg(1)=1.
End if.

Do if (nserial='Y21037M').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute key40(16)=16.
Compute key40(21)=21.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
Compute keypreg(4)=4.
Compute keypreg(5)=5.
End if.

Do if (nserial='041034A').
Compute changed=2.
Compute key40(1)=3.
Compute key40(6)=2.
Compute key40(11)=1.
Compute keypreg(1)=1.
Compute keypreg(2)=1.
Compute keypreg(3)=1.
End if.


Do if (nserial='130007T').
Compute changed=2.
Compute key40(1)=11.
Compute key40(2)=12.
Compute key40(6)=6.
Compute key40(11)=1.
Compute keypreg(1)=3.
Compute keypreg(2)=2.
Compute keypreg(3)=1.
End if.
```

```
Do if (nserial='283003R').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=3.
Compute key40(6)=6.
Compute key40(7)=7.
Compute key40(8)=8.
Compute key40(11)=11.
Compute key40(16)=1.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
Compute keypreg(4)=1.
End if.

Do if (nserial='422073K').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.

Do if (nserial='424029S').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute key40(16)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
Compute keypreg(4)=4.
End if.

Do if (nserial='511159R').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=2.
Compute keypreg(1)=1.
Compute keypreg(2)=1.
End if.

Do if (nserial='561006W').
Compute changed=2.
Compute key40(1)=11.
Compute key40(6)=6.
Compute key40(11)=1.
Compute keypreg(1)=3.
Compute keypreg(2)=2.
Compute keypreg(3)=1.
End if.

Do if (nserial='782013B').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=2.
Compute keypreg(1)=1.
```

```
Compute keypreg(2)=1.
End if.

Do if (nserial='Y33089Z').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=11.
Compute key40(11)=6.
Compute key40(16)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=3.
Compute keypreg(3)=2.
Compute keypreg(4)=4.
End if.

Do if (nserial='043013C').
Compute changed=2.
Compute key40(1)=6.
Compute key40(2)=7.
Compute key40(6)=1.
Compute keypreg(1)=2.
Compute keypreg(2)=1.
End if.


Do if (nserial='044001A').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=16.
Compute key40(11)=21.
Compute keypreg(1)=1.
Compute keypreg(2)=4.
Compute keypreg(3)=5.
End if.


Do if (nserial='183016T').
Compute changed=2.
Compute key40(1)=3.
Compute keypreg(1)=1.
End if.


Do if (nserial='188030P').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=3.
Compute key40(3)=1.
Compute keypreg(1)=1.
End if.


Do if (nserial='500015C').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
```

```
Compute keypreg(3)=3.
End if.


Do if (nserial='500219V').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute key40(12)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
End if.


Do if (nserial='509254W').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=1.
Compute keypreg(1)=1.
End if.


Do if (nserial='730084J').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(7)=7.
Compute key40(11)=11.
Compute key40(16)=21.
Compute key40(21)=16.
Compute key40(26)=26.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
Compute keypreg(4)=5.
Compute keypreg(5)=4.
Compute keypreg(6)=6.
End if.


Do if (nserial='950290Q').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=3.
Compute key40(3)=6.
Compute key40(4)=7.
Compute key40(5)=1.
Compute key40(6)=12.
Compute key40(7)=11.
Compute keypreg(1)=1.
Compute keypreg(2)=3.
End if.


Do if (nserial='980004E').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=3.
```

```
Compute key40(3)=1.
Compute keypreg(1)=1.
End if.


Do if (nserial='986100J').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=3.
Compute key40(6)=1.
Compute keypreg(1)=1.
Compute keypreg(6)=1.
End if.


Do if (nserial='X38008Q').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(7)=11.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.


Do if (nserial='Y00174D').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute key40(16)=16.
Compute key40(17)=21.
Compute key40(21)=26.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
Compute keypreg(4)=4.
Compute keypreg(5)=6.
End if.


Do if (bserial='05486061').
Compute changed=2.
Compute key40(1)=6.
Compute key40(6)=1.
Compute key40(11)=12.
Compute key40(12)=11.
Compute key40(16)=16.
Compute key40(21)=21.
Compute keypreg(1)=2.
Compute keypreg(2)=1.
Compute keypreg(3)=3.
Compute keypreg(4)=4.
Compute keypreg(5)=5.
End if.


Do if (bserial='08052057').
Compute changed=2.
Compute key40(1)=7.
```

```
Compute key40(2)=6.
Compute key40(6)=11.
Compute keypreg(1)=2.
Compute keypreg(2)=3.
End if.


Do if (bserial='09362043' or bserial='09541070').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=16.
Compute key40(16)=11.
Compute key40(21)=21.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=4.
Compute keypreg(4)=3.
Compute keypreg(5)=5.
End if.


Do if (bserial='00279065').
Compute changed=2.
Compute key40(1)=6.
Compute key40(6)=2.
Compute key40(11)=1.
Compute keypreg(1)=2.
Compute keypreg(2)=1.
Compute keypreg(3)=1.
End if.

Do if (bserial='05794018').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=2.
Compute key40(11)=6.
Compute keypreg(1)=1.
Compute keypreg(2)=1.
Compute keypreg(3)=2.
End if.

Do if (bserial='07020000').
Compute changed=2.
Compute key40(1)=2.
Compute key40(6)=1.
Compute keypreg(1)=1.
Compute keypreg(2)=1.
End if.


Do if (bserial='08531015').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=11.
Compute keypreg(1)=1.
Compute keypreg(2)=3.
End if.
```

```
Do if (bserial='09047035').
Compute changed=2.
Compute key40(1)=11.
Compute key40(6)=6.
Compute key40(11)=1.
Compute keypreg(1)=3.
Compute keypreg(2)=2.
Compute keypreg(3)=1.
End if.


Do if (bserial='13799017').
Compute changed=2.
Compute key40(1)=6.
Compute key40(6)=1.
Compute key40(11)=11.
Compute key40(16)=16.
Compute key40(21)=21.
Compute keypreg(1)=2.
Compute keypreg(2)=1.
Compute keypreg(3)=3.
Compute keypreg(4)=4.
Compute keypreg(5)=5.
End if.


Do if (bserial='00508088').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=11.
Compute key40(3)=6.
Compute key40(6)=16.
Compute key40(11)=21.
Compute key40(16)=26.
Compute keypreg(1)=1.
Compute keypreg(2)=4.
Compute keypreg(3)=5.
Compute keypreg(4)=6.
End if.

Do if (bserial='03279000').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=21.
Compute key40(11)=26.
Compute key40(16)=6.
Compute key40(21)=11.
Compute key40(26)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=5.
Compute keypreg(3)=6.
Compute keypreg(4)=2.
Compute keypreg(5)=3.
Compute keypreg(6)=4.
End if.

Do if (bserial='06196084').
Compute changed=2.
Compute key40(1)=6.
Compute key40(6)=1.
```

```
Compute key40(7)=11.
Compute keypreg(1)=2.
Compute keypreg(2)=1.
End if.


Do if (bserial='07342034').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(7)=7.
Compute key40(11)=26.
Compute key40(12)=27.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=6.
End if.


Do if (bserial='08039031').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=2.
Compute key40(3)=16.
Compute key40(6)=6.
Compute key40(11)=11.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
End if.


Do if (bserial='10231037').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute key40(6)=11.
Compute key40(11)=16.
Compute key40(16)=21.
Compute key40(17)=22.
Compute keypreg(1)=1.
Compute keypreg(2)=3.
Compute keypreg(3)=4.
Compute keypreg(4)=5.
End if.


Do if (bserial='10822000').
Compute changed=2.
Compute key40(1)=6.
Compute key40(6)=1.
Compute key40(7)=11.
Compute keypreg(1)=2.
Compute keypreg(2)=1.
End if.

Do if (bserial='02706048').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
```

```
Compute keypreg(1)=1.
End if.

Do if (bserial='03403069').
Compute changed=2.
Compute key40(1)=16.
Compute key40(2)=21.
Compute key40(3)=26.
Compute key40(6)=1.
Compute key40(11)=6.
Compute key40(16)=11.
Compute keypreg(1)=4.
Compute keypreg(2)=1.
Compute keypreg(3)=2.
Compute keypreg(4)=3.
End if.


Do if (bserial='03783010').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=11.
Compute key40(6)=6.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.


Do if (bserial='03947061').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute keypreg(1)=1.
End if.


Do if (bserial='05521004').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=2.
Compute key40(7)=3.
Compute keypreg(1)=1.
Compute keypreg(2)=1.
End if.

Do if (bserial='05671063').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=11.
Compute key40(6)=6.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.

Do if (bserial='06548062').
Compute changed=2.
Compute key40(1)=21.
Compute key40(6)=1.
Compute key40(11)=6.
Compute key40(12)=7.
```

```
Compute key40(16)=11.
Compute key40(21)=16.
Compute key40(26)=26.
Compute keypreg(1)=5.
Compute keypreg(2)=1.
Compute keypreg(3)=2.
Compute keypreg(4)=3.
Compute keypreg(5)=4.
Compute keypreg(6)=6.
End if.


Do if (bserial='06982024').
Compute changed=2.
Compute key40(1)=6.
Compute key40(2)=7.
Compute key40(6)=1.
Compute keypreg(1)=2.
Compute keypreg(2)=1.
End if.


Do if (bserial='07149006').
Compute changed=2.
Compute key40(1)=11.
Compute key40(2)=12.
Compute key40(6)=1.
Compute key40(11)=6.
Compute keypreg(1)=3.
Compute keypreg(2)=1.
Compute keypreg(3)=2.
End if.


Do if (bserial='07339085').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=11.
Compute key40(7)=12.
Compute key40(11)=6.
Compute keypreg(1)=1.
Compute keypreg(2)=3.
Compute keypreg(3)=2.
End if.


Do if (bserial='08995083').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=2.
Compute key40(6)=3.
Compute key40(11)=4.
Compute keypreg(1)=1.
Compute keypreg(2)=1.
Compute keypreg(3)=1.
End if.


Do if (bserial='11155092').
Compute changed=2.
```

```
Compute key40(1)=1.
Compute key40(2)=2.
Compute keypreg(1)=1.
End if.


Do if (bserial='12150094').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=11.
Compute key40(11)=16.
Compute key40(12)=17.
Compute key40(16)=6.
Compute keypreg(1)=1.
Compute keypreg(2)=3.
Compute keypreg(3)=4.
Compute keypreg(4)=2.
End if.


Do if (bserial='12693010').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute key40(12)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
End if.


Do if (bserial='12958013').
Compute changed=2.
Compute key40(1)=3.
Compute key40(2)=4.
Compute key40(6)=2.
Compute key40(11)=1.
Compute keypreg(1)=1.
Compute keypreg(2)=1.
Compute keypreg(3)=1.
End if.

Do if (bserial='14046048').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=16.
Compute key40(16)=21.
Compute key40(21)=26.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=4.
Compute keypreg(4)=5.
Compute keypreg(5)=6.
End if.

Do if (bserial='14989074').
Compute changed=2.
Compute key40(1)=6.
```

```
Compute key40(6)=1.
Compute keypreg(1)=2.
Compute keypreg(2)=1.
End if.

Do if (bserial='15372086').
Compute changed=2.
Compute key40(1)=6.
Compute key40(2)=7.
Compute key40(6)=1.
Compute keypreg(1)=2.
Compute keypreg(2)=1.
End if.


Do if (bserial='16102076').
Compute changed=2.
Compute key40(1)=4.
Compute key40(6)=3.
Compute key40(11)=1.
Compute key40(12)=2.
Compute keypreg(1)=1.
Compute keypreg(2)=1.
Compute keypreg(3)=1.
End if.


Do if (bserial='00250035').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute key40(12)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=1.
Compute keypreg(3)=1.
End if.


Do if (bserial='00665097').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(7)=11.
Compute key40(11)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=4.
End if.


Do if (bserial='00700040').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=1.
Compute keypreg(1)=1.
End if.


Do if (bserial='00820070').
```

```
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute keypreg(1)=1.
End if.


Do if (bserial='00912021').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=12.
Compute key40(12)=11.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
End if.


Do if (bserial='01266041').
Compute changed=2.
Compute key40(1)=11.
Compute key40(6)=6.
Compute key40(11)=1.
Compute key40(12)=2.
Compute keypreg(1)=3.
Compute keypreg(2)=2.
Compute keypreg(3)=1.
End if.


Do if (bserial='01695003').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute key40(16)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
Compute keypreg(4)=4.
End if.


Do if (bserial='02713099').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute keypreg(1)=1.
End if.


Do if (bserial='03379077').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=12.
Compute key40(12)=11.
Compute keypreg(1)=1.
```

```
Compute keypreg(2)=2.
Compute keypreg(3)=3.
End if.


Do if (bserial='04350025').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(7)=11.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.


Do if (bserial='04494008').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute key40(6)=11.
Compute key40(11)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=3.
Compute keypreg(3)=4.
End if.


Do if (bserial='04498009').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=7.
Compute key40(7)=6.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.

Do if (bserial='05170001').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(7)=11.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.

Do if (bserial='05806060').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute key40(6)=11.
Compute keypreg(1)=1.
Compute keypreg(2)=3.
End if.


Do if (bserial='05991047').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute key40(6)=11.
```

```
Compute key40(11)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=3.
Compute keypreg(3)=4.
End if.

Do if (bserial='06145027').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=1.
Compute keypreg(1)=1.
End if.


Do if (bserial='06177006').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute key40(12)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
End if.

Do if (bserial='06180130').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute keypreg(1)=1.
End if.

Do if (bserial='06313002').
Compute changed=2.
Compute key40(1)=2.
Compute key40(6)=1.
Compute keypreg(1)=1.
Compute keypreg(2)=1.
End if.


Do if (bserial='06394088').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=7.
Compute key40(7)=6.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.


Do if (bserial='06481063').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=1.
Compute key40(6)=6.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.
```

```
Do if (bserial='06922067').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute keypreg(1)=1.
End if.


Do if (bserial='07051003').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute key40(12)=16.
Compute key40(16)=21.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
Compute keypreg(4)=5.
End if.


Do if (bserial='07405032').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute key40(6)=11.
Compute keypreg(1)=1.
Compute keypreg(2)=3.
End if.


Do if (bserial='08162010').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute keypreg(1)=1.
End if.


Do if (bserial='08460092').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=1.
Compute key40(6)=6.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.


Do if (bserial='09002054').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(11)=12.
Compute key40(12)=11.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
```

End if.

Do if (bserial='10355068').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute key40(6)=11.
Compute key40(11)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=3.
Compute keypreg(3)=4.
End if.

Do if (bserial='10517068').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(7)=11.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.

Do if (bserial='11257019' or bserial='11519097').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute keypreg(1)=1.
End if.

Do if (bserial='11553050').
Compute changed=2.
Compute key40(1)=1.

Compute key40(6)=6.
Compute key40(11)=11.
Compute key40(12)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
End if.

Do if (bserial='12786037').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=6.
Compute key40(7)=11.
Compute key40(11)=16.
Compute key40(16)=21.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=4.
Compute keypreg(4)=5.
End if.

Do if (bserial='13189078').

```
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=7.
Compute key40(7)=6.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
End if.


Do if (bserial='13930037' or bserial='14137024').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=1.
Compute keypreg(1)=1.
End if.


Do if (bserial='14414078').
Compute changed=2.
Compute key40(1)=6.
Compute key40(2)=1.
Compute keypreg(1)=2.
End if.


Do if (bserial='14812036').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
End if.


Do if (bserial='15675043').
Compute changed=2.
Compute key40(1)=1.
Compute key40(6)=7.
Compute key40(7)=6.
Compute key40(11)=11.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
End if.


Do if (bserial='15838018').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute key40(6)=11.
Compute key40(11)=16.
Compute keypreg(1)=1.
Compute keypreg(2)=3.
Compute keypreg(3)=4.
```

End if.

Do if (bserial='16643042').
Compute changed=2.
Compute key40(1)=1.
Compute key40(2)=6.
Compute keypreg(1)=1.
End if.

Do if (bserial='16992080').
Compute changed=2.
Compute key40(1)=2.
Compute key40(2)=1.
Compute key40(6)=6.
Compute key40(11)=11.
Compute keypreg(1)=1.
Compute keypreg(2)=2.
Compute keypreg(3)=3.
End if.

* For all manually-altered cases, work out pregnum values for pregnancy slots 1-8.

Do if (changed=2).
loop #i=1 to 8.
do if (key40(5*#i-4) ne 0).
compute xpregnu(#i)=1.
loop #j=1 to 4.
if (key40(5*#i+#j-4) ne 0)xpregnu(#i)=xpregnu(#i)+1.
end loop.
end if.
end loop.

loop #i=1 to 40.
Do if(key40(#i) ne 0).
compute xprega(#i)=vprega(key40(#i)).
compute xpregc(#i)=vpregc(key40(#i)).
compute xpregd(#i)=vpregd(key40(#i)).
compute xkilo(#i)=vkilo(key40(#i)).
compute xpregf(#i)=vpregf(key40(#i)).
compute xpregh(#i)=vpregh(key40(#i)).
compute xwhopa(#i)=vwhopa(key40(#i)).
compute xwerkd(#i)=vwerkd(key40(#i)).
compute xwatkd(#i)=vwatkd(key40(#i)).
compute xabspa(#i)=vabspa(key40(#i)).
compute xabspb(#i)=vabspb(key40(#i)).
compute xabspc(#i)=vabspc(key40(#i)).
compute xabskb(#i)=vabskb(key40(#i)).
compute xabskc(#i)=vabskc(key40(#i)).
compute xabskd(#i)=vabskd(key40(#i)).
compute xabske(#i)=vabske(key40(#i)).
compute xpound(#i)=vpound(key40(#i)).
compute xounce(#i)=vounce(key40(#i)).
compute xprged(#i)=vprged(key40(#i)).
compute xprgem(#i)=vprgem(key40(#i)).
compute xpregg(#i)=vpregg(key40(#i)).
compute xwhopb(#i)=vwhopb(key40(#i)).
compute xabska(#i)=vabska(key40(#i)).

```
compute xabsmn(#i)=vabsmn(key40(#i)).
compute xgramm(#i)=vgramm(key40(#i)).
compute xprgey(#i)=vprgey(key40(#i)).
compute xabsyr(#i)=vabsyr(key40(#i)).
compute xprege(#i)=vprege(key40(#i)).
compute xpregi(#i)=vpregi(key40(#i)).
end if.
end loop.

Loop #i=1 to 8.
Do if (keypreg(#i) ne 0).
compute xpregj(#i)=vpregj(keypreg(#i)).
compute xcg1prg(#i)=vcg1prg(keypreg(#i)).
compute xcg2prg(#i)=vcg2prg(keypreg(#i)).
compute xcg3prg(#i)=vcg3prg(keypreg(#i)).
compute xcpregb(#i)=vcpregb(keypreg(#i)).
compute xpregk(#i)=vpregk(keypreg(#i)).
compute xpregkw(#i)=vpregkw(keypreg(#i)).
compute xpregl(#i)=vpregl(keypreg(#i)).
end if.
end loop.

end if.


* Compute new value of morepreg(1-8) from knowledge of whether there are any babies in
the next-most recent pregnancy slot.

Do if (changed=1 or changed=2).

loop #i=1 to 8.
compute xmorprg(#i)=2.
end loop.

loop #i=1 to 7.

If (xpregnu(#i+1) ge 1)xmorprg(#i)=1.
end loop.

* Copy all the data from the x-prefix vectors to the original variables, overwriting data in wrong
birthslots.

compute pregnum =xpregnu1.
compute pregnum2=xpregnu2.
compute pregnum3=xpregnu3.
compute pregnum4=xpregnu4.
compute pregnum5=xpregnu5.
compute pregnum6=xpregnu6.
compute pregnum7=xpregnu7.
compute pregnum8=xpregnu8.


compute pregj =xpregj1.
compute pregj2=xpregj2.
compute pregj3=xpregj3.
compute pregj4=xpregj4.
compute pregj5=xpregj5.
compute pregj6=xpregj6.
compute pregj7=xpregj7.
compute pregj8=xpregj8.
```

```
compute cgprega1=xcg1prg1.
compute cgprega4=xcg1prg2.
compute cgprega7=xcg1prg3.
compute cgpreg10=xcg1prg4.
compute cgpreg13=xcg1prg5.
compute cgpreg16=xcg1prg6.
compute cgpreg19=xcg1prg7.
compute cgpreg22=xcg1prg8.


compute cgprega2=xcg2prg1.
compute cgprega5=xcg2prg2.
compute cgprega8=xcg2prg3.
compute cgpreg11=xcg2prg4.
compute cgpreg14=xcg2prg5.
compute cgpreg17=xcg2prg6.
compute cgpreg20=xcg2prg7.
compute cgpreg23=xcg2prg8.


compute cgprega3=xcg3prg1.
compute cgprega6=xcg3prg2.
compute cgprega9=xcg3prg3.
compute cgpreg12=xcg3prg4.
compute cgpreg15=xcg3prg5.
compute cgpreg18=xcg3prg6.
compute cgpreg21=xcg3prg7.
compute cgpreg24=xcg3prg8.


compute cpregb =xcpregb1.
compute cpregb2=xcpregb2.
compute cpregb3=xcpregb3.
compute cpregb4=xcpregb4.
compute cpregb5=xcpregb5.
compute cpregb6=xcpregb6.
compute cpregb7=xcpregb7.
compute cpregb8=xcpregb8.


compute morepreg =xmorprg1.
compute morepre2=xmorprg2.
compute morepre3=xmorprg3.
compute morepre4=xmorprg4.
compute morepre5=xmorprg5.
compute morepre6=xmorprg6.
compute morepre7=xmorprg7.
compute morepre8=xmorprg8.


compute pregk =xpregk1.
compute pregk2=xpregk2.
compute pregk3=xpregk3.
compute pregk4=xpregk4.
compute pregk5=xpregk5.
compute pregk6=xpregk6.
compute pregk7=xpregk7.
compute pregk8=xpregk8.
```

```
compute pregkw =xpregkw1.
compute pregkw2=xpregkw2.
compute pregkw3=xpregkw3.
compute pregkw4=xpregkw4.
compute pregkw5=xpregkw5.
compute pregkw6=xpregkw6.
compute pregkw7=xpregkw7.
compute pregkw8=xpregkw8.


compute pregl =xpregl1.
compute pregl2=xpregl2.
compute pregl3=xpregl3.
compute pregl4=xpregl4.
compute pregl5=xpregl5.
compute pregl6=xpregl6.
compute pregl7=xpregl7.
compute pregl8=xpregl8.


compute prega =xprega1.
compute prega2=xprega2.
compute prega3=xprega3.
compute prega4=xprega4.
compute prega5=xprega5.
compute prega6=xprega6.
compute prega7=xprega7.
compute prega8=xprega8.
compute prega9=xprega9.
compute prega10=xprega10.
compute prega11=xprega11.
compute prega12=xprega12.
compute prega13=xprega13.
compute prega14=xprega14.
compute prega15=xprega15.
compute prega16=xprega16.
compute prega17=xprega17.
compute prega18=xprega18.
compute prega19=xprega19.
compute prega20=xprega20.
compute prega21=xprega21.
compute prega22=xprega22.
compute prega23=xprega23.
compute prega24=xprega24.
compute prega25=xprega25.
compute prega26=xprega26.
compute prega27=xprega27.
compute prega28=xprega28.
compute prega29=xprega29.
compute prega30=xprega30.
compute prega31=xprega31.
compute prega32=xprega32.
compute prega33=xprega33.
compute prega34=xprega34.
compute prega35=xprega35.
compute prega36=xprega36.
compute prega37=xprega37.
compute prega38=xprega38.
```

```
compute prega39=xprega39.
compute prega40=xprega40.


compute pregc =xpregc1.
compute pregc2=xpregc2.
compute pregc3=xpregc3.
compute pregc4=xpregc4.
compute pregc5=xpregc5.
compute pregc6=xpregc6.
compute pregc7=xpregc7.
compute pregc8=xpregc8.
compute pregc9=xpregc9.
compute pregc10=xpregc10.
compute pregc11=xpregc11.
compute pregc12=xpregc12.
compute pregc13=xpregc13.
compute pregc14=xpregc14.
compute pregc15=xpregc15.
compute pregc16=xpregc16.
compute pregc17=xpregc17.
compute pregc18=xpregc18.
compute pregc19=xpregc19.
compute pregc20=xpregc20.
compute pregc21=xpregc21.
compute pregc22=xpregc22.
compute pregc23=xpregc23.
compute pregc24=xpregc24.
compute pregc25=xpregc25.
compute pregc26=xpregc26.
compute pregc27=xpregc27.
compute pregc28=xpregc28.
compute pregc29=xpregc29.
compute pregc30=xpregc30.
compute pregc31=xpregc31.
compute pregc32=xpregc32.
compute pregc33=xpregc33.
compute pregc34=xpregc34.
compute pregc35=xpregc35.
compute pregc36=xpregc36.
compute pregc37=xpregc37.
compute pregc38=xpregc38.
compute pregc39=xpregc39.
compute pregc40=xpregc40.


compute pregd =xpregd1.
compute pregd2=xpregd2.
compute pregd3=xpregd3.
compute pregd4=xpregd4.
compute pregd5=xpregd5.
compute pregd6=xpregd6.
compute pregd7=xpregd7.
compute pregd8=xpregd8.
compute pregd9=xpregd9.
compute pregd10=xpregd10.
compute pregd11=xpregd11.
compute pregd12=xpregd12.
compute pregd13=xpregd13.
compute pregd14=xpregd14.
```

```
compute pregd15=xpregd15.
compute pregd16=xpregd16.
compute pregd17=xpregd17.
compute pregd18=xpregd18.
compute pregd19=xpregd19.
compute pregd20=xpregd20.
compute pregd21=xpregd21.
compute pregd22=xpregd22.
compute pregd23=xpregd23.
compute pregd24=xpregd24.
compute pregd25=xpregd25.
compute pregd26=xpregd26.
compute pregd27=xpregd27.
compute pregd28=xpregd28.
compute pregd29=xpregd29.
compute pregd30=xpregd30.
compute pregd31=xpregd31.
compute pregd32=xpregd32.
compute pregd33=xpregd33.
compute pregd34=xpregd34.
compute pregd35=xpregd35.
compute pregd36=xpregd36.
compute pregd37=xpregd37.
compute pregd38=xpregd38.
compute pregd39=xpregd39.
compute pregd40=xpregd40.


compute kilo =xkilo1.
compute kilo2=xkilo2.
compute kilo3=xkilo3.
compute kilo4=xkilo4.
compute kilo5=xkilo5.
compute kilo6=xkilo6.
compute kilo7=xkilo7.
compute kilo8=xkilo8.
compute kilo9=xkilo9.
compute kilo10=xkilo10.
compute kilo11=xkilo11.
compute kilo12=xkilo12.
compute kilo13=xkilo13.
compute kilo14=xkilo14.
compute kilo15=xkilo15.
compute kilo16=xkilo16.
compute kilo17=xkilo17.
compute kilo18=xkilo18.
compute kilo19=xkilo19.
compute kilo20=xkilo20.
compute kilo21=xkilo21.
compute kilo22=xkilo22.
compute kilo23=xkilo23.
compute kilo24=xkilo24.
compute kilo25=xkilo25.
compute kilo26=xkilo26.
compute kilo27=xkilo27.
compute kilo28=xkilo28.
compute kilo29=xkilo29.
compute kilo30=xkilo30.
compute kilo31=xkilo31.
compute kilo32=xkilo32.
```

```
compute kilo33=xkilo33.
compute kilo34=xkilo34.
compute kilo35=xkilo35.
compute kilo36=xkilo36.
compute kilo37=xkilo37.
compute kilo38=xkilo38.
compute kilo39=xkilo39.
compute kilo40=xkilo40.


compute pregf =xpregf1.
compute pregf2=xpregf2.
compute pregf3=xpregf3.
compute pregf4=xpregf4.
compute pregf5=xpregf5.
compute pregf6=xpregf6.
compute pregf7=xpregf7.
compute pregf8=xpregf8.
compute pregf9=xpregf9.
compute pregf10=xpregf10.
compute pregf11=xpregf11.
compute pregf12=xpregf12.
compute pregf13=xpregf13.
compute pregf14=xpregf14.
compute pregf15=xpregf15.
compute pregf16=xpregf16.
compute pregf17=xpregf17.
compute pregf18=xpregf18.
compute pregf19=xpregf19.
compute pregf20=xpregf20.
compute pregf21=xpregf21.
compute pregf22=xpregf22.
compute pregf23=xpregf23.
compute pregf24=xpregf24.
compute pregf25=xpregf25.
compute pregf26=xpregf26.
compute pregf27=xpregf27.
compute pregf28=xpregf28.
compute pregf29=xpregf29.
compute pregf30=xpregf30.
compute pregf31=xpregf31.
compute pregf32=xpregf32.
compute pregf33=xpregf33.
compute pregf34=xpregf34.
compute pregf35=xpregf35.
compute pregf36=xpregf36.
compute pregf37=xpregf37.
compute pregf38=xpregf38.
compute pregf39=xpregf39.
compute pregf40=xpregf40.


compute pregh =xpregh1.
compute pregh2=xpregh2.
compute pregh3=xpregh3.
compute pregh4=xpregh4.
compute pregh5=xpregh5.
compute pregh6=xpregh6.
compute pregh7=xpregh7.
compute pregh8=xpregh8.
```

```
compute pregh9=xpregh9.
compute pregh10=xpregh10.
compute pregh11=xpregh11.
compute pregh12=xpregh12.
compute pregh13=xpregh13.
compute pregh14=xpregh14.
compute pregh15=xpregh15.
compute pregh16=xpregh16.
compute pregh17=xpregh17.
compute pregh18=xpregh18.
compute pregh19=xpregh19.
compute pregh20=xpregh20.
compute pregh21=xpregh21.
compute pregh22=xpregh22.
compute pregh23=xpregh23.
compute pregh24=xpregh24.
compute pregh25=xpregh25.
compute pregh26=xpregh26.
compute pregh27=xpregh27.
compute pregh28=xpregh28.
compute pregh29=xpregh29.
compute pregh30=xpregh30.
compute pregh31=xpregh31.
compute pregh32=xpregh32.
compute pregh33=xpregh33.
compute pregh34=xpregh34.
compute pregh35=xpregh35.
compute pregh36=xpregh36.
compute pregh37=xpregh37.
compute pregh38=xpregh38.
compute pregh39=xpregh39.
compute pregh40=xpregh40.


compute whopara =xwhopa1.
compute whopara2=xwhopa2.
compute whopara3=xwhopa3.
compute whopara4=xwhopa4.
compute whopara5=xwhopa5.
compute whopara6=xwhopa6.
compute whopara7=xwhopa7.
compute whopara8=xwhopa8.
compute whopara9=xwhopa9.
compute whopar10=xwhopa10.
compute whopar12=xwhopa11.
compute whopar14=xwhopa12.
compute whopar16=xwhopa13.
compute whopar18=xwhopa14.
compute whopar20=xwhopa15.
compute whopar22=xwhopa16.
compute whopar24=xwhopa17.
compute whopar26=xwhopa18.
compute whopar28=xwhopa19.
compute whopar30=xwhopa20.
compute whopar32=xwhopa21.
compute whopar34=xwhopa22.
compute whopar36=xwhopa23.
compute whopar38=xwhopa24.
compute whopar40=xwhopa25.
compute whopar42=xwhopa26.
```

```
compute whopar44=xwhopa27.
compute whopar46=xwhopa28.
compute whopar48=xwhopa29.
compute whopar50=xwhopa30.
compute whopar52=xwhopa31.
compute whopar54=xwhopa32.
compute whopar56=xwhopa33.
compute whopar58=xwhopa34.
compute whopar60=xwhopa35.
compute whopar62=xwhopa36.
compute whopar64=xwhopa37.
compute whopar66=xwhopa38.
compute whopar68=xwhopa39.
compute whopar70=xwhopa40.


compute wherkid =xwerkd1.
compute wherkid2=xwerkd2.
compute wherkid3=xwerkd3.
compute wherkid4=xwerkd4.
compute wherkid5=xwerkd5.
compute wherkid6=xwerkd6.
compute wherkid7=xwerkd7.
compute wherkid8=xwerkd8.
compute wherkid9=xwerkd9.
compute wherki10=xwerkd10.
compute wherki11=xwerkd11.
compute wherki12=xwerkd12.
compute wherki13=xwerkd13.
compute wherki14=xwerkd14.
compute wherki15=xwerkd15.
compute wherki16=xwerkd16.
compute wherki17=xwerkd17.
compute wherki18=xwerkd18.
compute wherki19=xwerkd19.
compute wherki20=xwerkd20.
compute wherki21=xwerkd21.
compute wherki22=xwerkd22.
compute wherki23=xwerkd23.
compute wherki24=xwerkd24.
compute wherki25=xwerkd25.
compute wherki26=xwerkd26.
compute wherki27=xwerkd27.
compute wherki28=xwerkd28.
compute wherki29=xwerkd29.
compute wherki30=xwerkd30.
compute wherki31=xwerkd31.
compute wherki32=xwerkd32.
compute wherki33=xwerkd33.
compute wherki34=xwerkd34.
compute wherki35=xwerkd35.
compute wherki36=xwerkd36.
compute wherki37=xwerkd37.
compute wherki38=xwerkd38.
compute wherki39=xwerkd39.
compute wherki40=xwerkd40.


compute whatkid =xwatkd1.
compute whatkid2=xwatkd2.
```

```
compute whatkid3=xwatkd3.
compute whatkid4=xwatkd4.
compute whatkid5=xwatkd5.
compute whatkid6=xwatkd6.
compute whatkid7=xwatkd7.
compute whatkid8=xwatkd8.
compute whatkid9=xwatkd9.
compute whatki10=xwatkd10.
compute whatki11=xwatkd11.
compute whatki12=xwatkd12.
compute whatki13=xwatkd13.
compute whatki14=xwatkd14.
compute whatki15=xwatkd15.
compute whatki16=xwatkd16.
compute whatki17=xwatkd17.
compute whatki18=xwatkd18.
compute whatki19=xwatkd19.
compute whatki20=xwatkd20.
compute whatki21=xwatkd21.
compute whatki22=xwatkd22.
compute whatki23=xwatkd23.
compute whatki24=xwatkd24.
compute whatki25=xwatkd25.
compute whatki26=xwatkd26.
compute whatki27=xwatkd27.
compute whatki28=xwatkd28.
compute whatki29=xwatkd29.
compute whatki30=xwatkd30.
compute whatki31=xwatkd31.
compute whatki32=xwatkd32.
compute whatki33=xwatkd33.
compute whatki34=xwatkd34.
compute whatki35=xwatkd35.
compute whatki36=xwatkd36.
compute whatki37=xwatkd37.
compute whatki38=xwatkd38.
compute whatki39=xwatkd39.
compute whatki40=xwatkd40.


Compute abspara =xabspa1.
Compute abspara2=xabspa2.
Compute abspara3=xabspa3.
Compute abspara4=xabspa4.
Compute abspara5=xabspa5.
Compute abspara6=xabspa6.
Compute abspara7=xabspa7.
Compute abspara8=xabspa8.
Compute abspara9=xabspa9.
Compute abspar10=xabspa10.
Compute abspar13=xabspa11.
Compute abspar16=xabspa12.
Compute abspar19=xabspa13.
Compute abspar22=xabspa14.
Compute abspar25=xabspa15.
Compute abspar28=xabspa16.
Compute abspar31=xabspa17.
Compute abspar34=xabspa18.
Compute abspar37=xabspa19.
Compute abspar40=xabspa20.
```

Compute abspar43=xabspa21.
Compute abspar46=xabspa22.
Compute abspar49=xabspa23.
Compute abspar52=xabspa24.
Compute abspar55=xabspa25.
Compute abspar58=xabspa26.
Compute abspar61=xabspa27.
Compute abspar64=xabspa28.
Compute abspar67=xabspa29.
Compute abspar70=xabspa30.
Compute abspar73=xabspa31.
Compute abspar76=xabspa32.
Compute abspar79=xabspa33.
Compute abspar82=xabspa34.
Compute abspar85=xabspa35.
Compute abspar88=xabspa36.
Compute abspar91=xabspa37.
Compute abspar94=xabspa38.
Compute abspar97=xabspa39.
Compute abspa100=xabspa40.

Compute absparb =xabspb1.
Compute absparb2=xabspb2.
Compute absparb3=xabspb3.
Compute absparb4=xabspb4.
Compute absparb5=xabspb5.
Compute absparb6=xabspb6.
Compute absparb7=xabspb7.
Compute absparb8=xabspb8.
Compute absparb9=xabspb9.
Compute abspar11=xabspb10.
Compute abspar14=xabspb11.
Compute abspar17=xabspb12.
Compute abspar20=xabspb13.
Compute abspar23=xabspb14.
Compute abspar26=xabspb15.
Compute abspar29=xabspb16.
Compute abspar32=xabspb17.
Compute abspar35=xabspb18.
Compute abspar38=xabspb19.
Compute abspar41=xabspb20.
Compute abspar44=xabspb21.
Compute abspar47=xabspb22.
Compute abspar50=xabspb23.
Compute abspar53=xabspb24.
Compute abspar56=xabspb25.
Compute abspar59=xabspb26.
Compute abspar62=xabspb27.
Compute abspar65=xabspb28.
Compute abspar68=xabspb29.
Compute abspar71=xabspb30.
Compute abspar74=xabspb31.
Compute abspar77=xabspb32.
Compute abspar80=xabspb33.
Compute abspar83=xabspb34.
Compute abspar86=xabspb35.
Compute abspar89=xabspb36.
Compute abspar92=xabspb37.
Compute abspar95=xabspb38.

```
Compute abspar98=xabspb39.
Compute abspa101=xabspb40.


Compute absparc =xabspc1.
Compute absparc2=xabspc2.
Compute absparc3=xabspc3.
Compute absparc4=xabspc4.
Compute absparc5=xabspc5.
Compute absparc6=xabspc6.
Compute absparc7=xabspc7.
Compute absparc8=xabspc8.
Compute absparc9=xabspc9.
Compute abspar12=xabspc10.
Compute abspar15=xabspc11.
Compute abspar18=xabspc12.
Compute abspar21=xabspc13.
Compute abspar24=xabspc14.
Compute abspar27=xabspc15.
Compute abspar30=xabspc16.
Compute abspar33=xabspc17.
Compute abspar36=xabspc18.
Compute abspar39=xabspc19.
Compute abspar42=xabspc20.
Compute abspar45=xabspc21.
Compute abspar48=xabspc22.
Compute abspar51=xabspc23.
Compute abspar54=xabspc24.
Compute abspar57=xabspc25.
Compute abspar60=xabspc26.
Compute abspar63=xabspc27.
Compute abspar66=xabspc28.
Compute abspar69=xabspc29.
Compute abspar72=xabspc30.
Compute abspar75=xabspc31.
Compute abspar78=xabspc32.
Compute abspar81=xabspc33.
Compute abspar84=xabspc34.
Compute abspar87=xabspc35.
Compute abspar90=xabspc36.
Compute abspar93=xabspc37.
Compute abspar96=xabspc38.
Compute abspar99=xabspc39.
Compute abspa102=xabspc40.


Compute abskidb =xabskb1.
Compute abskidb2=xabskb2.
Compute abskidb3=xabskb3.
Compute abskidb4=xabskb4.
Compute abskidb5=xabskb5.
Compute abskidb6=xabskb6.
Compute abskidb7=xabskb7.
Compute abskidb8=xabskb8.
Compute abskidb9=xabskb9.
Compute abskid11=xabskb10.
Compute abskid16=xabskb11.
Compute abskid21=xabskb12.
Compute abskid26=xabskb13.
Compute abskid31=xabskb14.
```

```
Compute abskid36=xabskb15.
Compute abskid41=xabskb16.
Compute abskid46=xabskb17.
Compute abskid51=xabskb18.
Compute abskid56=xabskb19.
Compute abskid61=xabskb20.
Compute abskid66=xabskb21.
Compute abskid71=xabskb22.
Compute abskid76=xabskb23.
Compute abskid81=xabskb24.
Compute abskid86=xabskb25.
Compute abskid91=xabskb26.
Compute abskid96=xabskb27.
Compute abski101=xabskb28.
Compute abski106=xabskb29.
Compute abski111=xabskb30.
Compute abski116=xabskb31.
Compute abski121=xabskb32.
Compute abski126=xabskb33.
Compute abski131=xabskb34.
Compute abski136=xabskb35.
Compute abski141=xabskb36.
Compute abski146=xabskb37.
Compute abski151=xabskb38.
Compute abski156=xabskb39.
Compute abski161=xabskb40.


Compute abskidc =xabskc1.
Compute abskidc2=xabskc2.
Compute abskidc3=xabskc3.
Compute abskidc4=xabskc4.
Compute abskidc5=xabskc5.
Compute abskidc6=xabskc6.
Compute abskidc7=xabskc7.
Compute abskidc8=xabskc8.
Compute abskidc9=xabskc9.
Compute abskid12=xabskc10.
Compute abskid17=xabskc11.
Compute abskid22=xabskc12.
Compute abskid27=xabskc13.
Compute abskid32=xabskc14.
Compute abskid37=xabskc15.
Compute abskid42=xabskc16.
Compute abskid47=xabskc17.
Compute abskid52=xabskc18.
Compute abskid57=xabskc19.
Compute abskid62=xabskc20.
Compute abskid67=xabskc21.
Compute abskid72=xabskc22.
Compute abskid77=xabskc23.
Compute abskid82=xabskc24.
Compute abskid87=xabskc25.
Compute abskid92=xabskc26.
Compute abskid97=xabskc27.
Compute abski102=xabskc28.
Compute abski107=xabskc29.
Compute abski112=xabskc30.
Compute abski117=xabskc31.
Compute abski122=xabskc32.
```

Compute abski127=xabskc33.
Compute abski132=xabskc34.
Compute abski137=xabskc35.
Compute abski142=xabskc36.
Compute abski147=xabskc37.
Compute abski152=xabskc38.
Compute abski157=xabskc39.
Compute abski162=xabskc40.

Compute abskidd =xabskd1.
Compute abskidd2=xabskd2.
Compute abskidd3=xabskd3.
Compute abskidd4=xabskd4.
Compute abskidd5=xabskd5.
Compute abskidd6=xabskd6.
Compute abskidd7=xabskd7.
Compute abskidd8=xabskd8.
Compute abskidd9=xabskd9.
Compute abskid13=xabskd10.
Compute abskid18=xabskd11.
Compute abskid23=xabskd12.
Compute abskid28=xabskd13.
Compute abskid33=xabskd14.
Compute abskid38=xabskd15.
Compute abskid43=xabskd16.
Compute abskid48=xabskd17.
Compute abskid53=xabskd18.
Compute abskid58=xabskd19.
Compute abskid63=xabskd20.
Compute abskid68=xabskd21.
Compute abskid73=xabskd22.
Compute abskid78=xabskd23.
Compute abskid83=xabskd24.
Compute abskid88=xabskd25.
Compute abskid93=xabskd26.
Compute abskid98=xabskd27.
Compute abski103=xabskd28.
Compute abski108=xabskd29.
Compute abski113=xabskd30.
Compute abski118=xabskd31.
Compute abski123=xabskd32.
Compute abski128=xabskd33.
Compute abski133=xabskd34.
Compute abski138=xabskd35.
Compute abski143=xabskd36.
Compute abski148=xabskd37.
Compute abski153=xabskd38.
Compute abski158=xabskd39.
Compute abski163=xabskd40.

Compute abskide =xabske1.
Compute abskide2=xabske2.
Compute abskide3=xabske3.
Compute abskide4=xabske4.
Compute abskide5=xabske5.
Compute abskide6=xabske6.
Compute abskide7=xabske7.
Compute abskide8=xabske8.

```
Compute abskide9=xabske9.
Compute abskid14=xabske10.
Compute abskid19=xabske11.
Compute abskid24=xabske12.
Compute abskid29=xabske13.
Compute abskid34=xabske14.
Compute abskid39=xabske15.
Compute abskid44=xabske16.
Compute abskid49=xabske17.
Compute abskid54=xabske18.
Compute abskid59=xabske19.
Compute abskid64=xabske20.
Compute abskid69=xabske21.
Compute abskid74=xabske22.
Compute abskid79=xabske23.
Compute abskid84=xabske24.
Compute abskid89=xabske25.
Compute abskid94=xabske26.
Compute abskid99=xabske27.
Compute abski104=xabske28.
Compute abski109=xabske29.
Compute abski114=xabske30.
Compute abski119=xabske31.
Compute abski124=xabske32.
Compute abski129=xabske33.
Compute abski134=xabske34.
Compute abski139=xabske35.
Compute abski144=xabske36.
Compute abski149=xabske37.
Compute abski154=xabske38.
Compute abski159=xabske39.
Compute abski164=xabske40.


compute pound =xpound1.
compute pound2=xpound2.
compute pound3=xpound3.
compute pound4=xpound4.
compute pound5=xpound5.
compute pound6=xpound6.
compute pound7=xpound7.
compute pound8=xpound8.
compute pound9=xpound9.
compute pound10=xpound10.
compute pound11=xpound11.
compute pound12=xpound12.
compute pound13=xpound13.
compute pound14=xpound14.
compute pound15=xpound15.
compute pound16=xpound16.
compute pound17=xpound17.
compute pound18=xpound18.
compute pound19=xpound19.
compute pound20=xpound20.
compute pound21=xpound21.
compute pound22=xpound22.
compute pound23=xpound23.
compute pound24=xpound24.
compute pound25=xpound25.
compute pound26=xpound26.
```

```
compute pound27=xpound27.
compute pound28=xpound28.
compute pound29=xpound29.
compute pound30=xpound30.
compute pound31=xpound31.
compute pound32=xpound32.
compute pound33=xpound33.
compute pound34=xpound34.
compute pound35=xpound35.
compute pound36=xpound36.
compute pound37=xpound37.
compute pound38=xpound38.
compute pound39=xpound39.
compute pound40=xpound40.


compute ounce =xounce1.
compute ounce2=xounce2.
compute ounce3=xounce3.
compute ounce4=xounce4.
compute ounce5=xounce5.
compute ounce6=xounce6.
compute ounce7=xounce7.
compute ounce8=xounce8.
compute ounce9=xounce9.
compute ounce10=xounce10.
compute ounce11=xounce11.
compute ounce12=xounce12.
compute ounce13=xounce13.
compute ounce14=xounce14.
compute ounce15=xounce15.
compute ounce16=xounce16.
compute ounce17=xounce17.
compute ounce18=xounce18.
compute ounce19=xounce19.
compute ounce20=xounce20.
compute ounce21=xounce21.
compute ounce22=xounce22.
compute ounce23=xounce23.
compute ounce24=xounce24.
compute ounce25=xounce25.
compute ounce26=xounce26.
compute ounce27=xounce27.
compute ounce28=xounce28.
compute ounce29=xounce29.
compute ounce30=xounce30.
compute ounce31=xounce31.
compute ounce32=xounce32.
compute ounce33=xounce33.
compute ounce34=xounce34.
compute ounce35=xounce35.
compute ounce36=xounce36.
compute ounce37=xounce37.
compute ounce38=xounce38.
compute ounce39=xounce39.
compute ounce40=xounce40.


compute preged =xprged1.
compute preged2=xprged2.
```

```
compute preged3=xprged3.
compute preged4=xprged4.
compute preged5=xprged5.
compute preged6=xprged6.
compute preged7=xprged7.
compute preged8=xprged8.
compute preged9=xprged9.
compute preged10=xprged10.
compute preged11=xprged11.
compute preged12=xprged12.
compute preged13=xprged13.
compute preged14=xprged14.
compute preged15=xprged15.
compute preged16=xprged16.
compute preged17=xprged17.
compute preged18=xprged18.
compute preged19=xprged19.
compute preged20=xprged20.
compute preged21=xprged21.
compute preged22=xprged22.
compute preged23=xprged23.
compute preged24=xprged24.
compute preged25=xprged25.
compute preged26=xprged26.
compute preged27=xprged27.
compute preged28=xprged28.
compute preged29=xprged29.
compute preged30=xprged30.
compute preged31=xprged31.
compute preged32=xprged32.
compute preged33=xprged33.
compute preged34=xprged34.
compute preged35=xprged35.
compute preged36=xprged36.
compute preged37=xprged37.
compute preged38=xprged38.
compute preged39=xprged39.
compute preged40=xprged40.


compute pregem =xprgem1.
compute pregem2=xprgem2.
compute pregem3=xprgem3.
compute pregem4=xprgem4.
compute pregem5=xprgem5.
compute pregem6=xprgem6.
compute pregem7=xprgem7.
compute pregem8=xprgem8.
compute pregem9=xprgem9.
compute pregem10=xprgem10.
compute pregem11=xprgem11.
compute pregem12=xprgem12.
compute pregem13=xprgem13.
compute pregem14=xprgem14.
compute pregem15=xprgem15.
compute pregem16=xprgem16.
compute pregem17=xprgem17.
compute pregem18=xprgem18.
compute pregem19=xprgem19.
compute pregem20=xprgem20.
```

```
compute pregem21=xprgem21.
compute pregem22=xprgem22.
compute pregem23=xprgem23.
compute pregem24=xprgem24.
compute pregem25=xprgem25.
compute pregem26=xprgem26.
compute pregem27=xprgem27.
compute pregem28=xprgem28.
compute pregem29=xprgem29.
compute pregem30=xprgem30.
compute pregem31=xprgem31.
compute pregem32=xprgem32.
compute pregem33=xprgem33.
compute pregem34=xprgem34.
compute pregem35=xprgem35.
compute pregem36=xprgem36.
compute pregem37=xprgem37.
compute pregem38=xprgem38.
compute pregem39=xprgem39.
compute pregem40=xprgem40.


compute pregg =xpregg1.
compute pregg2=xpregg2.
compute pregg3=xpregg3.
compute pregg4=xpregg4.
compute pregg5=xpregg5.
compute pregg6=xpregg6.
compute pregg7=xpregg7.
compute pregg8=xpregg8.
compute pregg9=xpregg9.
compute pregg10=xpregg10.
compute pregg11=xpregg11.
compute pregg12=xpregg12.
compute pregg13=xpregg13.
compute pregg14=xpregg14.
compute pregg15=xpregg15.
compute pregg16=xpregg16.
compute pregg17=xpregg17.
compute pregg18=xpregg18.
compute pregg19=xpregg19.
compute pregg20=xpregg20.
compute pregg21=xpregg21.
compute pregg22=xpregg22.
compute pregg23=xpregg23.
compute pregg24=xpregg24.
compute pregg25=xpregg25.
compute pregg26=xpregg26.
compute pregg27=xpregg27.
compute pregg28=xpregg28.
compute pregg29=xpregg29.
compute pregg30=xpregg30.
compute pregg31=xpregg31.
compute pregg32=xpregg32.
compute pregg33=xpregg33.
compute pregg34=xpregg34.
compute pregg35=xpregg35.
compute pregg36=xpregg36.
compute pregg37=xpregg37.
compute pregg38=xpregg38.
```

```
compute pregg39=xpregg39.
compute pregg40=xpregg40.


compute whoparb =xwhopb1.
compute whoparb2=xwhopb2.
compute whoparb3=xwhopb3.
compute whoparb4=xwhopb4.
compute whoparb5=xwhopb5.
compute whoparb6=xwhopb6.
compute whoparb7=xwhopb7.
compute whoparb8=xwhopb8.
compute whoparb9=xwhopb9.
compute whopar11=xwhopb10.
compute whopar13=xwhopb11.
compute whopar15=xwhopb12.
compute whopar17=xwhopb13.
compute whopar19=xwhopb14.
compute whopar21=xwhopb15.
compute whopar23=xwhopb16.
compute whopar25=xwhopb17.
compute whopar27=xwhopb18.
compute whopar29=xwhopb19.
compute whopar31=xwhopb20.
compute whopar33=xwhopb21.
compute whopar35=xwhopb22.
compute whopar37=xwhopb23.
compute whopar39=xwhopb24.
compute whopar41=xwhopb25.
compute whopar43=xwhopb26.
compute whopar45=xwhopb27.
compute whopar47=xwhopb28.
compute whopar49=xwhopb29.
compute whopar51=xwhopb30.
compute whopar53=xwhopb31.
compute whopar55=xwhopb32.
compute whopar57=xwhopb33.
compute whopar59=xwhopb34.
compute whopar61=xwhopb35.
compute whopar63=xwhopb36.
compute whopar65=xwhopb37.
compute whopar67=xwhopb38.
compute whopar69=xwhopb39.
compute whopar71=xwhopb40.


Compute abskida=xabska1.
Compute abskida2=xabska2.
Compute abskida3=xabska3.
Compute abskida4=xabska4.
Compute abskida5=xabska5.
Compute abskida6=xabska6.
Compute abskida7=xabska7.
Compute abskida8=xabska8.
Compute abskida9=xabska9.
Compute abskid10=xabska10.
Compute abskid15=xabska11.
Compute abskid20=xabska12.
Compute abskid25=xabska13.
Compute abskid30=xabska14.
```

Compute abskid35=xabska15.
Compute abskid40=xabska16.
Compute abskid45=xabska17.
Compute abskid50=xabska18.
Compute abskid55=xabska19.
Compute abskid60=xabska20.
Compute abskid65=xabska21.
Compute abskid70=xabska22.
Compute abskid75=xabska23.
Compute abskid80=xabska24.
Compute abskid85=xabska25.
Compute abskid90=xabska26.
Compute abskid95=xabska27.
Compute abski100=xabska28.
Compute abski105=xabska29.
Compute abski110=xabska30.
Compute abski115=xabska31.
Compute abski120=xabska32.
Compute abski125=xabska33.
Compute abski130=xabska34.
Compute abski135=xabska35.
Compute abski140=xabska36.
Compute abski145=xabska37.
Compute abski150=xabska38.
Compute abski155=xabska39.
Compute abski160=xabska40.

Compute absmon =xabsmn1.
Compute absmon2=xabsmn2.
Compute absmon3=xabsmn3.
Compute absmon4=xabsmn4.
Compute absmon5=xabsmn5.
Compute absmon6=xabsmn6.
Compute absmon7=xabsmn7.
Compute absmon8=xabsmn8.
Compute absmon9=xabsmn9.
Compute absmon10=xabsmn10.
Compute absmon11=xabsmn11.
Compute absmon12=xabsmn12.
Compute absmon13=xabsmn13.
Compute absmon14=xabsmn14.
Compute absmon15=xabsmn15.
Compute absmon16=xabsmn16.
Compute absmon17=xabsmn17.
Compute absmon18=xabsmn18.
Compute absmon19=xabsmn19.
Compute absmon20=xabsmn20.
Compute absmon21=xabsmn21.
Compute absmon22=xabsmn22.
Compute absmon23=xabsmn23.
Compute absmon24=xabsmn24.
Compute absmon25=xabsmn25.
Compute absmon26=xabsmn26.
Compute absmon27=xabsmn27.
Compute absmon28=xabsmn28.
Compute absmon29=xabsmn29.
Compute absmon30=xabsmn30.
Compute absmon31=xabsmn31.
Compute absmon32=xabsmn32.

```
Compute absmon33=xabsmn33.
Compute absmon34=xabsmn34.
Compute absmon35=xabsmn35.
Compute absmon36=xabsmn36.
Compute absmon37=xabsmn37.
Compute absmon38=xabsmn38.
Compute absmon39=xabsmn39.
Compute absmon40=xabsmn40.


compute gramm =xgramm1.
compute gramm2=xgramm2.
compute gramm3=xgramm3.
compute gramm4=xgramm4.
compute gramm5=xgramm5.
compute gramm6=xgramm6.
compute gramm7=xgramm7.
compute gramm8=xgramm8.
compute gramm9=xgramm9.
compute gramm10=xgramm10.
compute gramm11=xgramm11.
compute gramm12=xgramm12.
compute gramm13=xgramm13.
compute gramm14=xgramm14.
compute gramm15=xgramm15.
compute gramm16=xgramm16.
compute gramm17=xgramm17.
compute gramm18=xgramm18.
compute gramm19=xgramm19.
compute gramm20=xgramm20.
compute gramm21=xgramm21.
compute gramm22=xgramm22.
compute gramm23=xgramm23.
compute gramm24=xgramm24.
compute gramm25=xgramm25.
compute gramm26=xgramm26.
compute gramm27=xgramm27.
compute gramm28=xgramm28.
compute gramm29=xgramm29.
compute gramm30=xgramm30.
compute gramm31=xgramm31.
compute gramm32=xgramm32.
compute gramm33=xgramm33.
compute gramm34=xgramm34.
compute gramm35=xgramm35.
compute gramm36=xgramm36.
compute gramm37=xgramm37.
compute gramm38=xgramm38.
compute gramm39=xgramm39.
compute gramm40=xgramm40.


compute pregey =xprgey1.
compute pregey2=xprgey2.
compute pregey3=xprgey3.
compute pregey4=xprgey4.
compute pregey5=xprgey5.
compute pregey6=xprgey6.
compute pregey7=xprgey7.
compute pregey8=xprgey8.
```

```
compute pregey9=xprgey9.
compute pregey10=xprgey10.
compute pregey11=xprgey11.
compute pregey12=xprgey12.
compute pregey13=xprgey13.
compute pregey14=xprgey14.
compute pregey15=xprgey15.
compute pregey16=xprgey16.
compute pregey17=xprgey17.
compute pregey18=xprgey18.
compute pregey19=xprgey19.
compute pregey20=xprgey20.
compute pregey21=xprgey21.
compute pregey22=xprgey22.
compute pregey23=xprgey23.
compute pregey24=xprgey24.
compute pregey25=xprgey25.
compute pregey26=xprgey26.
compute pregey27=xprgey27.
compute pregey28=xprgey28.
compute pregey29=xprgey29.
compute pregey30=xprgey30.
compute pregey31=xprgey31.
compute pregey32=xprgey32.
compute pregey33=xprgey33.
compute pregey34=xprgey34.
compute pregey35=xprgey35.
compute pregey36=xprgey36.
compute pregey37=xprgey37.
compute pregey38=xprgey38.
compute pregey39=xprgey39.
compute pregey40=xprgey40.


Compute absyr =xabsyr1.
Compute absyr2=xabsyr2.
Compute absyr3=xabsyr3.
Compute absyr4=xabsyr4.
Compute absyr5=xabsyr5.
Compute absyr6=xabsyr6.
Compute absyr7=xabsyr7.
Compute absyr8=xabsyr8.
Compute absyr9=xabsyr9.
Compute absyr10=xabsyr10.
Compute absyr11=xabsyr11.
Compute absyr12=xabsyr12.
Compute absyr13=xabsyr13.
Compute absyr14=xabsyr14.
Compute absyr15=xabsyr15.
Compute absyr16=xabsyr16.
Compute absyr17=xabsyr17.
Compute absyr18=xabsyr18.
Compute absyr19=xabsyr19.
Compute absyr20=xabsyr20.
Compute absyr21=xabsyr21.
Compute absyr22=xabsyr22.
Compute absyr23=xabsyr23.
Compute absyr24=xabsyr24.
Compute absyr25=xabsyr25.
Compute absyr26=xabsyr26.
```

```
Compute absyr27=xabsyr27.
Compute absyr28=xabsyr28.
Compute absyr29=xabsyr29.
Compute absyr30=xabsyr30.
Compute absyr31=xabsyr31.
Compute absyr32=xabsyr32.
Compute absyr33=xabsyr33.
Compute absyr34=xabsyr34.
Compute absyr35=xabsyr35.
Compute absyr36=xabsyr36.
Compute absyr37=xabsyr37.
Compute absyr38=xabsyr38.
Compute absyr39=xabsyr39.
Compute absyr40=xabsyr40.


Compute prege =xprege1.
Compute prege2=xprege2.
Compute prege3=xprege3.
Compute prege4=xprege4.
Compute prege5=xprege5.
Compute prege6=xprege6.
Compute prege7=xprege7.
Compute prege8=xprege8.
Compute prege9=xprege9.
Compute prege10=xprege10.
Compute prege11=xprege11.
Compute prege12=xprege12.
Compute prege13=xprege13.
Compute prege14=xprege14.
Compute prege15=xprege15.
Compute prege16=xprege16.
Compute prege17=xprege17.
Compute prege18=xprege18.
Compute prege19=xprege19.
Compute prege20=xprege20.
Compute prege21=xprege21.
Compute prege22=xprege22.
Compute prege23=xprege23.
Compute prege24=xprege24.
Compute prege25=xprege25.
Compute prege26=xprege26.
Compute prege27=xprege27.
Compute prege28=xprege28.
Compute prege29=xprege29.
Compute prege30=xprege30.
Compute prege31=xprege31.
Compute prege32=xprege32.
Compute prege33=xprege33.
Compute prege34=xprege34.
Compute prege35=xprege35.
Compute prege36=xprege36.
Compute prege37=xprege37.
Compute prege38=xprege38.
Compute prege39=xprege39.
Compute prege40=xprege40.


Compute pregi=xpregi1.
Compute pregi2=xpregi2.
```

```
Compute pregi3=xpregi3.
Compute pregi4=xpregi4.
Compute pregi5=xpregi5.
Compute pregi6=xpregi6.
Compute pregi7=xpregi7.
Compute pregi8=xpregi8.
Compute pregi9=xpregi9.
Compute pregi10=xpregi10.
Compute pregi11=xpregi11.
Compute pregi12=xpregi12.
Compute pregi13=xpregi13.
Compute pregi14=xpregi14.
Compute pregi15=xpregi15.
Compute pregi16=xpregi16.
Compute pregi17=xpregi17.
Compute pregi18=xpregi18.
Compute pregi19=xpregi19.
Compute pregi20=xpregi20.
Compute pregi21=xpregi21.
Compute pregi22=xpregi22.
Compute pregi23=xpregi23.
Compute pregi24=xpregi24.
Compute pregi25=xpregi25.
Compute pregi26=xpregi26.
Compute pregi27=xpregi27.
Compute pregi28=xpregi28.
Compute pregi29=xpregi29.
Compute pregi30=xpregi30.
Compute pregi31=xpregi31.
Compute pregi32=xpregi32.
Compute pregi33=xpregi33.
Compute pregi34=xpregi34.
Compute pregi35=xpregi35.
Compute pregi36=xpregi36.
Compute pregi37=xpregi37.
Compute pregi38=xpregi38.
Compute pregi39=xpregi39.
Compute pregi40=xpregi40.

end if.

* Transfer values of vector compdt to forty permanent variables to facilitate manual scrutiny

compute compda1=compdt1.
compute compda2=compdt2.
compute compda3=compdt3.
compute compda4=compdt4.
compute compda5=compdt5.
compute compda6=compdt6.
compute compda7=compdt7.
compute compda8=compdt8.
compute compda9=compdt9.
compute compda10=compdt10.
compute compda11=compdt11.
compute compda12=compdt12.
compute compda13=compdt13.
compute compda14=compdt14.
compute compda15=compdt15.
compute compda16=compdt16.
compute compda17=compdt17.
```

```
compute compda18=compdt18.
compute compda19=compdt19.
compute compda20=compdt20.
compute compda21=compdt21.
compute compda22=compdt22.
compute compda23=compdt23.
compute compda24=compdt24.
compute compda25=compdt25.
compute compda26=compdt26.
compute compda27=compdt27.
compute compda28=compdt28.
compute compda29=compdt29.
compute compda30=compdt30.
compute compda31=compdt31.
compute compda32=compdt32.
compute compda33=compdt33.
compute compda34=compdt34.
compute compda35=compdt35.
compute compda36=compdt36.
compute compda37=compdt37.
compute compda38=compdt38.
compute compda39=compdt39.
compute compda40=compdt40.

compute newtotby=0.
format newtotby(f1.0).
loop #i=1 to 40.
if(xprega(#i) ne 0)newtotby=newtotby+1.
end loop.

end if.

Variable labels missdany "Warning indicator for birthdates still not picked up".
Value labels missdany
 0 "None missed"
 1 "At least one missed".

fre missdany.

compute babydscp=0.
If(totbaby ne newtotby)babydscp=1.

* temporary.
* select if (changed=1 or changed=2).
* fre babydscp.

* temporary.
* select if (babydscp=1 and (changed=1 or changed=2)).
* fre bserial nserial.

fre pre91flg.
temporary.
select if (pre91flg=1).
fre nserial.

execute.

* re-declare missing values which were 'undeclared' at the start.

missing values prega prega2 prega3 prega4 prega5 prega6 prega7 prega8 prega9 prega10
```

prega11 prega12 prega13 prega14 prega15 prega16 prega17 prega18 prega19 prega20
prega21 prega22 prega23 prega24 prega25 prega26 prega27 prega28 prega29 prega30
prega31 prega32 prega33 prega34 prega35 prega36 prega37 prega38 prega39 prega40 (9).

missing values preged preged2 preged3 preged4 preged5 preged6 preged7 preged8
preged9 preged10
preged11 preged12 preged13 preged14 preged15 preged16 preged17 preged18 preged19
preged20
preged21 preged22 preged23 preged24 preged25 preged26 preged27 preged28 preged29
preged30
preged31 preged32 preged33 preged34 preged35 preged36 preged37 preged38 preged39
preged40
pregem pregem2 pregem3 pregem4 pregem5 pregem6 pregem7 pregem8 pregem9
pregem10
pregem11 pregem12 pregem13 pregem14 pregem15 pregem16 pregem17 pregem18
pregem19 pregem20
pregem21 pregem22 pregem23 pregem24 pregem25 pregem26 pregem27 pregem28
pregem29 pregem30
pregem31 pregem32 pregem33 pregem34 pregem35 pregem36 pregem37 pregem38
pregem39 pregem40 (99).

missing values pregey pregey2 pregey3 pregey4 pregey5 pregey6 pregey7 pregey8 pregey9
pregey10
pregey11 pregey12 pregey13 pregey14 pregey15 pregey16 pregey17 pregey18 pregey19
pregey20
pregey21 pregey22 pregey23 pregey24 pregey25 pregey26 pregey27 pregey28 pregey29
pregey30
pregey31 pregey32 pregey33 pregey34 pregey35 pregey36 pregey37 pregey38 pregey39
pregey40 (9999).