

昨年度はACCESSの基礎と応用を「さら〜っと」教えてもらった。  
先生によると、ACCESSというデータベースソフトは奥が深く、  
“業務”をまるごと作り上げるようなこともできるという。  
そこで、せっかく教わるなら自分の仕事に役立つように  
“運用実績管理業務”を作ってみようと考えた。  
目標は年度末の完成！（なんて、公言してイイんかなあ…）

第10回

これは、アクセスにあくせんす（悪戦す）る苦闘の日記である。（^^;）

### 前回までのあらすじ

ACCESSで業務処理を完成させる！と決意した野口は、システム部門の1年先輩に教を乞う。先輩は毎回ヤキニクを奢ることを条件に、手取り足取り教えてくれることになった。

（^^;）ネマツネジシモACCESSが、アタカラハルナイ…

テーブルの基本設計と作成、リレーションシップの作成、データ入力のフォームを作り、マクロを見よう見まねで作成、クエリ・フォームと連携させた。部分処理とはいえ、考えた業務の流れ通りに画面が動くので大感激の野口。さて、最後の難関（？）、“くり返し処理”（収益率の積算）ができるのかどうか？果たして業務システム構築のゆくえは如何に？

### 平成17年 元旦

野口：「もしもし。先輩、明けましておめでとうございます。」

先輩：「ふえ〜い、なに？だれ？なんだ、野口か。どうしたの、こんな朝早くから。」

野口：「もうお昼です！元旦ですよ。1年の始めなんだから、もう少しシャキッとしてくださいよ。」

先輩：「ああ、そうか。フむ、明けましておめでとうございます。もう少し寝かせてえ。じゃね。」

野口：「ちょっと！目え覚ましてくださいよ！ACCESSの宿題、見てもらいたいですよ。」

先輩：「えええ〜。新年早々だよ。会社始まってからでイイじゃん？」

野口：「それが、いろいろ気になるもので、大晦日も紅白歌合戦どころじゃなかったし…

一緒に初詣でに行きませんか？そのついでにちょっと見てくださいよ。お願いしますよ。」

先輩：「…？」

野口：「明治神宮でいいですか？渋谷の『ライオン』って名曲喫茶知ってます？そこに明日の午後2時に待ち合わせでどうです？」

先輩：「…」

野口：「先輩っ！聞いているんですか！？」

先輩：「へ？…ああ、びっくりした。聞こえた聞こえた。ヘンなやつだな。じゃ明日の2時ね。」

野口：「あっ、場所、だいじょ…」

先輩：「渋谷『ライオン』ね。じゃね。」（プツン！ツーツー♪）

野口：「あ〜あ。なに慌ててるんだろ？ま、いいか。」

### 翌日 渋谷・午後2時

先輩：「お待たせ！」

野口：「（！！…）」

先輩：「どしたの？」

野口：「どうしたって、こっちが聞きたい！先輩、和服着てきたんですか！？」

先輩：「見りゃわかるじゃん。お正月だもん、不思議くないでしょ。」

野口：「いや、そりゃ個人の自由…あ、いや不思議でもナンでもないっすけど、はあ。」

先輩：「なに、その間の抜けた言い方。張り切って着てきたんだぞ、ちょっとくらい誉めてみたら？」

野口：「あ、そです。え、え…と、とても似合ってますよ。ほんと。」

先輩：「はあ〜、まあキミにはそんなところかなあ。ま、いいか♪では初ACCESSを始めるかね。」

野口：「はいっ♪よろしく願います！」

先輩:「今回は、“くり返し処理”として『収益率の積算』をやりたいんだっけ。具体的な説明の前に、積算処理について確認しておこう。図1のような処理を想定しているんだけど、これで良いかな？」

野口:「はい。まず、データの“収益率”は100%を基準にプラスマイナスの表現なので、これを少数の単位に直す必要があります。つまり+15%の収益率は、1.15、▲5%の収益率は0.95とするために、収益率の数値を100で割って+1する必要があります。〔図1の①〕」

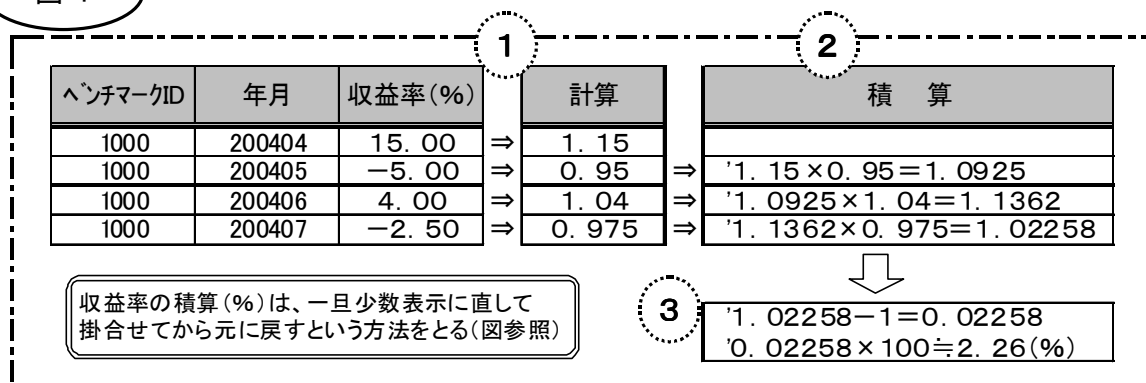
先輩:「小数に計算し直したデータを“計算”という名のフィールドを用意して取っておく。これが図の①で表していることだね。例えば、ベンチマークIDが1000の、年月が2004年4月から7月を積算する場合、順に1.15, 0.95, 1.04, 0.975, のようになる。」

野口:「次に、この“計算”フィールドを順に掛合せていく処理が必要です。これが、図の②のところですね。2レコード目から順に掛けた結果にまた掛けていくことになるので、前のレコードの結果を覚えておくような処理があるのではないのでしょうか。」

先輩:「最後のレコードの“積算”というフィールドに求める結果が入っているわけだけど、最終的にはこれを収益率(%)の形にしなければいけないと思うから、③のような処理、つまり結果から1を引いて100を掛けて表示する、というのが必要と思うよ。」

野口:「そうですね。」

図 1



先輩:「さて、じゃ具体的な方法としてボクの家を見てもらおう。まず、①の部分だけを行う処理を作る。[ベンチマーク]テーブルから、1つのベンチマークIDのデータだけ抽出するクエリを作る。この中に、“収益率を100で割って1を足す”フィールドを追加しておく。(図2, クエリのデザインを参照)

このクエリを、[テーブル作成クエリ]として実行・保存する。〕([クエリの種類]→[テーブル作成(K)])

野口:「テーブル名はどうしましょうか？」

先輩:「そうだね。『べ実\_抽出』テーブルでどう？」(野口:「ダサっ(ー;)」)

図 2

ベンチマーク実績テーブルから、1つのベンチマークだけを抽出して別のテーブルを作成する、「テーブル作成クエリ」を作る。

空きフィールドに「計算」として、「収益率」が%単位のデータなので、累積演算のために100分の1してから1を加えた値を入れる。

クエリのデザインビュー

フィールド:	ベンチマークID	年月	収益率	計算:[収益率]/100+1
テーブル:	ベンチマーク実績	ベンチマーク実績	ベンチマーク実績	
並べ替え:		昇順		
表示:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
抽出条件:	1000			
または:				

テーブル作成クエリにして、「べ実\_抽出」というテーブル名で保存する

先輩:「次に、②の部分処理を説明するね。今作った『ベ実\_抽出』テーブルをもとにもう1つクエリを作る。空きフィールドに、“積算”というフィールドを作って計算式を図3のように入れてもらう。出来たら、『Q\_ベ実\_累積』という名前で作って保存してね。」(図3, クエリのデザインと代入式を参照)

図 3

次に、「ベ実\_抽出」テーブルをもとに、全フィールドを指定した選択クエリを作る。空きフィールドに、図のような累積演算の式を入れる。

クエリのデザインビュー

フィールド:	ベンチマークID	年月	収益率	計算	積算:[計算]*.....
テーブル:	ベ実_抽出	ベ実_抽出	ベ実_抽出	ベ実_抽出	
並べ替え:					
表示:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
抽出条件:					
または:					

```
積算:[計算]*NZ(DLookup("積算","Q_ベ実_累積","年月=" &
NZ(DMax("年月","ベ実_抽出","年月"<" & [年月]),0),1)
```

野口:「うへー、何ですかこの長ったらしい式は？」

先輩:「関数だよ、“NZ”というのは、フィールドの値がNullの場合の対応、(この場合は1を入れている)

DLookupは指定したテーブルやクエリの中から条件に合ったレコードを持ってくる機能がある。

DMaxは指定した中から最大値を持ってくる。要するに、『ベ実\_抽出』テーブルの“年月”と比べながら、

今のレコードの1つ前の“積算”フィールドに“計算”フィールドを掛けていく処理になっているんだ。」

野口:「年月の範囲の指定は？」

先輩:「最初の『ベ実\_抽出』テーブルを作る処理の中で条件を入れておけばいい。」

野口:「ふ〜む。でも関数の形を見ただけで拒否反応示しそうな感じ。」

先輩:「じゃ、キミの案はできているのかな？」

野口:「では、私の案を見てもらいましょう。まず、対象のベンチマークと年月の範囲を指定するフォームを作ります。(図4)『F\_積算テスト\_条件入力』という名前にしています。」

先輩:「フム、フム。本格的だね。」

野口:「このフォームには、条件入力のテキストボックスの他に、計算結果や%表示の枠を作りました。条件を入れて、[計算開始]を押すと、結果が表示されるようになります。」

野口:「次に、処理の流れを作って、イメージがわかるようにしておきます。(図5)」

図 4

F\_積算テスト\_条件入力: フォーム

### 積算テスト

**条件入力**

ベンチマークID

抽出開始年月

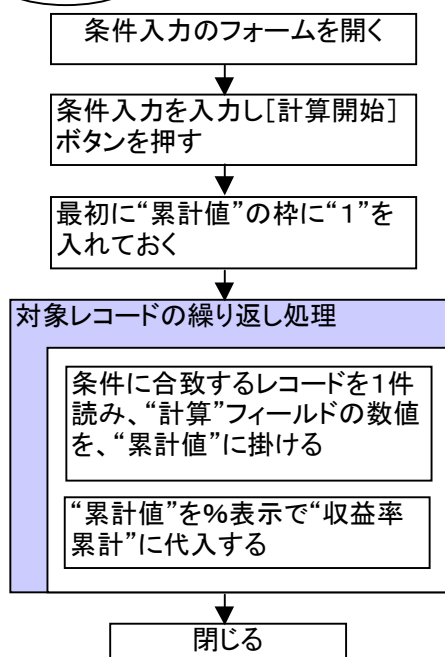
抽出終了年月

**計算結果**

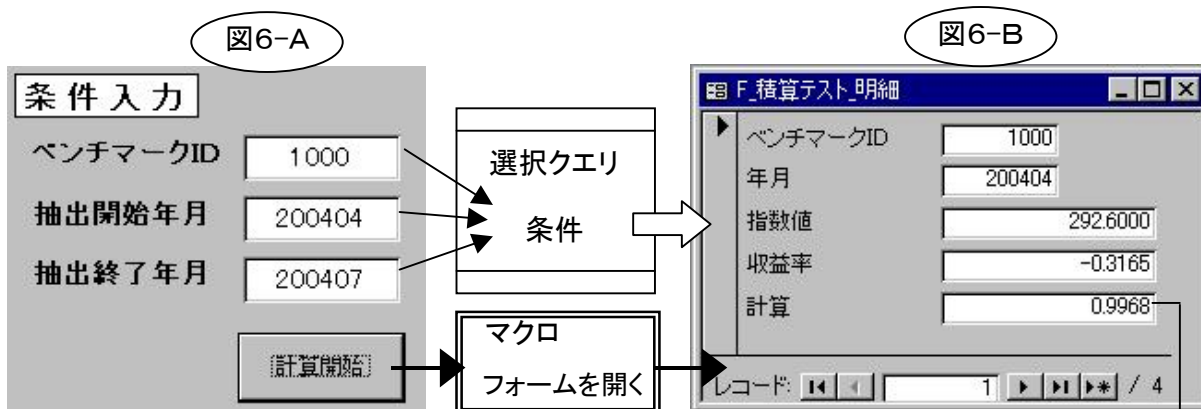
累計値

収益率累計

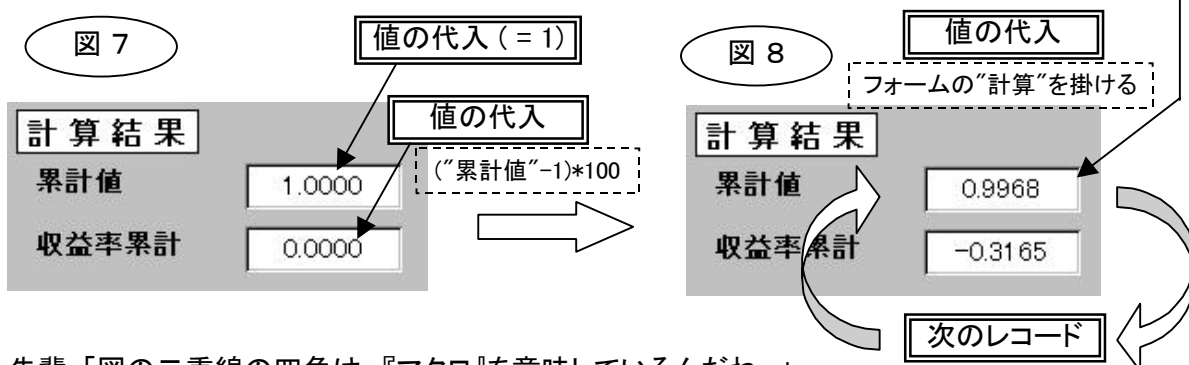
図 5



先輩:「この網掛けのところは、『繰り返し処理』になるんだね。具体的にはどんな処理をするの。」  
 野口:「まず、フォームの条件をもとにした、選択クエリを作ります。先輩の“図2”とほぼ同じ要領です。それから、このクエリをもとにした単票形式のフォームを作ります。これは、クエリのレコードを一件ずつ処理するためと、その様子を目に見えるようにするためです。」(図6. A, B)



野口:「繰り返し処理に入る前に、計算結果の累計値には“1”を代入しておきます。それと、収益率累計には“(累計値-1)\*100”を代入します。%表示の為です。」(図7)



先輩:「図の二重線の四角は、『マクロ』を意味しているんだね。」  
 野口:「そうです。次に、この“累計値”にデータ側のフォームの“計算”の値を掛けて、次のレコードを読む、という繰り返し処理を行います。(図8)この部分は、マクロ名を独立させておいて、“繰り返し条件式”には、『条件入力フォーム』の条件を指定します。」  
 先輩:「マクロとフォームの関係をもう一度詳細に教えてくれる？」  
 野口:「マクロについては、積算処理全体で『積算テスト』という名前にしました。その中に、機能を分けて、『開始』『積算開始』『くり返し処理』『閉じる』という構成になっています。」(表1参照)

マクロ名	アクション	備考
開始	フォームを開く	『条件入力』のフォームを開く
積算開始	フォームを開く	『明細』のフォーム(選択クエリをもとにした)を開く
	値の代入	“累計値”に1を入れる
	値の代入	“累計値”を100を基準の%表示に直し、“収益率累計”に入れる
	マクロの実行	マクロ『くり返し処理』を、条件に合致する間で行う
	閉じる	『明細』フォームを閉じる
くり返し処理	値の代入	“累計値”に、『明細』中の“計算”の値を掛ける
	値の代入	“累計値”を100を基準の%表示に直し、“収益率累計”に入れる
	レコードの移動	『明細』フォームを次のレコードに移動する
閉じる	閉じる	『条件入力』のフォームを閉じる

先輩:「『開始』は条件入力(図4)のフォームを開く。『閉じる』は閉じるボタンに対応しているんだね。あとの2つをもう少し詳しく説明してくれるかな。」  
 野口:「では、次ページの図で説明しましょう。」



野口:「まず選択クエリのデザインを説明します。もとなるテーブルは『ベンチマーク実績』です。図2と基本的に同じですが、抽出条件を図9の①②のように指定します。この条件の中にはフォーム『F\_積算テスト\_条件入力』のテキストボックスの名前が関係していて、フォームに入力した値や値の範囲に合致するレコードが抽出される、という仕組みです。」

先輩:「②のところの、Between…And…というのは、この範囲を抽出条件とするという意味だね。」

野口:「はい、そうです。“計算開始”というボタンには、クリックすると『積算開始』マクロが起動するようにプロパティに指定しておきます。『積算開始』マクロの最初の処理が[フォームを開く]で、“F\_積算テスト\_明細”を開きます。このフォームは、選択クエリから作っているの、入力した条件のレコードを抽出し1件目を表示する、ということです。」(図9参照)

図 9

クエリのデザインビュー

フィールド:	ベンチマークID	年月	指数値	収益率	計算:[収益率]/100+1
テーブル:	ベンチマーク実績	ベンチマーク実績	ベンチマーク実績	ベンチマーク実績	
並べ替え:					
表示:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
抽出条件:	①	②			

① [Forms]![F\_積算テスト\_条件入力]![txt\_ベンチマークID]

② Between [Forms]![F\_積算テスト\_条件入力]![txt\_抽出開始年月]  
And [Forms]![F\_積算テスト\_条件入力]![txt\_抽出終了年月]

フォーム  
「F\_積算テスト\_条件入力」

抽出クエリをもとにしたフォーム  
「F\_積算テスト\_明細」

テキストボックスの名前

- txt\_ベンチマークID
- txt\_抽出開始年月
- txt\_抽出終了年月

マクロ「積算開始」  
“フォームを開く”

表2の ①

野口:「さて、次にマクロ『積算開始』の内容を詳しく見てみましょう。下表のようにデザインしました。最初の“フォームを開く”については、上の図9の右下ですね。明細のフォームを開きます。また、②の“値の代入”については、図7をもう一度見ていただくとイメージがわかると思いますが、条件入力フォームの計算結果フィールドの累計値に初期値を入れています。また③の“値の代入”は%表示で見せるために、単に計算しているだけです。」

マクロ『積算開始』のデザイン

表 2

アクション	アクションの引数
① フォームを開く	フォーム名:F_積算テスト_明細 ビュー:フォームビュー
② 値の代入	アイテム:[Forms]![F_積算テスト_条件入力]![txt_累計値] 式:=1
③ 値の代入	アイテム:[Forms]![F_積算テスト_条件入力]![txt_収益率累計] 式:=( [Forms]![F_積算テスト_条件入力]![txt_累計値]-1)*100
④ マクロの実行	マクロ名:M_積算テスト.くり返し処理 式:([Forms]![F_積算テスト_明細]![txt_ベンチマークID]=[Forms]![F_積算テスト_条件入力]![txt_ベンチマークID]) And ([Forms]![F_積算テスト_明細]![txt_年月] Between [Forms]![F_積算テスト_条件入力]![txt_抽出開始年月] And [Forms]![F_積算テスト_条件入力]![txt_抽出終了年月])
⑤ 閉じる	オブジェクトの種類:フォーム オブジェクト名:F_積算テスト_明細

先輩:「次の行の“マクロの実行④”では、『くり返し処理』というマクロをさらに起動するんだね。」

野口:「“マクロの実行”というアクションには、引数としてマクロの名前の他に[実行回数]とか、[繰り返し条件式]というのが指定できます。ここでは、[繰り返し条件式]の方を使って、条件入力フォームに入力した条件(ベンチマークID, 抽出開始年月, 抽出終了年月)に一致するデータが終わるまで処理をくり返すように設定しました。」

先輩:「なるほど。」

野口:「次に『くり返し処理』マクロのほうですが、これは“値の代入”、“値の代入”、“レコードの移動”、という3つのアクションから出来ていて、この3つがくり返されるわけです。前の図8と図6-Bを見てイメージを理解してください。明細フォームの[計算]というフィールドの値を、累計値に掛け(“値の代入”⑥)、収益率累計%表示(“値の代入”③)をして、次のレコードに移動(“レコードの移動”⑦)、という処理のくり返しになります。」(表3)

マクロ『くり返し処理』のデザイン

アクション	アクションの引数
値の代入 ⑥	アイテム:[Forms]![F_積算テスト_条件入力][txt_累計値] 式:=[Forms]![F_積算テスト_条件入力][txt_累計値]*[Forms]![F_積算テスト_明細][txt_計算]
値の代入 ③	アイテム:[Forms]![F_積算テスト_条件入力][txt_収益率累計] 式:=[Forms]![F_積算テスト_条件入力][txt_累計値]*[Forms]![F_積算テスト_明細][txt_計算]
レコードの移動 ⑦	オブジェクトの種類:フォーム オブジェクト名:F_積算テスト_明細 レコード:次のレコード

表3

先輩:「引数の式などはとても長いように感じるけど、タイプインではなくて“式ビルダ”を使ったね。」

野口:「そうですね。例えば、“マクロの実行”アクションの引数には式を入れるところがありますね。

この右側に... というボタンがあるので、これを押すと“式ビルダ”が出てきます。」(図10)

先輩:「図のように、すでに作ったフォームやクエリの内容(テキストボックスやボタンの名前など)を表示できるので、選択(クリック)していったダブルクリックすれば、上の枠に式ができる。タイプインより間違えがなくて早くて便利だね。」

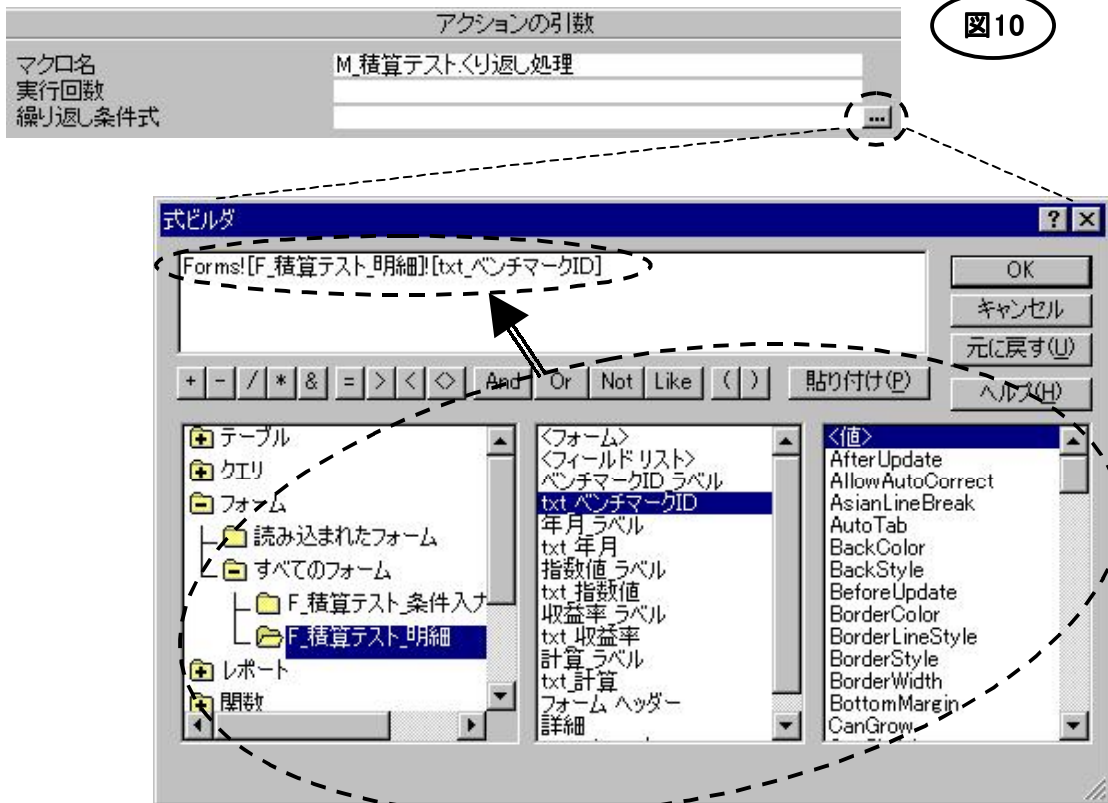


図10

野口:「どうですか?こんなやり方の“くり返し処理”なんですけど…」

先輩:「すごいっ!ボクのやり方より人にわかりやすく、関数やBASICが不要なもの。」

いや、よく出来ました。」

野口:「ホントにそう思います?やった!暮に紅白見ないで悩んだ甲斐があったゾ。」(TOT)カンルイ!!

