PROOF FOUNDATIONS

(W) WEISER DEFINITION OF SLICING:
Given a program P, a slicing criterion C=<v,s> where v is a variable at statement s, and a slice S:
If P halts on input I, then the value of v at statement s each time s is executed in P is the same in P and S. If P fails to terminate normally, s may be executed more times in S than in P, but P and S compute the same values for v each time s is executed by P.

(A) DATA DEPENDENCE:
We say there exists a data dependence between two expressions when the first expression defines the value of a variable and the second one uses this value in at least one of the possible program executions without being any other expression modifying it.
NOTE: We consider that the arguments passed in a function call and the parameters of that function are a specific case of data dependence where the expression changes its name.

(B) CONTROL DEPENDENCE:
There exists a control dependence between two expressions when the second expression cannot be evaluated without evaluating the first expression.

(C) SEQUENTIAL REDUNDANCE:
When the return expression of a block or a function (the last expression of the block in Erlang) is a variable defined in the previous expression, this can be deleted avoiding the definition of this variable and returning the result of the previous expression, taking this expression the last position of the block and being returned in consequence.

(D) SYNTAX ERRROR:
We say there exists a syntax error in a program when the removal or modification of a chosen expression transforms the program into a non-executable state.

(E) SEMANTIC MODIFICATION:
There exists a semantic modification in an expression when the modification of one of its subexpressions modifies the behaviour of the whole expression.

(F) ABSORBING PROPERTY:
A clause of a conditional or a function statement is absorbing when its guard is always evaluated to true or its pattern always matches.

(G) FULL TEST VALIDATION:
There exists full test validation when an original program and a slice extracted from it can be executed with all possible input values of the original program and the values of the slicing criterion are the same in both executions.
NOTE: We consider in this definition also programs with slicing criteria that are independent of program inputs, where there is only one possible execution.

COLOUR LEGEND

Black: Expressions deleted by executing phase 1 (iterative slicing with the selected slicers)
Red: Expressions deleted by executing phase 2 (modified ORBS algorithm)
Green: Expressions remaining in the quasi-minimal slices
Orange: Slicing Criterion
Brown: Expressions deleted by the demonstration but not automatically by the process

NOTE1: We will not prove whether black expressions of the program code can be deleted or not because they have been deleted by phase 1. Phase 1 produces a complete slice of the original code, so we can guarantee that these expressions are not part of the slice.
NOTE2: Our slices keep the syntax of the original program (we are not interested in amorphous slices). However, in order to make the final slice executable, some modifications of the source code are compulsory (e.g., replacing calls to deleted functions with a constant called "undef"). Therefore, we allow for some modifications of the source code to produce executable slices. The modifications made never affect the behaviour of the source code, they just ensure that the final code is a valid Erlang program.

```
%-------------------------------------------------------------------------------------------
%-------------------------------------------------------------------------------------------
%-- bench1.erl
%--
%-- AUTHORS:      Anonymous
%-- DATE:         2016
%-- PUBLISHED:    Software specially developed to test compression and expansion of
%--               complex structures.
%-- COPYRIGHT:    Bencher: The Program Slicing Benchmark Suite for Erlang
%--               (Universitat Politècnica de València)
%--               http://www.dsic.upv.es/~jsilva/slicing/bencher/
%-- DESCRIPTION
%-- This benchmark consists in a program with a database of singers. It returns a set
%-- of information of each singer (Name,Age,Last Album Name, and Location and Year of his
%-- or her next concerte) by receiving a number between 1 and 6 as input.
%-------------------------------------------------------------------------------------------
%-------------------------------------------------------------------------------------------

-module(bench1).
-export([main/1]).

%All expressions in red can be deleted because of (G) due to the limited number of different inputs allowed by the program.

main(Number) when Number > 0 andalso Number < 7 ->              %The when Number > 0 andalso Number < 7 guard can be deleted because
                                                                of (W). The computed values of the slice are not important if the
                                                                original program fails to terminate normally
                                                                %Given (A), Number is necessary w.r.t. lists:nth(Number,Database)
         Database = [                                           %Given (A), Database is necessary w.r.t. lists:nth(Number,Database)
                                                                %Replace […] with undef (NOTE2) would prevent to reach the SC due
                                                                to a matching error in lists:nth(Number,Database)
```

```
["Rihanna",28,
{albums,[{"Music of the Sun",2005},{"A Girl like me",2006},{"Good Girl Gone Bad",2007},{"Rater D",2009},
{"Loud",2010},{"Talk That Talk",2011},{"Unapologetic",2012},{"ANTi",2016}]},
{concerts,[{"Amsterdam",{2016,6,17}}, {"Manchester",{2016,6,29}}, {"Barcelona",{2016,7,21}}, {"Bucarest",{2022,8,14}}]}],

["Christina Aguilera",35,
{albums,[{"Christina Aguilera",1999},{"Mi Reflejo",2000},{"Stripped",2002},{"Back to Basics",2006},
{"Bionic",2010},{"Lotus",2012}]},
{concerts,[{"Tokio",{2007,6,21}},{"Abu Dabi",{2008,10,28}}]}],

["Bruno Mars",30,
{albums,[{"Doo-Wops & Hooligans",2010},{"Unorthodox Jukebox",2012}]},
{concerts,[{"Santo Domingo",{2014,10,4}},{"Las Vegas",{2014,10,18}},{"Liverpool",{2013,11,24}}]}],

["Daddy Yankee",39,
{albums,[{"No Mercy",1995},{"El Cangri.Com",2002},{"Barrio Fino",2004},{"El Cartel: The Big Boss",2007},
{"Mundial",2010},{"Prestige",2012},{"Cartel IV",2015}]},
{concerts,[{"Mexico City",{2015,11,8}},{"Las Vegas",{2016,5,6}},{"New York",{2022,7,30}}]}],

["Justin Bieber",22,
{albums,[{"My World 2.0",2010},{"Under the mistletoe",2011},{"Believe",2012},{"Purpose",2015}]},
{concerts,[{"Miami",{2016,7,3}},{"Munich",{2016,9,16}},{"Birmingham",{2022,10,24}}]}],
```

%Replace ["Rihanna",…] with undef (NOTE2) would prevent to reach the SC because of a matching error in expression [_,_,_,{concerts,Concerts}] = Artist of the getConcerts function

%Replace {concerts,[…]} with undef (NOTE2) would prevent to reach the SC because of a matching error in expression [_,_,_,{concerts,Concerts}] = Artist of the getConcerts function
%Replace [{"Amsterdam",{2016,6,17}},…] with undef (NOTE2) would prevent to reach the SC because of a matching error in the clauses of the getNext function
%Replace {"Amsterdam",{2016,6,17}},{"Manchester",{2016,6,29}}, {"Barcelona",{2016,7,21}} or {"Bucarest",{2022,8,14}} with undef (NOTE2) would prevent to reach the SC because of a matching error in expression {_,Date}=Concert of the getNext function
%Replace 2022 with undef (NOTE2) would prevent to reach the SC because of a matching error in expression {Location,Date} = Concert in the getConcertLocationAndYear function

%Replace ["Christina Aguilera",…] with undef (NOTE2) would prevent to reach the SC because of a matching error in expression [_,_,_,{concerts,Concerts}] = Artist of the getConcerts function

%Replace {concerts,[…]} with undef (NOTE2) would prevent to reach the SC because of a matching error in expression [_,_,_,{concerts,Concerts}] = Artist of the getConcerts function
%Replace [{"Tokio",{2007,6,21}},…] with undef (NOTE2) would prevent to reach the SC because of a matching error in the clauses of the getNext function
%Replace {"Tokio",{2007,6,21}} or {"Abu Dabi",{2008,10,28}} with undef (NOTE2) would prevent to reach the SC because of a matching error in expression {_,Date}=Concert of the getNext function
%Replace 2008 with undef (NOTE2) would prevent to satisfy (2) because of (E) in guard {YA,YC,_,_,_,_} when YC > YA of the case expression in the getNext function

%Replace ["Bruno Mars",…] with undef (NOTE2) would prevent to reach the SC because of a matching error in expression [_,_,_,{concerts,Concerts}] = Artist of the getConcerts function

%Replace {concerts,[…]} with undef (NOTE2) would prevent to reach the SC because of a matching error in expression [_,_,_,{concerts,Concerts}] = Artist of the getConcerts function
%Replace [{"Santo Domingo",{2014,10,4}},…] with undef (NOTE2) would prevent to reach the SC because of a matching error in the clauses of the getNext function
%Replace {"Santo Domingo",{2014,10,4}},{"Las Vegas",{2014,10,18}} or {"Liverpool",{2013,11,24}} with undef (NOTE2) would prevent to reach the SC because of a matching error in expression {_,Date}=Concert of the getNext function
%Replace 2013 with undef (NOTE2) would prevent to satisfy (3) because of (E) in guard {YA,YC,_,_,_,_} when YC > YA of the case expression in the getNext function

%Replace ["Daddy Yankee",…] with undef (NOTE2) would prevent to reach the SC because of a matching error in expression [_,_,_,{concerts,Concerts}] = Artist of the getConcerts function

%Replace {concerts,[…]} with undef (NOTE2) would prevent to reach the SC because of a matching error in expression [_,_,_,{concerts,Concerts}] = Artist of getConcerts function
%Replace [{"Mexico City",{2015,11,8}},…] with undef (NOTE2) would prevent to reach the SC because of a matching error in getNext function clauses
%Replace {"Mexico City",{2015,11,8}},{"Las Vegas",{2016,5,6}} or {"New York",{2022,7,30}} with undef (NOTE2) would prevent to reach the SC because of a matching error in expression {_,Date}=Concert of the getNext function
%Replace 2022 with undef (NOTE2) would prevent to reach the SC because of a matching error in expression {Location,Date} = Concert in the getConcertLocationAndYear function

%Replace ["Justin Bieber",…] with undef (NOTE2) would prevent to reach the SC because of a matching error in expression [_,_,_,{concerts,Concerts}] = Artist of the getConcerts function

%Replace {concerts,[…]} with undef (NOTE2) would prevent to reach the SC because of a matching error in expression [_,_,_,{concerts,Concerts}] = Artist of the getConcerts function
%Replace [{"Miami",{2016,7,3}},…] with undef (NOTE2) would prevent to reach the SC because of a matching error in the clauses of the getNext function
%Replace {"Miami",{2016,7,3}},{"Munich",{2016,9,16}} or {"Birmingham",{2022,10,24}} with undef (NOTE2) would prevent to reach the SC because of a matching error in expression {_,Date}=Concert of the getNext function

```
                                                    %Replace 2022 with undef (NOTE2) would prevent to reach the SC
                                                    because of a matching error in expression {Location,Date} = Concert
                                                    in the getConcertLocationAndYear function

        ["Adele",28,                                %Replace ["Adele",…] with undef (NOTE2) would prevent to reach the
                                                    SC because of a matching error in expression
                                                    [_,_,_,{concerts,Concerts}] = Artist of the getConcerts function

        {albums,[{"19",2008},{"21",2011},{"25",2015}]},
        {concerts,[{"Lisboa",{2016,5,22}},{"Paris",{2016,6,10}},{"Oakland",{2016,8,2}},{"Toronto",{2022,10,6}}]}]
                                                    %Replace {concerts,[…]} with undef (NOTE2) would prevent to reach
                                                    the SC because of a matching error in expression
                                                    [_,_,_,{concerts,Concerts}] = Artist of the getConcerts function
                                                    %Replace [{"Lisboa",{2016,5,22}},…] with undef (NOTE2) would
                                                    prevent to reach the SC because of a matching error in the clauses
                                                    of the getNext function
                                                    %Replace {"Lisboa",{2016,5,22}},{"Paris",{2016,6,10}},
                                                    {"Oakland",{2016,8,2}} or {"Toronto",{2022,10,6}} with undef
                                                    (NOTE2) would prevent to reach the SC because of a matching error
                                                    in expression {_,Date}=Concert of the getNext function
                                                    %Replace 2022 with undef (NOTE2) would prevent to reach the SC
                                                    because of a matching error in expression {Location,Date} = Concert
                                                    in the getConcertLocationAndYear function
        ],
        Artist = lists:nth(Number,Database),        %Given (A), Artist is necessary w.r.t getNextConcert(Artist)
                                                    %Replace lists:nth(Number,Database) with undef (NOTE2) would
                                                    prevent to reach the SC due to a matching error in function
                                                    getConcerts(Artist)
                                                    %Number or Database cannot be deleted because lists:nth is a remote
                                                    function of the erlang library and all parameters are required to
                                                    get the result
        ArtistName = getArtistName(Artist),
        Age = getAge(Artist),
        LastAlbum = getLastAlbum(Artist),
        AlbumName = getAlbumName(LastAlbum),
        NextConcert = getNextConcert(Artist),       %Given (A), NextConcert is necessary w.r.t.
                                                    getConcertLocationAndYear(NextConcert)
                                                    %Replace getNextConcert(Artist) with undef (NOTE2) would prevent
                                                    to reach the SC due to a matching error in function
                                                    getConcertLocationAndYear(NextConcert)
                                                    %Replace Artist with undef (NOTE2) would prevent to reach the SC
                                                    due to a matching error in function getConcerts(Artist)
        Info = getConcertLocationAndYear(NextConcert),  %Given (A), Info is necessary w.r.t. {Location,Year} = Info
                                                    %Replace getConcertLocationAndYear(NextConcert) with undef (NOTE2)
                                                    would prevent to reach the SC due to a matching error in
                                                    {Location,Year} = Info
                                                    %Replace NextConcert with undef (NOTE2) would produce a matching
                                                    error in the {Location,Date} = Concert expression in function
                                                    getConcerLocationAndYear(Concert)
        {Location,Year} = Info,                     %{Location,Year} cannot be replaced with _ (NOTE2) because it would
                                                    delete the SC
                                                    %Year cannot be deleted because it is the SC
                                                    %Replace Info with undef (NOTE2) would prevent to reach the SC due
                                                    to a matching error. It is also the only expression that can assign
                                                    a value to the SC
        {ArtistName,Age,AlbumName,Location,Year}.

getArtistName(Artist) ->
        [Name|_] = Artist,
        Name.

getAge(Artist) ->
        [_,Age|_] = Artist,
        Age.

getDiscography(Artist) ->
        [_,_,{albums,Discography}|_] = Artist,
        Discography.

getConcerts(Artist) ->                              %Given (A), Artist is necessary w.r.t.
                                                    [_,_,_,{concerts,Concerts}] = Artist
        [_,_,_,{concerts,Concerts}] = Artist,       %Replace [_,_,_{concerts,Concerts}] with _ (NOTE2) would prevent
                                                    to reach the SC because of (D). We can avoid this error by deleting
                                                    the expression Concerts, but this would prevent to reach the SC due
                                                    to a matching error in expression {_,Date}=Concert of function
                                                    getNext([Concert|Concerts],Actual,empty)
                                                    %Replace Concerts in {concerts,Concerts} with undef (NOTE2) would
                                                    prevent to reach the SC because of (D). We can avoid this error
                                                    replaicing Concerts with undef, but this would prevent to reach the
                                                    SC because of a matching error in the clauses of the getNext
                                                    function
        Concerts.                                   %Delete Concerts would prevent to reach the SC due to a matching
                                                    error in expression {_,Date}=Concert of function
                                                    getNext([Concert|Concerts],Actual,empty). Replacing it with undef
                                                    (NOTE2) would also prevent to reach the SC due to a matching error
                                                    in the clauses of the getNext function

getLastAlbum(Artist) ->
        Albums = getDiscography(Artist),
        case getLast(Albums,empty) of
                empty -> "No albums published";
                Album -> Album
        end.

getLast([],A) -> A;
getLast([Album|Albums],empty) ->
        getLast(Albums,Album);
```

```
getLast([Album|Albums],Newest) ->
        {_,YearN} = Newest,
        {_,YearA} = Album,
        case YearA > YearN of
                true -> getLast(Albums,Album);
                _ -> getLast(Albums,Newest)
        end.

getNextConcert(Artist) ->
        ActualDate = erlang:date(),

        Concerts = getConcerts(Artist),

        case getNext(Concerts,ActualDate,empty) of

                empty -> "No future concerts planned";

                Concert -> Concert

        end.

getNext([],_,NextConcert) ->

        NextConcert;

getNext([Concert|Concerts],Actual,empty) ->

        {YearA,MonthA,DayA} = Actual,

        {_,Date}=Concert,

        {YearC,MonthC,DayC} = Date,

        case {YearA,YearC,MonthA,MonthC,DayA,DayC} of
```

%Given (A), Artist is necessary w.r.t. getConcerts(Artist)
%Given (A), ActualDate is necessary w.r.t. getNext(Concerts,ActualDate,empty)
%Replace erlang:date() with undef (NOTE2) would prevent to reach the SC due to a matching error in expression {YearA,MonthA,DayA} = Actual in the getNext function
%Given (A), Concerts is necessary w.r.t. getNext(Concerts,ActualDate,empty)
%Replace getConcerts(Artist) with undef (NOTE2) would prevent to reach the SC due to a matching error in the clauses of the getNext function
%Replace Artist with undef (NOTE2) would prevent to reach the SC because of a matching error in expression [_,_,_,{concerts,Concerts}] = Artist in the getConcerts(Artist) function

%The case expression cannot be deleted because it is the last expression of the function and its value will be the returned value of the getNextConcert function. Delete it would also cause a matching error in function call getConcertLocationAndYear(NextConcert) in the main function
%Replace getNext(Concerts,ActualDate,empty) with undef (NOTE2) would prevent to reach the SC because of a matching error in function getConcertLocationAndYear. This could be solved by replacing the empty clause with _ but this would prevent to fulfill (1),(4),(5)&(6)
%Replace Concerts with undef (NOTE2) would prevent to reach the SC due to a matching error in the clauses of the getNext function
%Replace ActualDate with undef (NOTE2) would prevent to reach the SC due to a matching error in the getNext function
%Delete this clause would prevent to reach the SC because of a matching error in expression {Location,Date} = Concert in the getConcertLocationAndYear function
%Replace empty with undef (NOTE2) would prevent to satisfy (1),(4),(5)&(6) because this clause would fulfill (F)
%Replace "No future concerts planned" with undef (NOTE2) would prevent to reach the SC because of a matching error in expression {Location,Date} = Concert in function getConcertLocationAndYear
%Delete this clause would prevent to reach the SC in (1),(4),(5)&(6) because of a matching error in the case expression
%Replace the Concert pattern with _ would produce (D). This could be solved by replacing the Concert expression in the clause body with undef, but this would lead to a matching error in expression {Location,Date} = Concert in the getConcertLocationAndYear function

%This clause cannot be deleted because it is the base case of the getNext function. Deleting this clause would prevent to reach the SC because of a matching error
%Replace [] with _ (NOTE2) would make this clause fulfills (F), and this would prevent to reach the SC because of a matching error in expression {Location,Date} = Concert of the getConcertLocationAndYear function
%Given (A), NextConcert is necessary w.r.t. NextConcert;
%NextConcert cannot be deleted because it is the only expression of the function clause. It is the only expression that can assign a value to the getNext function calls because this clause is the base case of the function. Replace it with undef (NOTE2) would prevent to reach the SC because of a matching error

%Given (A), Concert in [Concert|Concerts] is necessary w.r.t. {_,Date}=Concert
%Given (A), Concerts in [Concert|Concerts] is necessary w.r.t. the case expression
%Given (A), Actual is necessary w.r.t. {YearA,MonthA,DayA} = Actual
%Replace Actual with undef (NOTE2) would prevent to reach the SC due to a matching error
%Given (A), YearA is necessary w.r.t. the case expression
%Replace Date with undef (NOTE2) would prevent to reach the SC because of a matching error
%Given (A), Date is necessary w.r.t. {YearC,MonthC,DayC} = Date
%Replace Date with undef (NOTE2) would prevent to reach the SC due to a matching error
%Given (A), YearC is necessary w.r.t. the case expression
%case expression is the last expression of the clause and its returned value is the returned value of the getNext function. Deleting it would prevent to reach the SC because of a matching error in expression {Location,Date} = Concert of the getConcertLocationAndYear function
%Replace {YearA,YearC,MonthA,MonthC,DayA,DayC} with undef (NOTE2) would prevent to satisfy (1),(4),(5)&(6)
%Replace YearA with undef (NOTE2) would prevent to satisfy (1),(4),(5)&(6) because of (E). Case clause {YA,YC,_,_,_,_} when YC > YA would never match
%Replace YearC with undef (NOTE2) would prevent to satisfy (2)&(3) because case clause 1 {YA,YC,_,_,_,_} when YC > YA would fulfill (F)

```
                {YA,YC,_,_,_,_} when YC > YA ->              %This clause cannot be deleted because it would prevent to satisfy
                                                             (1),(4),(5)&(6)
                                                             %{YA,YC,_,_,_,_} when YC > YA cannot be replaced with _ (NOTE2)
                                                             because it would prevent to satisfy (2)&(3) because the clause
                                                             would fulfill (F)
                                                             %Guard when YC > YA cannot be replaced with true (NOTE2) because
                                                             it would prevent to satisfy (2)&(3) because the clause would fulfill
                                                             (F)
                                                             %Replace YC in when YC > YA with undef (NOTE2) would prevent to
                                                             satisfy (2)&(3) because this clause would fulfill (F)
                                                             %Replace YA in when YC > YA with undef (NOTE2) would prevent to
                                                             satisfy (1),(4),(5)&(6) because of (E). Case clause {YA,YC,_,_,_,_}
                                                             when YC > YA would never match
                                                             %Given (A), YA and YC are necessary w.r.t. the guard when YC > YA
                        getNext(Concerts,Actual,Concert);
                                                             %getNext(Concerts,Actual,Concert) cannot be deleted because it is
                                                             the only expression of the clause. Replace it with undef (NOTE2)
                                                             would prevent to reach the SC because of a matching error in
                                                             expression {Location,Date} = Concert of the
                                                             getConcertLocationAndYear function
                                                             %Given (A), Concerts is necessary w.r.t. the first parameter of the
                                                             clauses of the getNext function. Replace Concerts with undef
                                                             (NOTE2) would prevent to reach the SC because of a matching error
                                                             in function getNext. In order to avoid this error, we can replace
                                                             the getNext([Concert|Concerts],Actual,empty) clause with
                                                             getNext(_,Actual,empty) but this would produce another matching
                                                             error because part of the replaced list ([Concert|Concerts]) cannot
                                                             be deleted since it would prevent to reach the SC due to a matching
                                                             error. There is also possible to replace the first clause of the
                                                             getNext([],_,NextConcert) function with getNext(_,_,NextConcert)
                                                             but this would make it to fulfill (F) and it would prevent to
                                                             satisfy (1),(4),(5)&(6)
                                                             %Given (A), Actual is necessary w.r.t.
                                                             getNext([Concert|Concerts],Actual,empty). It cannot be replaced
                                                             with undef (NOTE2) because it is used in the body of the clause and
                                                             replacing it with undef would prevent to reach the SC due to a
                                                             matching error in the {YearA,MonthA,DayA} = Actual expression
                                                             %Given (A), Concert is necessary w.r.t. getNext([],_,NextConcert)
                                                             in the base case function clause. Replace it with undef (NOTE2)
                                                             would prevent to reach the SC because of a matching error in
                                                             expression {Location,Date} = Concert of the
                                                             getConcertLocationAndYear function
                {Y,Y,MA,MC,_,_} when MC > MA -> getNext(Concerts,Actual,Concert);
                {Y,Y,M,M,DA,DC} when DC >= DA -> getNext(Concerts,Actual,Concert);
                _ -> getNext(Concerts,Actual,empty)          %This clause cannot be deleted because it would prevent to reach
                                                             the SC because of a matching error in the case expression. This
                                                             could be solved by replacing the previous clause with _ but this
                                                             would prevent to satisfy (2)&(3)
                                                             %getNext(Concerts,Actual,empty) cannot be deleted because it is the
                                                             only expression of the clause. Replace it with undef (NOTE2) would
                                                             prevent to reach the SC because of a matching error in expression
                                                             {Location,Date} = Concert of the getConcertLocationAndYear function
                                                             %Given (A), Concerts is necessary w.r.t. the first parameter of the
                                                             clauses of the getNext function. Replace Concerts with undef
                                                             (NOTE2) would prevent to reach the SC because of a matching error
                                                             in function getNext. In order to avoid this error we can replace
                                                             the getNext([Concert|Concerts],Actual,empty) clause with
                                                             getNext(_,Actual,empty) but this would produce another matching
                                                             error because part of the replaced list ([Concert|Concerts]) cannot
                                                             be deleted since it would prevent to reach the SC due to a matching
                                                             error. It is also possible to replace the first clause of the
                                                             getNext([],_,NextConcert) function with getNext(_,_,NextConcert)
                                                             but this would make it to fulfill (F) and it would prevent to
                                                             satisfy (1),(4),(5)&(6)
                                                             %Given (A), empty is necessary w.r.t. getNext([],_,NextConcert) in
                                                             the base case function clause. Replace it with undef (NOTE2) would
                                                             prevent to reach the SC because of a matching error in expression
                                                             {Location,Date} = Concert of the getConcertLocationAndYear function

        end;
getNext([Concert|Concerts],Actual,NextConcert) ->
        {YearA,MonthA,DayA} = Actual,
        {_,DateN}=NextConcert,
        {YearN,MonthN,DayN} = DateN,
        {_,DateC}=Concert,
        {YearC,MonthC,DayC} = DateC,

        Next = case {YearA,YearC,MonthA,MonthC,DayA,DayC} of
                {YA,YC,_,_,_,_} when YC < YA -> getNext(Concerts,Actual,NextConcert);
                {Y,Y,MA,MC,_,_} when MC < MA -> getNext(Concerts,Actual,NextConcert);
                {Y,Y,M,M,DA,DC} when DC < DA -> getNext(Concerts,Actual,NextConcert);
                _ -> empty
        end,
        case Next of
                empty ->
                        case {YearN,YearC,MonthN,MonthC,DayN,DayC} of
                                {YN,YC2,_,_,_,_} when YC2 < YN -> getNext(Concerts,Actual,Concert);
                                {Y2,Y2,MN,MC2,_,_} when MC2 < MN -> getNext(Concerts,Actual,Concert);
                                {Y2,Y2,M2,M2,DN,DC2} when DC2 < DN -> getNext(Concerts,Actual,Concert);
                                _ -> getNext(Concerts,Actual,NextConcert)
                        end;
                _ -> Next
        end.
```

```
getAlbumName(Album) ->
        {Name,_} = Album,
        Name.

getStringDate(Concert) ->
        {_,Date} = Concert,
        {Y,M,D} = Date,
        integer_to_list(D)++"/"++integer_to_list(M)++"/"++integer_to_list(Y).

getConcertLocationAndYear("No future concerts planned") ->      %This clause cannot be deleted because it would prevent to satisfy (2)&(3)
                                                                %Replace "No future concerts planned" with _ (NOTE2) would prevent to satisfy (1),(4),(5)&(6) because this clause would fulfill (F)
        {"an undefined City", "Not planned yet"};               %Replace {"an undefined City", "Not planned yet"} with undef (NOTE2) would prevent to reach the SC because of a matching error in expression {Location,Year} = Info in function main
                                                                %Replace "Not planned yet" with undef (NOTE2) would prevent to satisfy (1),(2),(3),(4),(5)&(6)
getConcertLocationAndYear(Concert) ->                           %This clause cannot be deleted because it would prevent to reach the SC in (1),(4),(5)&(6) because of a matching error
                                                                %Given (A), Concert is necessary w.r.t. {Location,Date} = Concert
        {Location,Date} = Concert,                              %Given (A), Date is necessary w.r.t. {Year,Month,Day} = Date
                                                                %Replace Concert with undef (NOTE2) would prevent to reach the SC due to a matching error
        {Year,Month,Day} = Date,                                %Given (A), Year is necessary w.r.t. Info = {Location,Year}
                                                                %Replace Date with undef (NOTE2) would prevent to reach the SC due to a matching error
        Info = {Location,Year},                                 %Replace Year with undef (NOTE2) would prevent to satisfy (1),(4),(5)&(6)

        Info.
```

EXECUTION RESULTS:

```
Number = 1                                                      SLICING CRITERION
Number = 2                                                          SC = 2022
Number = 3                                                          SC = "Not planned yet"
Number = 4                                                          SC = "Not planned yet"
Number = 5                                                          SC = 2022
Number = 6                                                          SC = 2022
                                                                    SC = 2022
```